

```

LOC  OBJ      SER      SOURCE STATEMENT
1
2 ;*****
3 ;*
4 ;*              INTELLEC SERIES II BOOT/MONITOR          9800805A  *
5 ;*              VERSION 1.2                               *
6 ;*
7 ;*      COPYRIGHT (C) 1978 INTEL CORPORATION.  ALL RIGHTS  *
8 ;*      RESERVED.  NO PART OF THIS PROGRAM OR PUBLICATION *
9 ;*      MAY BE REPRODUCED, TRANSMITTED, TRANSCRIBED,     *
10 ;*      STORED IN A RETRIEVAL SYSTEM, OR TRANSLATED INTO  *
11 ;*      ANY LANGUAGE OR COMPUTER LANGUAGE, IN ANY FORM   *
12 ;*      OR BY ANY MEANS, ELECTRONIC, MECHANICAL, MAGNETIC,*
13 ;*      OPTICAL, CHEMICAL, MANUAL OR OTHERWISE, WITHOUT  *
14 ;*      THE PRIOR WRITTEN PERMISSION OF INTEL CORPORATION.*
15 ;*      3065 BOWERS AVENUE, SANTA CLARA, CALIFORNIA 95051.*
16 ;*
17 ;*****
18 ; <LEGAL COMMAND> ::= <ASSIGN I/O COMMAND>
19 ;                   <DISPLAY MEMORY COMMAND>
20 ;                   <ENDFILE COMMAND>
21 ;                   <FILL MEMORY COMMAND>
22 ;                   <PROGRAM EXECUTE COMMAND>
23 ;                   <HEXADECIMAL ARITHMETIC COMMAND>
24 ;                   <MOVE MEMORY COMMAND>
25 ;                   <LEADER COMMAND>
26 ;                   <QUERY STATUS COMMAND>
27 ;                   <READ HEXADECIMAL FILE COMMAND>
28 ;                   <SUBSTITUTE MEMORY COMMAND>
29 ;                   <WRITE HEXADECIMAL RECORD COMMAND>
30 ;                   <REGISTER MODIFY COMMAND>
31 ;                   <TRANSFER CONTROL TO DIAGNOSTIC PROGRAM COMMAND>
32 ; <ASSIGN I/O COMMAND> ::= A<LOGICAL DEVICE>=<PHYSICAL DEVICE>
33 ; <DISPLAY MEMORY COMMAND> ::= D<NUMBER>,<NUMBER>
34 ; <ENDFILE COMMAND> ::= E<NUMBER>
35 ; <FILL MEMORY COMMAND> ::= F<NUMBER>,<NUMBER>,<NUMBER>
36 ; <PROGRAM EXECUTE COMMAND> ::= G<NUMBER>,<NUMBER>,<NUMBER>
37 ; <HEXADECIMAL ARITHMETIC COMMAND> ::= H<NUMBER>,<NUMBER>
38 ; <MOVE MEMORY COMMAND> ::= M<NUMBER>,<NUMBER>,<NUMBER>
39 ; <LEADER COMMAND> ::= H
40 ; <QUERY STATUS COMMAND> ::= Q
41 ; <READ HEXADECIMAL FILE COMMAND> ::= R<NUMBER>
42 ; <SUBSTITUTE MEMORY COMMAND> ::= S<NUMBER><COMMA>...
43 ; <WRITE HEXADECIMAL RECORD COMMAND> ::= W<NUMBER>,<NUMBER>
44 ; <REGISTER MODIFY COMMAND> ::= X<REGISTER IDENTIFIER><NUMBER>...
45 ; <TRANSFER CONTROL TO DIAGNOSTIC PROGRAM COMMAND> ::= Z$
46 ; <LOGICAL DEVICE> ::= LOCAL!CONSOLE!READER!LIST!PUNCH
47 ; <PHYSICAL DEVICE> ::= CRT!TTY!PTR!PTP!LPT!BATCH!1!2
48 ; <REGISTER IDENTIFIER> ::= A!B!C!D!E!F!H!I!L!M!P!S
49 ; <NUMBER> ::= <HEX DIGIT>
50 ;                   <NUMBER><HEX DIGIT>
51 ; <HEX DIGIT> ::= 0!1!2!3!4!5!6!7!8!9!A!B!C!D!E!F
52 ;*****

```

```

LOC  OBJ      SEQ      SOURCE STATEMENT
0000      53 $      TITLE      (' INTELLEC SERIES II MONITOR, VERSION 1.2, 4 JANUARY 1978 ')
0012      54 VER      EQU        12          ; VERSION 1.2
0401      55 VERH     EQU        12H        ; STORAGE REPRESENTATION OF VERSION
0401      56 DATE     EQU        0401H      ; CREATION DATE, 04 JANUARY 1978
57 ; NOTE:
58 ; THE DATE SHOWN ABOVE IS ENCODED IN A TWO BYTE FIELD IN BOTH THE BOOTSTRAP
59 ; PROM AND THE MONITOR PROM IN ORDER TO CONTROL NEW RELEASES OF THIS PROGRAM.
60 ; THE DATE CODE IS LOCATED AT ADDRESSES 0E804H AND 0E805H IN THE BOOTSTRAP
61 ; AND AT ADDRESSES DF824H AND DF825H IN THE MONITOR.
62 ; THE VERSION CODE IS LOCATED IN THE MONITOR ROM AT ADDRESS DF82FH.
63 ; WHEN A NEW RELEASE IS ISSUED, PLEASE CHANGE THE DATE AND VERSION CODES.
64 ; THE COPYRIGHT NOTICE IS LOCATED IN THE MONITOR ROM BEGINNING AT DF830H.
65 ;*****
66 ;*
67 ;*          SYMBOL DEFINITIONS
68 ;*
69 ;*****
70 ;
71 ; INTELLEC SERIES II SYSTEM CONSTANTS
72 ;
73 ; INTEGRATED CONSOLE I/O PORTS
74 ;
00C0      75 COMI     EQU        0C0H          ; CONSOLE INPUT DATA PORT
00C0      76 CONO     EQU        0C0H          ; CONSOLE OUTPUT DATA PORT
00C1      77 COMS     EQU        0C1H          ; CONSOLE STATUS PORT
00C1      78 CONC     EQU        0C1H          ; CONSOLE CONTROL PORT
79 ;
80 ; SYSTEM BOOTSTRAP CONSTANTS (ISSUED TO PORT CPUC)
81 ;
0000      82 DISABL   EQU        00H          ; DISABLE INTERRUPTS
0005      83 ENABL    EQU        05H          ; ENABLE INTERRUPTS
0000      84 DISAXP   EQU        00H          ; DISABLE AUXILIARY PROM
85
0008      86 ENAXP    EQU        08H          ; ENABLE AUXILIARY PROM
0001      87 BOVROF   EQU        01H          ; TURN OFF BUS OVERRIDE
0009      88 BOVRON   EQU        09H          ; TURN ON BUS OVERRIDE
0004      89 BTDOGDF  EQU        04H          ; TURN OFF BOOT/DIAGNOSTIC
000C      90 BTDOGN   EQU        0CH          ; TURN ON BOOT/DIAGNOSTIC
0002      91 MOVBOT   EQU        02H          ; MOVE BOOT TO 0E800H
92 ;
93 ; SYSTEM I/O PORTS
94 ;
00FE      95 CPUS     EQU        0FEH          ; CPU STATUS PORT
00FF      96 CPUC     EQU        0FFH          ; CPU CONTROL PORT (CONTROLS BOOT & AUX.PROM)
97 ;
98 ; SYSTEM INTERRUPT CONSTANTS
99 ;
0012      100 ICM1    EQU        00010010B    ; INITIALIZATION COMMAND WORD 1
0000      101 ICM2    EQU        00000000B    ; INITIALIZATION COMMAND WORD 2
0008      102 OCM3    EQU        00001011B    ; OPERATION COMMAND WORD 3
0020      103 EDI     EQU        00100000B    ; END OF INTERRUPT
104 ;
105 ; SYSTEM INTERRUPT MASKS AND VALUES
106 ;
0001      107 INTO     EQU        00000001B    ; MASK FOR INTERRUPT LEVEL 0

```

```

LOC  OBJ          SEQ      SOURCE STATEMENT
0002          108 INT1     EQU      00000010B
0004          109 INT2     EQU      00000100B
0008          110 INT3     EQU      00001000B
0010          111 INT4     EQU      00010000B
0020          112 INT5     EQU      00100000B
0040          113 INT6     EQU      01000000B
0080          114 INT7     EQU      10000000B
0000          115 INTA     EQU      00000000B      ; NO INTERRUPTS ALLOWED AT ALL
          116 ;
          117 ; SYSTEM INTERRUPT I/O PORTS
          118 ;
00FD          119 SICP0     EQU      0FDH          ; INITIALIZATION COMMAND PORT 0
00FC          120 SICP1     EQU      0FCH          ; INITIALIZATION COMMAND PORT 1
00FD          121 SOCP0     EQU      0FDH          ; OPERATION COMMAND PORT 0
00FC          122 SOCP1     EQU      0FCH          ; OPERATION COMMAND PORT 1
          123 ;
          124 ; DEDICATED PROM PROGRAMMER CONSTANTS (USED IN C,P,T COMMANDS)
          125 ;
0002          126 PCOMP     EQU      00000010B      ; PROGRAMMING COMPLETE
0001          127 PGRDY     EQU      00000001B      ; PROM READY
0020          128 PSOCK     EQU      00100000B      ; 16 PIN SOCKET SELECTED
0010          129 PWIB      EQU      00010000B      ; SELECT UPPER NIBBLE
          130 ;*****
          131 ;
          132 ; INTELLEC SERIES II I/O SUBSYSTEM CONSTANTS
          133 ;
          134 ; TTY AND CRT MODE INSTRUCTION DEFINITIONS, I.E. USART MODE CONTROL
          135 ; WORD (FIRST CONTROL BYTE AFTER RESET)
          136 ;
0003          137 R64X      EQU      00000011B      ; 64 X BAUD RATE
0002          138 R16X      EQU      00000010B      ; 16 X BAUD RATE
0001          139 R1X       EQU      00000001B      ; 1 X BAUD RATE
0000          140 SYNC      EQU      00000000B      ; SYNC MODE
003C          141 CL8       EQU      00001100B      ; CHARACTER LENGTH = 8
0098          142 CL7       EQU      00001000B      ; CHARACTER LENGTH = 7
0004          143 CL6       EQU      00000100B      ; CHARACTER LENGTH = 6
0000          144 CL5       EQU      00000000B      ; CHARACTER LENGTH = 5
0010          145 PENB      EQU      00010000B      ; PARITY ENABLE
0020          146 PEVEN     EQU      00100000B      ; EVEN PARITY
00C0          147 ST2       EQU      11000000B      ; 2 STOP BITS
0080          148 ST15      EQU      10000000B      ; 1 5 STOP BITS
0040          149 ST1       EQU      01000000B      ; 1 STOP BIT
          150 ;
          151 ; TTY AND CRT COMMAND INSTRUCTION DEFINITIONS (USART COMMAND CONTROL WORD)
          152 ;
0001          153 TXEN      EQU      00000001B      ; TRANSMITTER ENABLE
0002          154 DTR       EQU      00000010B      ; DATA TERMINAL READY
0004          155 RXEN      EQU      00000100B      ; ENABLE RECEIVER
0008          156 SBCH      EQU      00001000B      ; SEND BREAK CHARACTER
0010          157 CLERR     EQU      00010000B      ; CLEAR ERROR
0020          158 RTS       EQU      00100000B      ; SET REQUEST TO SEND OUTPUT
0040          159 USRST     EQU      01000000B      ; USART RESET - RETURN TO MODE CONTROL CYCLE
0080          160 ENHM      EQU      10000000B      ; ENABLE HUNT MODE
          161 ;
          162 ; TTY/CRT STATUS WORD BIT DEFINITIONS

```

```

LOC  OBJ          SEQ          SOURCE STATEMENT
163 ;
0001          164 TRDY   EQU    0000001B      ; TRANSMIT READY
0002          165 RRDY   EQU    00000010B     ; RECEIVE BUFFER READY
0004          166 TKBE   EQU    00000100B     ; TRANSMIT BUFFER EMPTY
0008          167 RPAR   EQU    00001000B     ; RECEIVE PARITY ERROR
0010          168 ROV    EQU    00010000B     ; RECEIVE OVERRUN ERROR
0020          169 RFR    EQU    00100000B     ; RECEIVE FRAMING ERROR
0040          170 SYND   EQU    01000000B     ; SYNC DETECTED
0080          171 DSR    EQU    10000000B     ; DATA SET READY INPUT
172 ;
173 ; TTY TAPE READER CONSTANTS
174 ;
0028          175 RADCT  EQU    40              ; TTY TAPE READER ADVANCE TIMER COUNT
00FA          176 RTOCT  EQU    250            ; TTY TAPE READER TIMEOUT COUNT
0027          177 TADY   EQU    TXEN OR RXEN OR RTS OR DTR
0025          178 COMD   EQU    TXEN OR RXEN OR RTS
179 ;
180 ; TTY I/O PORTS
181 ;
00F4          182 TTYI   EQU    0F4H             ; TTY INPUT DATA PORT
00F4          183 TTYO   EQU    0F4H             ; TTY OUTPUT DATA PORT
00F5          184 TTY5   EQU    0F5H             ; TTY INPUT STATUS PORT
00F5          185 TTYC   EQU    0F5H             ; TTY OUTPUT CONTROL PORT
186 ;
187 ; USER I/O PORTS
188 ;
00F6          189 USCI   EQU    0F6H             ; USER INPUT DATA PORT
00F7          190 USCS   EQU    0F7H             ; USER INPUT STATUS PORT
00F6          191 USCO   EQU    0F6H             ; USER OUTPUT DATA PORT
00F7          192 USCC   EQU    0F7H             ; USER OUTPUT CONTROL PORT
193 ;
194 ; INTERVAL TIMER CONSTANTS
195 ;
0000          196 CTROS  EQU    00000000B       ; COUNTER 0 SELECT
0040          197 CTR1S  EQU    01000000B       ; COUNTER 1 SELECT
0080          198 CTR2S  EQU    10000000B       ; COUNTER 2 SELECT
0000          199 LCTR   EQU    00000000B       ; LATCHING COUNTER
0020          200 RLMB   EQU    00100000B       ; READ/LOAD MSB ONLY
0010          201 RLLB   EQU    00010000B       ; READ/LOAD LSB ONLY
0030          202 RLLM   EQU    00110000B       ; READ/LOAD LSB,MSB
0000          203 MODE0  EQU    00000000B       ; MODE 0
0002          204 MODE1  EQU    00000010B       ; MODE 1
0004          205 MODE2  EQU    00000100B       ; MODE 2
0006          206 MODE3  EQU    00000110B       ; MODE 3
0008          207 MODE4  EQU    00001000B       ; MODE 4
000A          208 MODE5  EQU    00001010B       ; MODE 5
0001          209 BCDC   EQU    00000001B       ; BCD COUNTER
0007          210 B9600  EQU    7                ; 9600 BAUD RATE FACTOR
0020          211 B2400  EQU    32               ; 2400 BAUD RATE FACTOR
02BA          212 B0110  EQU    698              ; 110 BAUD RATE FACTOR
213 ;
214 ; INTERVAL TIMER (8253) I/O PORTS
215 ;
00FD          216 CTROP  EQU    0F0H             ; LOAD COUNTER 0 OUTPUT COMMAND PORT
00F1          217 CTR1P  EQU    0F1H             ; LOAD COUNTER 1 OUTPUT COMMAND PORT

```

```

LOC OBJ          SEQ          SOURCE STATEMENT
00F2            218 CTR2P    EQU      0F2H          ; LOAD COUNTER 2 OUTPUT COMMAND PORT
00F3            219 ITCP     EQU      0F3H          ; INTERVAL TIMER OUTPUT COMMAND PORT
                220 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
                221 ;
                222 ; I/O CONTROLLER SYSTEM CONSTANTS
                223 ;
                224 ; I/O CONTROLLER PORTS
                225 ;
00C0            226 IOC1     EQU      0C0H          ; I/O CONTROLLER INPUT DATA (FROM DBB) PORT
00C0            227 IOCO     EQU      0C0H          ; I/O CONTROLLER OUTPUT DATA (TO DBB) PORT
00C1            228 IOCS     EQU      0C1H          ; I/O CONTROLLER INPUT DBB STATUS PORT
00C1            229 IOCC     EQU      0C1H          ; I/O CONTROLLER OUTPUT CONTROL COMMAND PORT
                230 ;
                231 ; CRT, KEYBOARD, AND FLOPPY DISK COMMANDS
                232 ;
0010            233 CRTC     EQU      10H          ; CRT OUTPUT DATA COMMAND
0011            234 CRTS     EQU      11H          ; CRT DEVICE STATUS COMMAND
0012            235 KEYC     EQU      12H          ; KEYBOARD INPUT DATA COMMAND
0013            236 KSTS     EQU      13H          ; KEYBOARD DEVICE STATUS COMMAND
0014            237 KIHT     EQU      14H          ; RESERVED
0015            238 WPBC     EQU      15H          ; FLOPPY PARAMETER BLOCK TRANSFER COMMAND
0016            239 WPBCC    EQU      16H          ; FLOPPY PARAMETER BLOCK(CONT) TRANSFER COMMAND
0017            240 WDBC     EQU      17H          ; FLOPPY DATA BLOCK OUTPUT COMMAND
0018            241 WDBCC    EQU      18H          ; RESERVED
0019            242 RDBC     EQU      19H          ; FLOPPY INPUT DATA BLOCK COMMAND
001A            243 RDBCC    EQU      1AH          ; RESERVED
001B            244 RRSTS     EQU      1BH          ; FLOPPY RESULT STATUS COMMAND
001C            245 RDSTS     EQU      1CH          ; FLOPPY DEVICE STATUS COMMAND
                246 ;
                247 ; CRT, KEYBOARD, AND FLOPPY STATUS BITS
                248 ;
0001            249 KRDY     EQU      0000001B        ; KEYBOARD READY WITH DATA
0001            250 FRDY     EQU      0000001B        ; FLOPPY READY FOR DATA
                251 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*
                252 ;
                253 ; PARALLEL I/O SYSTEM CONSTANTS
                254 ;
                255 ; PARALLEL I/O PORTS
                256 ;
00F8            257 P10I     EQU      0F8H          ; PARALLEL I/O INPUT DATA (FROM DBB) PORT
00F8            258 P10O     EQU      0F8H          ; PARALLEL I/O OUTPUT DATA (TO DBB) PORT
00F9            259 P10S     EQU      0F9H          ; PARALLEL I/O INPUT DBB STATUS PORT
00F9            260 P10C     EQU      0F9H          ; PARALLEL I/O OUTPUT CONTROL COMMAND PORT
                261 ;
                262 ; PTR, PTP, LPT AND UPP COMMANDS
                263 ;
0010            264 RDRC     EQU      010H          ; READER DATA TRANSFER COMMAND
0060            265 PTRREV    EQU      01100000B        ; READER REVERSE DIRECTION 1 FRAME OPTION
0040            266 PTRADV    EQU      01000000B        ; READER ADVANCE DIRECTION 1 FRAME OPTION
0011            267 RSTC     EQU      011H          ; READER DEVICE STATUS COMMAND
0012            268 PUNC     EQU      012H          ; PUNCH DATA TRANSFER COMMAND
0013            269 PSTC     EQU      013H          ; PUNCH DEVICE STATUS COMMAND
0014            270 LPTC     EQU      014H          ; LINE PRINTER DATA TRANSFER COMMAND
0015            271 LSTC     EQU      015H          ; LINE PRINTER STATUS COMMAND
0016            272 WPPC     EQU      016H          ; WRITE TO UPP COMMAND

```

```

LOC  OBJ          SEQ          SOURCE STATEMENT
0017          273 RPPC   EQU      017H          ; READ FROM UPP COMMAND
0018          274 RPSTC  EQU      018H          ; READ UPP STATUS COMMAND
          275 ;
          276 ; LPT, PTR AND PTP STATUS BITS
          277 ;
0001          278 LPTRY   EQU      00000001B     ; LPT READY
0001          279 PTRDY   EQU      00000001B     ; PTR READY WITH DATA
0001          280 PTPRY   EQU      00000001B     ; PTP READY FOR DATA
          281 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*-----*
          282 ;
          283 ; PARALLEL I/O AND I/O CONTROLLER SYSTEM COMMANDS
          284 ;
0000          285 PACIFY  EQU      00H           ; REINITIALIZE SYSTEM
0001          286 ERESET  EQU      01H           ; ERROR RESET
0002          287 SYSTAT  EQU      02H           ; SYSTEM STATUS
0003          288 DSTAT   EQU      03H           ; DEVICE STATUS
0004          289 SRQACK  EQU      04H           ; DEVICE SERVICE REQUEST ACK
0005          290 SRQACK  EQU      05H           ; SYSTEM SERVICE REQUEST ACK
0006          291 SRQ     EQU      06H           ; SERVICE REQUEST
          292 ;
          293 ; PARALLEL I/O AND I/O CONTROLLER DIAGNOSTIC COMMANDS
          294 ;
0007          295 DECHO   EQU      07H           ; DATA ECHO TEST
0008          296 CSMEM   EQU      08H           ; CHECKSUM MEMORY
0009          297 TRAM    EQU      09H           ; TEST RAM
000A          298 SINT    EQU      0AH           ; SYSTEM INTERRUPT CONTROL
          299 ;
          300 ;
          301 ; PARALLEL I/O AND I/O CONTROLLER STATUS CONSTANTS
          302 ;
0001          303 OBF     EQU      00000001B     ; SLAVE OUTPUT BUFFER IS FULL
0002          304 IBF     EQU      00000010B     ; SLAVE INPUT BUFFER IS FULL
0004          305 FO      EQU      00000100B     ; FLAG 0 - SLAVE IS BUSY, MASTER IS LOCKED OUT
0008          306 CNOTD  EQU      00001000B     ; DBB CONTAINS CONTROL INFO NOT DATA
          307 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*-----*
          308 ;
          309 ; FDCC (FLOPPY DISKETTE CHANNEL COMMAND) CONSTANTS
          310 ;
0004          311 OPCPL   EQU      4             ; DISK COMPLETION STATUS
0079          312 LOWM    EQU      79H           ; LOW(IOPB)
007A          313 HI     EQU      7AH           ; HIGH(IOPB)
007B          314 RSTS    EQU      7BH           ; DISK RESULT STATUS INPUT PORT
0078          315 DSTS    EQU      7CH           ; DISK STATUS INPUT PORT
3000          316 TRKO    EQU      3000H        ; FIRST ADDRESS OF DISK BOOTSTRAP
          317 ;
          318 ;       CONDITIONAL ASSEMBLY SWITCHES
          319 ;
0000          320 FALSE  EQU      0             ;
FFFF          321 TRUE   EQU      NOT FALSE
00FF          322 MNSK   EQU      0FFH        ; SAFE MOVE OF 16 BITS INTO 8 BIT REGISTER
          323 ;
          324 ; GLOBAL CONSTANTS
          325 ;
007D          326 ONEMS  EQU      112          ; 1 MILLISECOND TIME CONSTANT
00FA          327 TOUT   EQU      250          ; 250 MS. COUNTER FOR READER TIMEOUT
  
```

LOC	OBJ	SEQ	SOURCE STATEMENT
000D		328 CR	EQU 0DH ; ASCII VALUE OF CARRIAGE RETURN
000A		329 LF	EQU 0AH ; ASCII VALUE OF LINE FEED
0003		330 ETX	EQU 03H ; MONITOR BREAK CHARACTER (CONTROL C)
		331 ;	
		332 ;	MONITOR I/O STATUS BYTE MASKS AND VALUES
		333 ;	
00FC		334 CMSK	EQU 11111100B ; MASK FOR LOCAL CONSOLE I/O
00F3		335 RMSK	EQU 11110011B ; MASK FOR READER INPUT
00CF		336 PMSK	EQU 11001111B ; MASK FOR PUNCH OUTPUT
003F		337 LMSK	EQU 00111111B ; MASK FOR LIST OUTPUT
		338 ;	----
0000		339 CTTY	EQU 00000000B ; LOCAL CONSOLE = TTY
0001		340 CCRT	EQU 00000010B ; LOCAL CONSOLE = CRT
0002		341 BATCH	EQU 00000010B ; BATCH MODE:
		342 ;	CONSOLE INPUT = READER, CONSOLE OUTPUT = LIST
0003		343 CUSE	EQU 00000011B ; USER DEFINED LOCAL CONSOLE I/O
		344 ;	----
0000		345 RTTY	EQU 00000000B ; READER = TTY
0004		346 RPTR	EQU 00000100B ; READER = PTR
0008		347 RUSE1	EQU 00001000B ; USER DEFINED READER (1)
000C		348 RUSE2	EQU 00001100B ; USER DEFINED READER (2)
		349 ;	----
0000		350 PTTY	EQU 00000000B ; PUNCH = TTY
0010		351 PPTP	EQU 00010000B ; PUNCH = PTP
0020		352 PUSE1	EQU 00100000B ; USER DEFINED PUNCH (1)
0030		353 PUSE2	EQU 00110000B ; USER DEFINED PUNCH (2)
		354 ;	----
0000		355 LTTY	EQU 00000000B ; LIST = TTY
0040		356 LCRT	EQU 01000000B ; LIST = CRT
0080		357 LLPT	EQU 10000000B ; LIST = LPT
00C0		358 LUSE	EQU 11000000B ; USER DEFINED LIST
		359 ;	
		360 ;	LOCAL I/O SUBSYSTEM INTERRUPT PORTS
		361 ;	
00FB		362 IICPO	EQU 0FBH ; INITIALIZATION COMMAND PORT 0
00FA		363 IICPI	EQU 0FAH ; INITIALIZATION COMMAND PORT 1
00FB		364 IOCPO	EQU 0FBH ; OPERATION COMMAND PORT 0
00FA		365 IOCP1	EQU 0FAH ; OPERATION COMMAND PORT 1
		366 ;	
		367 ;	LOCAL INTERRUPT STATUS AND CONTROL BITS
		368 ;	
0001		369 ITTYO	EQU 00000010B ; TTY OUTPUT INTERRUPT
0002		370 ITTYI	EQU 00000100B ; TTY INPUT INTERRUPT
0004		371 IPTP	EQU 00000100B ; PTP OUTPUT INTERRUPT
0008		372 IPTR	EQU 00001000B ; PTR INPUT INTERRUPT
0010		373 ICRT0	EQU 00010000B ; CRT OUTPUT INTERRUPT
0020		374 ICRT1	EQU 00100000B ; CRT INPUT INTERRUPT
0040		375 ILPT	EQU 01000000B ; LPT OUTPUT INTERRUPT
0080		376 MENB	EQU 10000000B ; ENABLE MONITOR INTERRUPTS EXCEPT LEVEL 7
		377 ;	*****
		378 ;	
		379 ;	BOOTSTRAP CONSTANTS
		380 ;	
00E7		381 FSTOP	EQU 0E7H ; FULL SYSTEM TOP OF MEMORY ADDRESS
00F7		382 FSTP	EQU 0F7H ; FULL SYSTEM TOP PAGE ADDRESS

```

LOC  OBJ          SEQ          SOURCE STATEMENT
0004          383 FDOC   EQU    004H          ; FLOPPY DISK OPERATION COMPLETE
007F          384 ACHRM  EQU    07FH          ; ASCII CHARACTER MASK
00FF          385 ITIMO  EQU    OFFH          ; IOC TIMEOUT CONSTANT
00FF          386 LBNK   EQU    OFFH          ; LOWER BYTE MASK
0041          387 ICFG   EQU    041H          ; CONSOLE CONFIGURATION STATUS
0001          388 ICNP   EQU    001H          ; INTEGRATED CONSOLE NOT PRESENT STATUS
0040          389 LSTE   EQU    040H          ; LIST DEVICE VALUE FOR CONSOLE
04CD          390 RTCC   EQU    1229          ; REAL TIME CLOCK 1MS CONSTANT
0008          391 DPRNT  EQU    08H           ; DISK READY MASK
0D00          392 TRKL   EQU    26*128          ; TRACK LENGTH
0004          393 PARML  EQU     4             ; PARAMETER LENGTH - 1
F809          394 COP    EQU    DF809H          ; ENTRY POINT FOR CONSOLE OUT
F821          395 IOCDP1 EQU    DF821H          ; ENTRY POINT FOR IOC DRIVER 1
F844          396 IOCDP2 EQU    DF844H          ; ENTRY POINT FOR IOC DRIVER 2
397 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*-----*
398 ;
399 ; PAGE 0 DEDICATED RAM LOCATIONS, INITIALIZED BY BOOTSTRAP PROM CODE.
400 ;
0000          401          ORG     0
0003          402 RESET:
403          403          DS     3             ; TRAP TO MONITOR RESTART
0001          404 IOBYT:
405          405          DS     1             ; I/O SYSTEM STATUS BYTE
0002          406 MENTOP:
407          407          DS     2             ; TOP OF RAM, ONLY H SAVED
0001          408 INITIO:
409          409          DS     1             ; INITIAL I/O CONFIGURATION
410 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*-----*
411 ;
412 ; BOOTSTRAP PROM CODE
413 ;
E900          414 BBASE   SET    DE800H
E900          415          ORG    BBASE
E800 C306E8    416          JMP    BSO             ; BRANCH AROUND DATE CODE BYTE
417 INIT:
E803 00       418          DB     0             ; INITIALLY
419          ; CONSOLE = TTY,
420          ; READER = TTY,
421          ; PUNCH = TTY,
422          ; LIST = TTY
E804 0104     423          DW    DATE             ; DATE STAMP FOR BOOTSTRAP PROM
424 ;
425 ; FUNCTIONS:
426 ;
427 ;       A.       INITIALIZE INTERRUPT SYSTEM AND REAL TIME CLOCK
428 ;               0. INITIALIZE PORT OFFH (CPUC)
429 ;               1. PROGRAM SYSTEM INTERRUPTS (8259)
430 ;               2. MASK ALL SYSTEM INTERRUPTS BUT TRAP LOGIC
431 ;               3. PROGRAM I/O SUBSYSTEM INTERRUPTS (8259)
432 ;               4. MASK ALL I/O SUBSYSTEM INTERRUPTS
433 ;               5. PROGRAM REAL TIME CLOCK
434 ;
435 BSO:
E806 F3       436          DI             ; DISABLE INTERRUPT SYSTEM
E807 3E02     437          MVI    A,NOVBOT      ; TURN ON RAM (ROM WILL NOW RESPOND ONLY TO ADDRESS E800H)

```



LOC	OBJ	SEQ	SOURCE STATEMENT
E009	D3FF	438	OUT CPUC
E00B	3E01	439	MVI A,BOVRDF ; TURN OFF BUS OVERRIDE
E00D	D3FF	440	OUT CPUC
E00F	3E05	441	MVI A,ENABL ; PSEUDO ENABLE OF INTERRUPTS
E011	D3FF	442	OUT CPUC
E013	3E08	443	MVI A,ENAXP ; ENABLE AUXILIARY PROM
E015	D3FF	444	OUT CPUC
E017	3E12	445	MVI A,ICW1 ; OUTPUT INITIALIZATION COMMAND WORD 1
E019	D3FD	446	OUT SICPO ; TO SYSTEM 8259
E01B	D3FD	447	OUT IICPO ; TO I/O 8259
E01D	3E0D	448	MVI A,ICW2 ; OUTPUT INITIALIZATION COMMAND WORD 2
E01F	D3FC	449	OUT SICP1 ; TO SYSTEM 8259
E021	D3FA	450	OUT IICP1 ; TO I/O 8259
E023	3EFE	451	MVI A,NOT INTO ; INITIALIZE MASK REGISTER
E025	D3FC	452	OUT SOCP1 ; FOR SYSTEM 8259
E027	3EFF	453	MVI A,NOT INTA ; INITIALIZE MASK REGISTER
E029	D3FA	454	OUT IOCP1 ; FOR I/O 8259
E02B	3EB6	455	MVI A,CTR2S OR MODE3 OR RLLM ; INITIALIZE 1MS REAL TIME CLOCK
E02D	D3F3	456	OUT ITCP
E02F	21CD04	457	LXI H,RTCC
E032	7D	458	MOV A,L
E033	D3F2	459	OUT CTR2P
E035	7C	460	MOV A,H
E036	D3F2	461	OUT CTR2P
		462 ;	
		463 ;	B. INITIALIZE RAM.
		464 ;	1. COMPUTE SIZE OF RAM MEMORY.
		465 ;	2. SET UP DEDICATED MEMORY LOCATIONS
		466 ;	USER I/O ENTRY POINTS (TOP OF MEMORY)
		467 ;	EXIT TEMPLATE
		468 ;	USER REGISTERS
		469 ;	USER INTERRUPT MASK
		470 ;	USER STACK
		471 ;	MONITOR STACK
		472 ;	RESTART ROUTINE JUMP ADDRESS
		473 ;	
E038	210000	474	LXI H,0 ; INITIAL VALUE H:=0, L:=0
		475 BS1:	
E03B	24	476	INR H ; INCREMENT BY 256 BYTE PAGES, I.E. 100H, 20CH, ..., F800H
E03C	7E	477	MOV A,H ; FETCH CONTENTS OF MEMORY
E03D	2F	478	CMA ; INVERT IT
E03E	77	479	MOV M,A ; ATTEMPT TO WRITE IT BACK INTO MEMORY
E03F	BE	480	CMP M ; IS LOCATION READ/WRITE, I.E. EXISTING RAM
E040	2F	481	CMA ; INVERT AGAIN BACK TO ORIGINAL VALUE
E041	77	482	MOV M,A ; WRITE ORIGINAL DATA VALUE BACK IN
E042	CA3BE8	483	JZ BS1 ; YES, CONTINUE (I.E. STILL CONTIGUOUS RAM)
E045	2B	484	DCX H ; OTHERWISE, IT'S LAST ADDRESS IN RAM
E046	3EE7	485	MVI A,FSTP ; LOAD FULL SYSTEM PAGE ADDRESS
E048	BC	486	CMP H ; TEST FOR FULL SYSTEM
E049	C24EE8	487	JNZ BS2 ; JUMP IF NOT FULL SYSTEM
E04C	26F7	488	MVI H,FSTP ; LOAD H WITH TOP PAGE ADDRESS
		489 BS2:	
E04E	220400	490	SHLD MEMTOP ; STORE TOP OF MEMORY
E051	01C8EA	491	LXI B,TDS ; MOVE EXIT TEMPLATE TO RAM
E054	69	492	MOV L,C

LOC	OBJ	SEQ	SOURCE STATEMENT
E855	F9	493	SPHL ; SET MONITOR'S STACK POINTER
		494	BS3:
E856	0A	495	LDAK B
E857	77	496	MOV H,A
E858	0C	497	INR C ; MOVE BOTH POINTERS
E859	2C	498	INR L
E85A	C256E8	499	JNZ BS3 ; END ON PAGE BOUNDARY
E85D	2ED1	500	MVI L,SLOC AND HMSK ; SET UP INITIAL VALUE FOR USER STACK
E85F	74	501	MOV H,H ; LOWER HALF OF STACK POINTER IS KNOWN
E860	35	502	DCR M ; MERELY SET UPPER HALF
		503	; TRAP TO MONITOR (AT LOCATIONS 0,1,2)
E861	3EC3	504	MVI A,(JMP RESTART)
E863	320000	505	STA RESET
E866	21D4FE	506	LXI H,RESTART ; SET UP RESTART 0 FOR BREAKPOINT
E869	220100	507	SHLD RESET+1 ; LOGIC
		508	;
		509	C. PROGRAM I/O DEVICES.
		510	1. BAUD RATE GENERATOR FOR CRT
		511	2. USART FOR CRT
		512	3. BAUD RATE GENERATOR FOR TTY
		513	4. USART FOR TTY
		514	;
E86C	3E76	515	MVI A,CTRIS OR MODE3 OR RLLM
E86E	D3F3	516	OUT ITCP
E870	212000	517	LXI H,B2400 ; CRT BAUD RATE
E873	7D	518	MOV A,L
E874	D3F1	519	OUT CTRIP
E876	7C	520	MOV A,H
E877	D3F1	521	OUT CTRIP
E879	3ECE	522	MVI A,ST2 OR R16X OR CL8
E87B	D3F7	523	OUT USCC
E87D	3E27	524	MVI A,TXEN OR DTR OR RXEN OR RTS
E87F	D3F7	525	OUT USCC
E881	3E36	526	MVI A,CTROS OR MODE3 OR RLLM
E883	D3F3	527	OUT ITCP
E885	21BA02	528	LXI H,B0110 ; TTY BAUD RATE
E888	7D	529	MOV A,L
E889	D3F0	530	OUT CTRDP
E88B	7C	531	MOV A,H
E88C	D3F0	532	OUT CTRDP
E88E	3ECE	533	MVI A,ST2 OR R16X OR CL8
E890	D3F5	534	OUT TTYC
E892	3E25	535	MVI A,TXEN OR RXEN OR RTS
E894	D3F5	536	OUT TTYC
		537	;
		538	D. DETERMINE IF INTEGRATED CONSOLE PRESENT
		539	;
E896	2EFF	540	MVI L,ITINO ; LOAD TIMEOUT CONSTANT
		541	BS4:
E898	DBC1	542	IN IDCS ; INPUT DBB STATUS
E89A	E607	543	ANI IBF OR OBF OR FD ; MASK OFF STATUS FLAGS
		544	; AND TEST FOR SLAVE PRESENCE
E89C	CAACE8	545	JZ BS5 ; JUMP IF INTEGRATED CONSOLE PRESENT
E89F	CD07EA	546	CALL BDLY ; DELAY 1 MS FOR ANY RESETS TO COMPLETE
E8A2	CD07EA	547	CALL BDLY ; DELAY 1 MS



LOC	OBJ	SEQ	SOURCE STATEMENT	
E8F2	05	603	DCR	B ; DECREMENT IOPB LENGTH COUNT
E8F3	C2EEE8	604	JNZ	NLP ; CONTINUE UNTIL ALL OF IOPB MOVED
E8F6	210010	605	LXI	H,1000H ; RELOAD POINTER TO IOPB
E8F9	7D	606	MOV	A,L ; A CONTAINS LSB OF IOPB ADDRESS
E8FA	D379	607	OUT	LOWH ; LOW(IOPB)
E8FC	7C	608	MOV	A,H ; A CONTAINS MSB OF IOPB ADDRESS
E8FD	D37A	609	OUT	HI ; HIGH(IOPB), START DISK I/O
		610	BS10:	
E8FF	D878	611	IN	DSTS ; WAIT FOR FDCC TO COMPLETE
E901	E604	612	ANI	DPCPL ; TEST FOR DISK COMPLETION
E903	CAFFE8	613	JZ	BS10
E906	F1	614	POP	PSW ; GET READY TO SET FLAG TO NEW VALUE
E907	AF	615	XRA	A ; SET A TO ZERO TO INDICATE DRIVE OTHER THAN INTEGRATED
		616		; FLOPPY WAS ACCESSED CORRECTLY
E908	F5	617	PUSH	PSW ; SAVE ON STACK
E909	C37AE9	618	JMP	BSX1 ; BYPASS INTEGRATED FLOPPY BOOT
		619		;
		620		; LOAD ISIS TO FROM INTEGRATED DISK IF AVAILABLE
		621		;
		622	BS11:	
E90C	2A0400	623	LHLD	MEMTOP ; LOAD TOP OF MEMORY PAGE ADDRESS
E90F	2ECC	624	MVI	L,BLOC+1 AND LBNK ; LOAD CONFIGURATION ADDRESS
E911	7E	625	MOV	A,H
E912	0F	626	RRC	
				; TEST FOR INTEGRATED CONSOLE PRESENT
E913	DA7AE9	627	JC	BSX1 ; JUMP IF IOC NOT PRESENT OR FUNCTIONAL
E916	061C	628	MVI	B,RDSTS ; LOAD FLOPPY DEVICE STATUS COMMAND
E918	CD21F8	629	CALL	IOCDP1 ; READ STATUS FROM I/O CONTROLLER
E918	E608	630	ANI	DPRT ; TEST FOR DRIVE PRESENT
E91D	CA7AE9	631	JZ	BSX1 ; JUMP IF NOT PRESENT
E920	061C	632	MVI	B,RDSTS ; LOAD FLOPPY DEVICE STATUS COMMAND
E922	CD21F8	633	CALL	IOCDP1 ; READ STATUS FROM I/O CONTROLLER
		634		; SECOND STATUS READ USED TO INSURE DRIVE READY
E925	0F	635	RRC	
				; TEST FOR DRIVE READY
E926	D27AE9	636	JNC	BSX1 ; JUMP IF DRIVE NOT READY
E929	F1	637	POP	PSW ; UNLOAD STACK
E92A	AF	638	XRA	A ; SET A TO 1 TO INDICATE
E92B	3C	639	INR	A ; INTEGRATED FLOPPY WAS ACCESSED
E92C	F5	640	PUSH	PSW ; SAVE ON STACK
E92D	2118EA	641	LXI	H,IOPB ; LOAD POINTER TO IOPB
E930	0615	642	MVI	B,WPBC ; LOAD WRITE IOPB COMMAND
E932	4E	643	MOV	C,H ; LOAD FIRST BYTE OF IOPB
E933	CD44F8	644	CALL	IOCDP2 ; SEND BYTE TO IOC
E936	1E04	645	MVI	E,PARML ; LOAD IOPB LENGTH REMAINING
E938	0616	646	MVI	B,WPBCC ; LOAD WRITE IOPB CONTINUE COMMAND
		647	BS12:	
E93A	23	648	INX	H ; MOVE POINTER TO NEXT BYTE OF IOPB
E93B	4E	649	MOV	C,H ; MOVE TO C
E93C	CD44F8	650	CALL	IOCDP2 ; SEND TO IOC
E93F	1D	651	DCR	E ; DECREMENT IOPB LENGTH
E940	C23AE9	652	JNZ	BS12 ; CONTINUE UNTIL ALL DATA TRANSMITTED
E943	061C	653	MVI	B,RDSTS ; LOAD DEVICE STATUS COMMAND
		654	BS13:	
E945	CD21F8	655	CALL	IOCDP1 ; READ STATUS FROM IOC
E948	E604	656	ANI	DPCPL ; TEST FOR OPERATION COMPLETE
E94A	CA45E9	657	JZ	BS13 ; LOOP UNTIL DONE

LOC	OBJ	SEQ	SOURCE STATEMENT
E94D	061B	658	MVI B,RRSTS ; LOAD RESULT STATUS COMMAND
E94F	CD21F8	659	CALL IOCDPI ; READ RESULT STATUS FROM IOC
E952	32FE2F	660	STA TRKO-2 ; SAVE FOR TEST LATER
E955	B7	661	ORA A ; SET CONDITION CODES
E956	C27AE9	662	JNZ BSX1 ; JUMP IF DISK OPERATION UNSUCCESSFUL
E959	210030	663	LXI H,TRKO ; LOAD POINTER TO DISK DESTINATION ADDRESS
E95C	11000D	664	LXI D,TRKL ; LOAD TRACK LENGTH
E95F	0619	665	MVI B,RDBC ; LOAD DISK READ DATA COMMAND
E961	CD21F8	666	CALL IOCDPI ; LOAD DATA FROM IOC
E964	77	667	MOV M,A ; MOVE TO MEMORY
E965	1B	668	DCX D ; DECREMENT LENGTH
E966	23	669	INX H ; MOVE POINTER TO NEXT LOCATION
		670	BS14:
E967	DBC1	671	IN IOCS ; INPUT DBB STATUS
E969	E607	672	ANI IBF OR OBF OR FD ; MASK OFF STATUS FLAGS
E96B	FE01	673	CPI DBF ; TEST FOR DATA IN BUFFER
E96D	C267E9	674	JNZ BS14 ; JUMP IF NO DATA
E970	DBC0	675	IN IOCI ; INPUT DATA FROM DBB
E972	77	676	MOV M,A ; MOVE TO MEMORY
E973	23	677	INX H ; MOVE POINTER TO NEXT LOCATION
E974	1B	678	DCX D ; DECREMENT LENGTH
E975	7A	679	MOV A,D ; LOAD D FOLLOWED BY E
E976	B3	680	ORA E ;
E977	C267E9	681	JNZ BS14 ; CONTINUE UNTIL DONE
		682 ;	
		683 ;	F. DETERMINE COLD START LOCAL CONSOLE.
		684 ;	
		685 ;	-----
		686 ;	CONSOLE IS EITHER INTEGRATED CRT, SERIAL CRT, OR TTY
		687	BSX1:
E97A	2A0400	688	LHLD MENTOP ; LOAD TOP OF MEMORY PAGE ADDRESS
E97D	ZECC	689	MVI L,BLOC+1 AND LBMK ; LOAD CONFIGURATION ADDRESS
E97F	7E	690	MOV A,M ; LOAD INTEGRATED CONSOLE FLAG
E980	0F	691	RRC ; TEST FOR INTEGRATED CONSOLE PRESENT
E981	DA90E9	692	JC BSX2 ; JUMP IF INTEGRATED CONSOLE NOT PRESENT
E984	0613	693	MVI B,KSTS ; LOAD KEYBOARD STATUS COMMAND
E986	CD21F8	694	CALL IOCDPI ; READ STATUS FROM IOC
E989	0F	695	RRC ; TEST FOR KEYBOARD PRESENT
E98A	0F	696	RRC ;
E98B	1641	697	MVI D,ICFG ; LOAD INITIAL CONFIGURATION
E98D	DAB0E9	698	JC BSX5 ; JUMP IF KEYBOARD PRESENT
		699 ;	-----
		700 ;	CONSOLE IS EITHER SERIAL CRT OR TTY
		701	BSX2:
E990	AF	702	XRA A ; ZERO A
E991	57	703	MOV D,A ; D CONTAINS OH, I.E.C=T,R=T,P=T,L=T
E992	DBF5	704	IN TTYS ; GET TTY STATUS
E994	E602	705	ANI RRDY ; IS IT READY?
E996	CA9EE9	706	JZ BSX3 ; JUMP IF TTY NOT READY
E999	DBF4	707	IN TTYI ; OTHERWISE GET CHARACTER FROM TTY
E99B	C3A9E9	708	JMP BSX4 ;
		709	BSX3:
E99E	1641	710	MVI D,ICFG ; LOAD INITIAL CONFIGURATION STATUS
E9A0	DBF7	711	IN USCS ; GET SERIAL CRT STATUS
E9A2	E602	712	ANI RRDY ; IS IT READY?



LOC	OBJ	SEQ	SOURCE STATEMENT
		768	BSX9:
E9F1	211FEA	769	LXI    H,VERS            ; HL POINTS TO ADDRESS OF SIGN-ON MESSAGE
E9F4	061B	770	MVI    B,LVER           ; B CONTAINS LENGTH OF MESSAGE
E9F6	C00EEA	771	CALL   PRM              ; PRINT SIGN-ON MESSAGE
		772	;
		773	J.     BOOTSTRAP ALL DONE, SO BRANCH TO MONITOR.
		774	;
E9F9	C300F8	775	JMP    BEGIN            ; AT THIS POINT, INTERRUPTS ARE DISABLED
		776	;
		777	K.     PRINT DISK ERROR MESSAGE
		778	;
		779	BSX10:
E9FC	213AEA	780	LXI    H,ERMSG           ; HL POINTS TO ADDRESS OF DISK ERROR MESSAGE
E9FF	060E	781	MVI    B,LERM           ; B CONTAINS LENGTH OF MESSAGE
EA01	C00EEA	782	CALL   PRM              ; PRINT SIGN-ON MESSAGE
EA04	C3F1E9	783	JMP    BSX9             ; PRINT MESSAGE
		784	;-*****-
		785	;
		786	BDLY - BOOTSTRAP DELAY 1 MS SUBROUTINE
		787	;
		788	BDLY:
EA07	0E70	789	MVI    C,ONEHS          ; LOAD 1 MS. CONSTANT
		790	BDLY1:
EA09	0D	791	DCR    C                ; DECREMENT COUNTER
EA0A	C209EA	792	JNZ    BDLY1            ; JUMP IF NOT EXPIRED
EA0D	C9	793	RET
		794	;-*****-
		795	;
		796	PRM - PRT SUBROUTINE FOR SIGN-ON MESSAGES
		797	;
		798	PRM:
EA0E	4E	799	MOV    C,M              ; C CONTAINS A CHARACTER FROM THE MESSAGE
EA0F	C009F8	800	CALL   COP              ; PRINT ON CONSOLE
EA12	23	801	INX    H
EA13	05	802	DCR    B
EA14	C20EEA	803	JNZ    PRM              ; KEEP LOOPING UNTIL ENTIRE MESSAGE IS OUTPUT
EA17	C9	804	RET
		805	;-*****-
		806	;
		807	DISK I/O PARAMETER BLOCK
		808	;
		809	IOPB:
EA18	80	810	DB     80H              ; IOCW, NO UPDATE BIT SET
EA19	04	811	DB     04H              ; I/O INSTRUCTION, READ DISK 0
EA1A	1A	812	DB     26               ; READ 26 SECTORS
EA1B	00	813	DB     0                ; TRACK 0
EA1C	01	814	DB     1                ; SECTOR 1
EA1D	0030	815	DW     TRKO             ; LOAD ADDRESS
		816	;
		817	MONITOR SIGN-ON MESSAGE
		818	;
EA1F	0D	819	VERS: DB     CR,LF,'SERIES II MONITOR, V'
EA20	0A		
EA21	53455249		
EA25	45532049		

LOC	OBJ	SEQ	SOURCE STATEMENT
EA29	49204D4F		
EA2D	4E49544F		
EA31	522C2056		
EA35	31	820	DB VER/10+'0',',',',VER MOD 10+'0'
EA36	2E		
EA37	32		
EA38	0D	821	DB CR,LF
EA39	0A		
001B		822	LVER EQU \$-VERS ; LENGTH OF SIGN-ON MESSAGE
		823	;
		824	MONITOR ERROR SIGN-ON MESSAGE
		825	;
EA3A	0D	826	ERMSC: DB CR,LF,'DISK ERROR',CR,LF
EA3B	0A		
EA3C	4449534B		
EA40	20455252		
EA44	4F52		
EA46	0D		
EA47	0A		
00DE		827	LERM EQU \$-ERMSC ; LENGTH OF ERROR SIGN-ON MESSAGE
EA4B	06	828	BTCKSM: DB DA6H ; BOOT CHKSUMS TO 55H
		829	*****
		830	;
		831	EXIT CODE TEMPLATE, TO BE EXECUTED IN RAM
		832	THIS CODE IS ORIGINATED SO AS TO BE ALIGNED
		833	AGAINST THE TOP OF A PAGE (1 PAGE = 256 BYTES)
		834	;
EACB		835	ORG BBASE + 02CBH
		836	TDS: ; BASE OF MONITOR WORK STACK
EAC0		837	USER EQU TDS-8 ; BASE OF DEFAULT USER WORK STACK
EAC8	EE	838	ELOC: DB DEEH ; E REGISTER STORAGE
EAC9	DD	839	DLOC: DB ODDH ; D REGISTER
EACA	CC	840	CLOC: DB CCCH ; C REGISTER
EACB	BB	841	BLOC: DB OBBH ; B REGISTER
EACC	00	842	DB 0 ; CONFIGURATION BYTE
		843	;
		844	BIT 0 = 0 IF INTEGRATED CRT IS PRESENT = 1 IF INTEGRATED CRT NOT PRESENT
EACD	FE	845	ILOC: DB NOT INTO ; INTERRUPT MASK
EACE	FF	846	FLOC: DB DFFH ; CPU FLAGS
EACF	AA	847	ALOC: DB DAAH ; A REGISTER
EAD0	CD	848	DB USER AND HNSK ; LOW(SP)
EAD1	00	849	SLOC: DB 0 ; HIGH(SP)
		850	;
		851	EXIT: ; MONITOR STACK ORIGIN
EAD2	F3	852	DI ; DISABLE INTERRUPTS TO PROTECT THIS SEQUENCE
EAD3	D1	853	POP D ; RESTORE D,E
EAD4	C1	854	POP B ; RESTORE B,C
EAD5	F1	855	POP PSW
EAD6	D3FC	856	OUT SOCP1
EAD8	F1	857	POP PSW ; RESTORE A AND FLAGS
EAD9	E1	858	POP H ; RESTORE ORIGINAL STACK VALUE
EADA	F9	859	SPHL
EADB	213412	860	LXI H,1234H ; RESTORE H,L; 1234H IS FILLER WHICH WILL BE
		861	;
		862	LLOC EQU \$-2 ; OVERWRITTEN BY RESTART ROUTINE





```

LOC OBJ      SEQ      SOURCE STATEMENT
          918 ;*****
          919 ;*****
          920 ;*****
          921 ;***
          922 ;***          START OF MONITOR PROPER          ***
          923 ;***
          924 ;*****
          925 ;*****
          926 ;*****
F800      927 BASE   SET   0F800H          ; BASE ADDRESS OF MONITOR
F800      928 ORG    BASE          ; TOP 2K OF 64K ADDRESS SPACE
          929 ;-----
          930 ;
          931 ; BRANCH TABLE FOR I/O SYSTEM (EXTERNAL I/O ENTRY POINTS)
          932 ;
          933 ; THE MONITOR IS ENTERED AT ENTRY POINT 'BEGIN' VIA A JUMP FROM THE BOOTSTRAP;
          934 ; THIS IN TURN LEADS TO A JUMP TO ENTRY POINT 'START' THE OTHER ENTRIES
          935 ; IN THIS "TABLE" ARE EXTERNAL I/O ENTRY POINTS KNOWN TO THE USER PLUS
          936 ; THE DATE, VERSION, AND COPYRIGHT STAMPS.
          937 BEGIN:
F800 C351F8  938      JMP     START0          ; RESET ENTRY POINT
F803 C3BEF8  939      JMP     CI              ; LOCAL CONSOLE INPUT
F806 C30FFC  940      JMP     RI              ; READER INPUT
F809 C39FFC  941      JMP     CO              ; LOCAL CONSOLE OUTPUT
F80C C3E9FC  942      JMP     PD              ; PUNCH OUTPUT
F80F C31EFD  943      JMP     LD              ; LIST OUTPUT
F812 C344FD  944      JMP     CSTS           ; LOCAL CONSOLE INPUT STATUS
F815 C383FD  945      JMP     IOCHK          ; I/O SYSTEM STATUS
F818 C387FD  946      JMP     IOSET          ; SET I/O CONFIGURATION
F81B C38CFD  947      JMP     MEMCHK         ; COMPUTE SIZE OF MEMORY
F81E C394FD  948      JMP     IODEF          ; DEFINE USER I/O ENTRY POINTS
F821 C37FFF  949      JMP     IOCDR1         ; IOC INPUT
F824 0104   950      DB     DATE            ; DATE STAMP FOR MONITOR ROM
F826 C3ADF8  951      JMP     UI              ; UPP INPUT
F829 C3BEFD  952      JMP     UO              ; UPP OUTPUT
F82C C3CEFD  953      JMP     UPPS           ; UPP STATUS
F82F 12     954      DB     VERH            ; VERSION STAMP FOR MONITOR ROM
F830 28432949 955      DB     '(C)INTEL CORP1978' ; COPYRIGHT NOTICE IN ASCII REP
F834 4E54454C
F838 20434F52
F83C 50313937
F840 38
F841 C3A6FF  956      JMP     IOCCOM         ; IOCCOM ENTRY POINT
F844 C394FF  957      JMP     IOCDR2         ; IOC OUTPUT
          958 ;-----
          959 ;
          960 ; 'ERROR' - ENTERED VIA JUMP FROM VARIOUS ROUTINES WHEN AN ERROR IS DETECTED
          961 ; PROCESS: ABNORMAL EXIT FOR ALL MONITOR ERROR CONDITIONS. BECAUSE OF THE
          962 ; UNKNOWN STATE OF THE MONITOR AS A RESULT OF A COMMAND OR DATA ERROR,
          963 ; THE VALUE OF THE MONITOR STACK POINTER IS REINITIALIZED AND
          964 ; EXECUTION CONTINUES TO THE MAIN COMMAND LOOP.
          965 ; INPUT: HENTOP, TOS
          966 ; OUTPUT: SP POINTS TO BASE OF MONITOR STACK IN TOP PAGE OF CONTIGUOUS RAM
          967 ; MODIFIED: H,L, SP
          968 ; STACK USAGE:
    
```

```

LOC  OBJ          SER          SOURCE STATEMENT
969 ;
970 ; REGISTER USAGE
971 ; X = MODIFIED BY THIS ROUTINE, CONTENTS UNDEFINED.
972 ; S = SET BY THIS ROUTINE, RETURNED AS A RESULT.
973 ; U = USED AS INPUT.
974 ;     A -
975 ;     B -                   C - S
976 ;     D -                   E -
977 ;     H - X                 L - X
978 ;     CARRY - X            ZERO - X
979 ;     SIGN - X             PARITY - X
980 ;     SP - S                PC -
981 ;     STACK USAGE: 2 BYTES
982 ERROR:
F847 2A0400   983     LHLD   MEMTOP           ; M POINTS TO TOP PAGE OF MEMORY
F84A 2EC8     984     MVI     L,TOS AND OFFH   ; L POINTS TO BASE OF STACK WITHIN THAT PAGE
F84C F9       985     SPHL                   ; SP NOW POINTS TO BASE OF MONITOR STACK
F84D CDDEFC   986     CALL   COMC            ; OUTPUT THE ERROR INDICATOR CHAR '#'
F850 23       987     DB     '#'
988                               ; FALL THROUGH TO MAIN COMMAND LOOP
989 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*-----*
990 ;
991 ; MAIN COMMAND LOOP.
992 ;
993 ; THIS LOOP IS THE STARTING POINT OF ALL COMMAND SEQUENCES.
994 ; IT IS ENTERED VIA A JUMP FROM THE BEGINNING OF THE MONITOR PROPER CODE,
995 ; A FALL THROUGH FROM THE ERROR ROUTINE, OR A RETURN FROM A MONITOR COMMAND
996 ; ROUTINE.
997 ; IN THIS CODE INTERRUPTS ARE ENABLED AND A CARRIAGE RETURN
998 ; AND LINE FEED ARE TYPED ALONG WITH THE PROMPT CHARACTER, ' '
999 ; WHEN A CHARACTER IS ENTERED FROM THE LOCAL CONSOLE KEYBOARD, IT
1000 ; IS CHECKED FOR VALIDITY, THEN A BRANCH TO THE PROPER
1001 STARTO:
F851 3E04     1002     MVI     A,BTDCOF           ; DISABLE BOOT, I.E. SWITCH BOOT PROM
F853 D3FF     1003     OUT     CPUC            ;     OUT OF ADDRESSABLE MEMORY SPACE
1004 START:
F855 FB       1005     EI                    ; ENABLE INTERRUPTS
F856 CDFFD    1006     CALL   CRLF            ; TYPE <CR>,<LF>
F859 CDDEFC   1007     CALL   COMC            ; OUTPUT A PERIOD
F85C 2E       1008     DB     '#'
F85D CD61FF   1009     CALL   TI                ; GET A CHARACTER, ECHO IT.
F860 FE0D     1010     CPI     CR            ; IS IT A CARRIAGE RETURN?
F862 CA55F8   1011     JZ     START            ; JUMP IF IT IS
F865 D641     1012     SUI     'A'           ; OTHERWISE TEST FOR A-Z (VALID COMMAND RANGE)
F867 FA47F8   1013     JM     ERROR            ; LESS THAN A, NOT A VALID COMMAND
F86A 0E02     1014     MVI     C,2           ; ASSUME THE COMMAND NEEDS 2 PARAMETERS
F86C 1155F8   1015     LXI     D,START          ; SET UP PSEUDO RETURN ADDRESS TO SIMULATE
F86F 05       1016     PUSH   D            ;     EFFECT OF A CALL. COMMANDS WHICH PERFORM
1017                               ;     A RETURN WILL CAUSE THE STACK TO BE
1018                               ;     POPPED, THUS RETURNING TO ENTRY POINT
1019                               ;     START. THE 'G' COMMAND, HOWEVER, WIPES
1020                               ;     OUT THIS ADDRESS WITH ANOTHER ADDRESS
1021                               ;     OF ITS OWN CHOOSING (I.E. USER'S PC).
F870 2182F8   1022     LXI     H,CTBL           ; LOAD POINTER TO PROCESSING ROUTINE PTRS
F873 FE1A     1023     CPI     LCT            ; TEST FOR OVERRUN

```

```

LOC  OBJ          SEQ      SOURCE STATEMENT
F875 F247F8      1024      JP      ERROR          ; IF SO, THEN ERROR
F878 5F          1025      MOV     E,A           ; OTHERWISE, MOVE INDEX TO DE
F879 1600        1026      MVI    D,0
F87B 19          1027      DAD    D
F87C 19          1028      DAD    D              ; HL := CTBLBASE + (2 * INDEX); HL NOW POINTS
1029      ; TO PROPER COMMAND IN COMMAND BRANCH TABLE
F87D 7E          1030      MOV     A,M          ; GET LSB OF BRANCH LOCATION
F87E 23          1031      INX    H
F87F 66          1032      MOV     H,M          ; GET MSB OF BRANCH LOCATION
F880 6F          1033      MOV     L,A          ; HL POINTS TO ADDRESS OF COMMAND CODE
F881 E9          1034      PCHL                   ; TAKE THE BRANCH
1035 ; *-----*
1036 ;
1037 ; COMMAND BRANCH TABLE.
1038 ;
1039 ; THIS TABLE CONTAINS THE ADDRESSES OF THE ENTRY POINTS OF
1040 ; ALL THE COMMAND PROCESSING ROUTINES. IT IS ENTERED FROM THE MAIN
1041 ; COMMAND LOOP. NOTE THAT AN ENTRY TO 'ERROR'
1042 ; IS AN ERROR CONDITION, I.E., NO COMMAND CORRESPONDING TO THAT
1043 ; CHARACTER EXISTS.
1044 CTBL:
F882 86F8        1045      DW     ASSIGN        ; A - ASSIGN I/O UNITS
F884 47F8        1046      DW     ERROR         ; B -
F886 47F8        1047      DW     ERROR         ; C -
F888 33F9        1048      DW     DISP          ; D - DISPLAY RAM MEMORY
F88A 5FF9        1049      DW     EOF           ; E - ENDFILE A HEXADECIMAL FILE
F88C 7DF9        1050      DW     FILL          ; F - FILL MEMORY
F88E 8CF9        1051      DW     GOTO          ; G - GO TO MEMORY ADDRESS
F890 05F9        1052      DW     HEXN          ; H - HEXADECIMAL SUM AND DIFFERENCE
F892 47F8        1053      DW     ERROR         ; I -
F894 47F8        1054      DW     ERROR         ; J -
F896 47F8        1055      DW     ERROR         ; K -
F898 47F8        1056      DW     ERROR         ; L -
F89A F0F9        1057      DW     MOVE          ; M - MOVE MEMORY
F89C 01FA        1058      DW     NULL          ; N - PUNCH NULLS FOR LEADER ON PAPER TAPE
F89E 47F8        1059      DW     ERROR         ; O -
F8A0 47F8        1060      DW     ERROR         ; P -
F8A2 14FA        1061      DW     QUERY         ; Q - QUERY I/O SYSTEM STATUS
F8A4 52FA        1062      DW     READ          ; R - READ HEXADECIMAL PAPER TAPE FILE
F8A6 8FFA        1063      DW     SUBS          ; S - SUBSTITUTE MEMORY
F8A8 47F8        1064      DW     ERROR         ; T -
F8AA 47F8        1065      DW     ERROR         ; U -
F8AC 47F8        1066      DW     ERROR         ; V -
F8AE DDFA        1067      DW     WRITE         ; W - WRITE FILE TO PAPER TAPE IN HEX FORMAT
F8B0 26FB        1068      DW     X             ; X - EXAMINE AND MODIFY REGISTERS
F8B2 47F8        1069      DW     ERROR         ; Y -
F8B4 A6FB        1070      DW     Z             ; Z - INVOKE THE DIAGNOSTIC PROGRAM
1071 LCT          EQU     (CTBL)/2      ; LCT = NUMBER OF 16-BIT ENTRIES IN TABLE
1072 ; *-----*
1073 ;
1074 ; 'A' COMMAND - ASSIGN I/O DEVICE
1075 ;
1076 ; THIS ROUTINE MAPS SYMBOLIC DEVICE IDENTIFIERS TO BITS
1077 ; IN THE I/O STATUS BYTE (IOBYT) TO ALLOW FOR LOCAL CONSOLE
1078 ; MODIFICATION OF SYSTEM I/O CONFIGURATION.

```

LOC	OBJ	SEQ	SOURCE STATEMENT
		1079	ASSIGN:
F8B6	CD61FF	1080	CALL TI ; GET LOGICAL DEVICE CHARACTER (C,R,P,L)
F8B9	2103F9	1081	LXI H,LTBL ; ADDRESS OF MASTER TABLE
F8BC	0E04	1082	MVI C,4 ; MAXIMUM OF 4 ENTRIES
		1083	;
		1084	ASO: ; HL POINTS TO IDENTIFYING CHARACTER IN LTBL
F8BE	BE	1085	CMP H ; DOES A-REG CONTAIN C,R,P, OR L?
F8BF	23	1086	INX H ; HL POINTS TO CORRESPONDING DEVICE MASK
F8C0	CACDF8	1087	JZ AS1 ; YES IT DOES
F8C3	23	1088	INX H
F8C4	23	1089	INX H
F8C5	23	1090	INX H ; HL POINTS TO NEXT 4-BYTE ENTRY IN LTBL
F8C6	0D	1091	DCR C ; DECREMENT LOOP COUNT
F8C7	C2BEF8	1092	JNZ ASO ; TRY NEXT ENTRY
F8CA	C347F8	1093	JMP ERROR ; NO MATCH, ERROR
		1094	;
		1095	AS1: ; USER HAS SPECIFIED A VALID LOGICAL DEVICE
F8CD	46	1096	MOV B,M ; B := LOGICAL DEVICE MASK
F8CE	23	1097	INX H ; HL CONTAINS SUBORDINATE PHYS.DEV.TBL.ADDRESS
F8CF	5E	1098	MOV E,M ; E CONTAINS LSB OF PDT ADDRESS
F8D0	23	1099	INX H
F8D1	56	1100	MOV D,M ; D CONTAINS MSB OF PDT ADDRESS
F8D2	EB	1101	XCHG ; HL POINTS TO I/O SYSTEM PHYSICAL DEVICE
		1102	TABLE (I.E. ACT,ART,APT, OR ALT)
		1103	;
		1104	ALUP1: ; SCAN INPUT UNTIL '='
F8D3	CD61FF	1105	CALL TI
F8D6	FE3D	1106	CPI '='
F8D8	C2D3F8	1107	JNZ ALUP1
		1108	;
		1109	ALUP2: ; SCAN INPUT WHILE ' ' (BLANK)
F8DB	CD61FF	1110	CALL TI
F8DE	FE2D	1111	CPI ' '
F8E0	CADBF8	1112	JZ ALUP2
		1113	;
F8E3	0E04	1114	MVI C,4 ; SET TABLE LENGTH
		1115	AS2: ; INDEX THROUGH PHYSICAL UNIT TABLE
F8E5	BE	1116	CMP H ; COMPARE DEVICE CHAR WITH LEGAL VALUES
F8E6	23	1117	INX H ; HL CONTAINS DEVICE SELECT BIT PATTERN
F8E7	CAF2F8	1118	JZ AS3 ; USER HAS SPECIFIED A VALID PHYS.DEVICE ASSIGNMNT
F8EA	23	1119	INX H ; HL POINTS TO NEXT ENTRY WITHIN THE TABLE
F8EB	0D	1120	DCR C
F8EC	C2E5F8	1121	JNZ AS2 ; CONTINUE LOOKUP
F8EF	C347F8	1122	JMP ERROR ; ERROR RETURN
		1123	;
		1124	AS3:
		1125	ALUP3: ; SCAN INPUT UNTIL (CR)
F8F2	CD61FF	1126	CALL TI
F8F5	FE0D	1127	CPI CR
F8F7	C2F2F8	1128	JNZ ALUP3
F8FA	3A0300	1129	LDA IOBYT ; GET I/O STATUS
F8FD	80	1130	ANA B ; B CONTAINS LOG DEV MASK. CLEAR OUT THE
		1131	APPROPRIATE FIELD IN IOBYT BECAUSE WE ARE
		1132	GOING TO CHANGE IT.
F8FE	86	1133	ORA H ; PUT IN THE NEW STATUS FIELD

LOC	OBJ	SEQ	SOURCE STATEMENT
F8FF	320300	1134	STA IOBYT ; RETURN IT TO MEMORY
F902	C9	1135	RET ; RETURN CONTROL TO MAIN COMMAND LOOP
		1136 ;	
		1137 ;	MASTER I/O DEVICE TABLE
		1138 ;	4 BYTES/ENTRY
		1139 ;	
		1140 ;	BYTE 0 = IDENTIFYING CHARACTER
		1141 ;	BYTE 1 = LOGICAL DEVICE MASK
		1142 ;	BYTES 2,3 = ADDRESS OF SUBORDINATE PHYSICAL DEVICE TABLE
		1143 ;	
		1144	LTBL:
F903	43	1145	DB 'C',CNSK
F904	FC		
F905	13F9	1146	DW ACT
F907	52	1147	DB 'R',RNSK
F908	F3		
F909	1BF9	1148	DW ART
F90B	50	1149	DB 'P',PNSK
F90C	CF		
F90D	23F9	1150	DW APT
F90F	4C	1151	DB 'L',LNSK
F910	3F		
F911	2BF9	1152	DW ALT
		1153 ;	
		1154 ;	I/O SYSTEM PHYSICAL DEVICE TABLES
		1155 ;	2 BYTES/ENTRY
		1156 ;	
		1157 ;	BYTE 0 = IDENTIFYING CHARACTER
		1158 ;	BYTE 1 = DEVICE SELECT BIT PATTERN
		1159 ;	
		1160	ACT:
F913	54	1161	DB 'T',CTTY ; LOCAL CONSOLE = TTY
F914	00		
F915	43	1162	DB 'C',CCRT ; LOCAL CONSOLE = CRT
F916	01		
F917	42	1163	DB 'B',BATCH ; BATCH NODE LOCAL CONSOLE = READ,LIST
F918	02		
F919	31	1164	DB '1',CUSE ; USER DEFINED LOCAL CONSOLE DEVICE
F91A	03		
		1165	ART:
F91B	54	1166	DB 'T',RTTY ; READER = TTY
F91C	00		
F91D	50	1167	DB 'P',RPTR ; READER = PTR
F91E	04		
F91F	31	1168	DB '1',RUSE1 ; USER DEFINED READER DEVICE 1
F920	08		
F921	32	1169	DB '2',RUSE2 ; USER DEFINED READER DEVICE 2
F922	0C		
		1170	APT:
F923	54	1171	DB 'T',PTTY ; PUNCH = TTY
F924	00		
F925	50	1172	DB 'P',PPTP ; PUNCH = PTP
F926	10		
F927	31	1173	DB '1',PUSE1 ; USER DEFINED PUNCH DEVICE 1
F928	20		

LOC	OBJ	SEQ	SOURCE STATEMENT
F929	32	1174	DB '2',PUSE2 ; USER DEFINED PUNCH DEVICE 2
F92A	30		
		1175	ALT:
F92B	54	1176	DB 'T',LTTY ; LIST = TTY
F92C	00		
F92D	43	1177	DB 'C',LCRT ; LIST = CRT
F92E	40		
F92F	4C	1178	DB 'L',LLPT ; LIST = LPT
F930	80		
F931	31	1179	DB '1',LUSE ; USER DEFINED LIST DEVICE
F932	C0		
		1180	;-*****-
		1181	;
		1182	; 'D' COMMAND - DISPLAY CONTENTS OF MEMORY ON LIST DEVICE
		1183	;
		1184	; THIS ROUTINE EXPECTS TWO HEXADECIMAL PARAMETERS SPECIFYING
		1185	; THE BOUNDS OF A MEMORY AREA TO BE DISPLAYED ON THE
		1186	; LIST DEVICE. THE MEMORY AREA IS DISPLAYED 16 BYTES
		1187	; PER LINE, WITH THE MEMORY ADDRESS OF THE FIRST BYTE
		1188	; PRINTED FOR REFERENCE. ALL LINES ARE BLOCKED INTO INTEGRAL
		1189	; MULTIPLES OF 16 FOR CLARITY, SO THE FIRST AND LAST LINES MAY
		1190	; BE LESS THAN 16 BYTES IN ORDER TO SYNCHRONIZE THE DISPLAY.
		1191	DISP:
F933	CD39FE	1192	CALL EXPR ; GET TWO ADDRESSES
F936	D1	1193	POP D ; GET HIGH ADDRESS
F937	E1	1194	POP H ; GET LOW ADDRESS
		1195	D10:
F938	CD6AFE	1196	CALL LCRLF ; PRINT CR,LF
F93B	CD07FE	1197	CALL DADR ; PRINT MEMORY ADDRESS
		1198	D11:
F93E	0E20	1199	MVI C, ' '
F940	CD14FD	1200	CALL LOM ; PRINT SPACE
F943	7E	1201	MOV A,H
F944	CD0CFE	1202	CALL DBYTE ; PRINT DATA
F947	CD4CFE	1203	CALL HILO ; TEST FOR COMPLETION
F94A	DA56F9	1204	JC D12 ; RETURN TO MAIN LOOP
F94D	7D	1205	MOV A,L
F94E	E60F	1206	ANI OFH ; PRINT CR,LF,ADDRESS ON MULTIPLE OF 16
F950	C23EF9	1207	JNZ D11
F953	C338F9	1208	JMP D10
		1209	D12:
F956	CD6AFE	1210	CALL LCRLF ; WRITE CR,LF
F959	0E00	1211	MVI C,0
F95B	CD14FD	1212	CALL LOM ; WRITE A NULL TO TRIGGER CLOSE
F95E	C9	1213	RET
		1214	;-*****-
		1215	;
		1216	; 'E' COMMAND - PUNCH HEXADECIMAL END-OF-FILE
		1217	;
		1218	; THIS ROUTINE PRODUCES A TERMINATION RECORD WHICH PROPERLY
		1219	; COMPLETES A HEXADECIMAL FILE CREATED BY 'W' COMMANDS.
		1220	; IT EXPECTS ONE HEXADECIMAL PARAMETER, WHICH IT INTERPRETS AS THE
		1221	; START ADDRESS TO BE LOADED INTO THE USER'S PROGRAM COUNTER (LOCATED
		1222	; IN EXIT TEMPLATE) ON A SUBSEQUENT 'R' COMMAND; THIS START ADDRESS
		1223	; WILL REPLACE THE STORED VALUE OF THE USER'S PROGRAM COUNTER ONLY

```

LOC  OBJ      SEQ      SOURCE STATEMENT
1224 ; IF THE START ADDRESS IS NONZERO.
1225 ;
1226 EOF:
F95F 0D      1227      DCR      C          ; C:=1; GET ONE PARAMETER
F960 CD39FE  1228      CALL     EXPR       ; PUT (START ADDRESS) ON TOP OF STACK
F963 CDE5FC  1229      CALL     POC        ; OUTPUT RECORD MARK (':')
F966 3A      1230      DB      ':'
F967 AF      1231      XRA     A          ; ZERO CHECKSUM
F968 57      1232      MOV     D,A        ; D := 0; A := 0
F969 CDAFFE  1233      CALL     PBYTE     ; OUTPUT A RECORD LENGTH OF ZERO
F96C E1      1234      POP     H          ; RETRIEVE START ADDRESS
F96D CDAAFE  1235      CALL     PADR      ; OUTPUT IT AS THE LOAD ADDRESS
F970 3E01    1236      MVI     A,1        ; RECORD TYPE = 1
F972 CDAFFE  1237      CALL     PBYTE     ; OUTPUT RECORD TYPE
F975 AF      1238      XRA     A          ; A := 0
F976 92      1239      SUB     D          ; D CONTAINS RUNNING CHECKSUM
F977 CDAFFE  1240      CALL     PBYTE     ; OUTPUT CHECKSUM := -D
F97A C309FA  1241      JMP     HLD        ; PUNCH TRAILER AND RETURN
1242 ;-----
1243 ;
1244 ; 'F' COMMAND - FILL RAM WITH 8-BIT CONSTANT
1245 ;
1246 ; THIS ROUTINE EXPECTS THREE HEXADECIMAL PARAMETERS, THE
1247 ; FIRST AND SECOND (16 BITS) ARE INTERPRETED AS THE BOUNDS
1248 ; OF A MEMORY AREA TO BE INITIALIZED TO A CONSTANT VALUE,
1249 ; THE THIRD PARAMETER (8 BITS) IS THAT VALUE.
1250 FILL:
F97D 0C      1251      INR     C          ; C:=3; GET 3 PARAMETERS
F97E CD39FE  1252      CALL     EXPR       ;
F981 C1      1253      POP     B          ; C := 8-BIT CONSTANT
F982 01      1254      POP     D          ; DE := HIGH ADDRESS
F983 E1      1255      POP     H          ; HL := LOW ADDRESS
1256 FID:
F984 71      1257      MOV     M,C        ; STORE CONSTANT IN MEMORY
F985 CD4CFE  1258      CALL     HLD        ; TEST FOR COMPLETION
F988 0284F9  1259      JNC     FID        ; CONTINUE LOOPING
F98B C9      1260      RET          ; GO BACK TO START
1261 ;-----
1262 ;
1263 ; 'G' COMMAND - GO TO (ADDRESS), OPTIONALLY SET BREAKPOINT(S)
1264 ;
1265 ; THE G COMMAND IS USED FOR TRANSFERRING CONTROL FROM THE
1266 ; MONITOR TO A USER PROGRAM. IT HAS SEVERAL MODES OF
1267 ; OPERATION.
1268 ; IF ONE HEXADECIMAL PARAMETER IS ENTERED, IT IS INTERPRETED
1269 ; AS THE ENTRY POINT OF THE USER PROGRAM AND A TRANSFER TO
1270 ; THAT LOCATION IS EXECUTED.
1271 ; IF ADDITIONAL (UP TO 2) PARAMETERS ARE ENTERED, THESE ARE
1272 ; CONSIDERED 'BREAKPOINTS', I.E., LOCATIONS WHERE
1273 ; CONTROL IS TO BE RETURNED TO THE MONITOR WHEN THEY ARE
1274 ; ENCOUNTERED IN COURSE OF EXECUTING THE USER PROGRAM.
1275 ; IF THE FIRST PARAMETER IS NOT ENTERED, THE STORED VALUE
1276 ; OF THE USER'S PROGRAM COUNTER (REGISTER P) IS USED AS
1277 ; THE USER PROGRAM ENTRY POINT.
1278 ;

```





```

LOC  OBJ          SEQ      SOURCE STATEMENT

F9C2  C1          1334      POP      B              ; GET A TRAP (BREAKPOINT) ADDRESS
F9C3  71          1335      MOV      H,C           ; STORE LSB OF TRAP ADDRESS INTO TRAP AREA
F9C4  23          1336      INX     H
F9C5  70          1337      MOV      H,B           ; STORE MSB OF TRAP ADDRESS INTO TRAP AREA
F9C6  23          1338      INX     H
F9C7  0A          1339      LDAX   B              ; FETCH OPCODE BYTE
F9C8  77          1340      MOV      H,A           ; PUT IN TRAP AREA
F9C9  23          1341      INX     H
F9CA  3EC7        1342      MVI     A,(RST 0)     ; REPLACE THE USER'S OPCODE IN USER PROGRAM
F9CC  02          1343      STAX   B              ; WITH A RST 0
F9CD  15          1344      DCR    D
F9CE  C2C2F9      1345      JNZ    G03            ; DO SAME THING AGAIN FOR 2ND BREAKPOINT
                          1346      G04:
F9D1  CDFEFD      1347      CALL   CRLF           ; EXECUTE MONITOR EXIT CODE, RETURNING TO
F9D4  C9          1348      RET                    ; USER CODE
                          1349
1350 ;*****
1351 ; 'H' COMMAND - COMPUTE HEXADECIMAL SUM AND DIFFERENCE
1352 ;
1353 ; THIS ROUTINE EXPECTS TWO HEXADECIMAL PARAMETERS.
1354 ; IT COMPUTES THE SUM AND DIFFERENCE OF THE TWO VALUES
1355 ; AND DISPLAYS THEM ON THE LOCAL CONSOLE DEVICE AS FOLLOWS:
1356 ; <P1+P2> <P1-P2>
1357 HEXN:
F9D5  CD39FE      1358      CALL   EXPR           ; GET TWO NUMBERS
F9D8  CDFEFD      1359      CALL   CRLF
F9DB  01          1360      POP    D              ; DE CONTAINS P2
F9DC  E1          1361      POP    H
F9DD  E5          1362      PUSH  H              ; HL CONTAINS P1
F9DE  19          1363      DAD   D              ; HL := HL + DE := P1 + P2
F9DF  CD56FE      1364      CALL  LADR           ; DISPLAY SUM
F9E2  CD93FC      1365      CALL  BLK           ; TYPE A SPACE
F9E5  E1          1366      POP    H              ; HL CONTAINS P1 AGAIN
F9E6  7D          1367      MOV   A,L            ; COMPUTE HL-DE
F9E7  93          1368      SUB   E              ; A := LSB OF P1 - LSB OF P2
F9E8  6F          1369      MOV   L,A            ; A := LSB OF (P1 - P2)
F9E9  7C          1370      MOV   A,H
F9EA  9A          1371      SBB   D              ; A := MSB OF P1 - MSB OF P2 WITH CARRY
F9EB  67          1372      MOV   H,A            ; H := MSB OF (P1 - P2)
F9EC  CD56FE      1373      CALL  LADR           ; DISPLAY DIFFERENCE
F9EF  C9          1374      RET
                          1375 ;*****
1376 ;
1377 ; 'M' COMMAND - MOVE A BLOCK OF MEMORY
1378 ;
1379 ; THIS ROUTINE EXPECTS THREE HEXADECIMAL PARAMETERS FROM THE
1380 ; LOCAL CONSOLE. THE FIRST AND SECOND PARAMETERS ARE THE BOUNDS OF
1381 ; THE MEMORY AREA TO BE MOVED, THE THIRD PARAMETER IS THE
1382 ; STARTING ADDRESS OF THE DESTINATION AREA.
1383 MOVE:
F9F0  0C          1384      INR   C              ; GET THREE ADDRESSES
F9F1  CD39FE      1385      CALL  EXPR
F9F4  C1          1386      POP   B              ; DESTINATION ADDRESS
F9F5  01          1387      POP   D              ; SOURCE END ADDRESS
F9F6  E1          1388      POP   H              ; SOURCE START ADDRESS

```

LOC	OBJ	SEQ	SOURCE STATEMENT
		1389	MVD:
F9F7	7E	1390	MOV    A,M            ; GET A DATA BYTE
F9F8	02	1391	STAX   B            ; STORE AT DESTINATION
F9F9	03	1392	INX    B            ; MOVE DESTINATION POINTER
F9FA	CD4CFE	1393	CALL   HILO         ; TEST FOR COMPLETION
F9FD	D2F7F9	1394	JNC    MVD
FA00	C9	1395	RET
		1396	;-*****-
		1397	;
		1398	; 'N' COMMAND - PUNCH NULL CHARACTERS FOR TAPE LEADER/TRAILER
		1399	;
		1400	; THIS ROUTINE PUNCHES 60 NULL CHARACTERS ON THE DEVICE ASSIGNED
		1401	; AS THE PUNCH. IT IS ENTERED VIA A JUMP TO ENTRY POINT NUD
		1402	; FROM THE 'E' COMMAND AS WELL AS BEING INVOKED BY
		1403	; THE 'N' COMMAND.
		1404	NULL:
FA01	CD61FF	1405	CALL   TI            ; REQUIRE CR
FA04	FE0D	1406	CPI    CR
FA06	C247F8	1407	JNZ    ERROR
		1408	MVD:
FA09	063C	1409	MVI    B,60         ; SET TO PUNCH 60 NULLS
		1410	NLEADX:
FA0B	CDE5FC	1411	CALL   POC          ; PUNCH ONE ASCII NULL CHARACTER (=00H)
FA0E	00	1412	DB    0
FA0F	05	1413	DCR    B
FA10	C208FA	1414	JNZ    NLEADX       ; DO IT FOR 60 TIMES
FA13	C9	1415	RET
		1416	;-*****-
		1417	;
		1418	; 'Q' COMMAND - I/O SYSTEM STATUS QUERY
		1419	;
		1420	; THIS COMMAND IS INVOKED BY TYPING THE LETTER Q. THIS
		1421	; COMMAND PRODUCES A LISTING OF LOGICAL I/O DEVICES AND
		1422	; THEIR CORRESPONDING PHYSICAL DEVICE ASSIGNMENTS. THE
		1423	; DATA DISPLAYED IS EQUIVALENT TO THE CURRENT VALUE OF IOBYT.
		1424	QUERY:
FA14	CD61FF	1425	CALL   TI            ; REQUIRE CR
FA17	FE0D	1426	CPI    CR
FA19	C247F8	1427	JNZ    ERROR
FA1C	0604	1428	MVI    B,4          ; SET UP OUTER LOOP COUNTER.
		1429	;        THERE ARE 4 LOGICAL DEVICES.
FA1E	2103F9	1430	LXI    H,LTBL       ; POINT HL AT LOGICAL DEVICE TABLE.
		1431	OD:                 ; OUTER LOOP
FA21	CDFEFD	1432	CALL   CRLF         ; START A NEW LINE.
FA24	4E	1433	MOV    C,M          ; DISPLAY LOGICAL DEVICE IDENTIFIER.
FA25	CD95FC	1434	CALL   CON
FA28	CDDEFC	1435	CALL   CONC         ; DISPLAY '='
FA2B	3D	1436	DB    '='
FA2C	23	1437	INX    H            ; POINT AT MASK FOR LOGICAL DEVICE.
FA2D	7E	1438	MOV    A,M          ; FETCH MASK.
FA2E	2F	1439	CMA                 ; INVERT IT
FA2F	4F	1440	MOV    C,A          ; PUT IN C
FA30	23	1441	INX    H            ; POINT AT PHYSICAL DEVICE TABLE
FA31	5E	1442	MOV    E,M          ; ADDRESS OF SUBORDINATE
FA32	23	1443	INX    H            ; TABLE

LOC	OBJ	SEQ	SOURCE STATEMENT
FA33	56	1444	MOV D,M
FA34	23	1445	INX H
FA35	EB	1446	XCHG ; HL ← PHYSICAL DEVICE TABLE
FA36	3A0300	1447	LDA IOBYT
FA39	A1	1448	ANA C ; PHYSICAL SELECTION
FA3A	C5	1449	PUSH B ; SAVE OUTER LOOP COUNTER
FA3B	0604	1450	MVI B,4 ; SET UP INNER LOOP COUNTER
		1451	Q1: ; INNER LOOP
FA3D	4E	1452	MOV C,M ; GET PHYSICAL DEVICE IDENTIFIER
FA3E	23	1453	INX H
FA3F	8E	1454	CMF H ; TEST FOR EQUALITY
FA40	CA48FA	1455	JZ Q2
FA43	23	1456	INX H ; POINT AT NEXT ENTRY
FA44	05	1457	DCR B ; DECREMENT INNER LOOP
FA45	C230FA	1458	JNZ Q1
		1459	Q2:
FA48	CD95FC	1460	CALL COM ; DISPLAY PHYSICAL DEVICE
FA4B	EB	1461	XCHG ; POINT AT MASTER TABLE
FA4C	C1	1462	PDP B
FA4D	05	1463	DCR B ; DECREMENT OUTER LOOP
FA4E	C221FA	1464	JNZ Q0
FA51	C9	1465	RET
		1466	*****
		1467	;
		1468	;'R' COMMAND - READ HEXADECIMAL FILE
		1469	;
		1470	; THIS ROUTINE READS A HEXADECIMAL FILE FROM THE ASSIGNED
		1471	; READER DEVICE AND LOADS IT INTO MEMORY. ONE HEXADECIMAL
		1472	; PARAMETER IS EXPECTED. THIS PARAMETER IS A BASE ADDRESS
		1473	; TO BE ADDED TO THE MEMORY ADDRESS OF EACH DATA BYTE ENCOUNTERED.
		1474	; IN THIS WAY, HEXADECIMAL FILES MAY BE LOADED INTO MEMORY
		1475	; IN AREAS OTHER THAN THAT FOR WHICH THEY WERE ASSEMBLED OR COMPILED.
		1476	; ALL RECORDS READ ARE CHECKSUMMED AND COMPARED AGAINST THE
		1477	; CHECKSUM IN THE RECORD. IF A CHECKSUM ERROR (OR TAPE READ ERROR)
		1478	; OCCURS, THE ROUTINE TAKES AN ERROR EXIT. NORMAL LOADING IS
		1479	; TERMINATED WHEN AN EOF RECORD IS ENCOUNTERED. THE ADDRESS
		1480	; GIVEN WHEN THE EOF RECORD WAS CREATED (VIA THE 'E' COMMAND) REPLACES
		1481	; THE USER'S STORED PC VALUE ONLY IF THE ADDRESS WAS NONZERO.
		1482	; A TRANSFER TO THE PROGRAM MAY THEN BE ACCOMPLISHED BY A 'G<CR>'.
		1483	READ:
FA52	0D	1484	DCR C ; GET ONE ADDRESS; C := 1
FA53	CD39FE	1485	CALL EXPR ; GET THE HEX BASE ADDRESS
FA56	CDFEFD	1486	CALL CRLF ; OUTPUT A <CR>,<LF>
		1487	REDO:
FA59	CD58FF	1488	CALL RIX ; GET AN ASCII CHARACTER FROM THE READER
FA5C	FE3A	1489	CPI ':' ; IS IT A START OF RECORD MARK (':')?
FA5E	C259FA	1490	JNZ REDO ; LOOP UNTIL WE FIND SUCH A RECORD MARK
FA61	AF	1491	XRA A
FA62	57	1492	MOV D,A ; D WILL CONTAIN THE CHECKSUM; INITIALIZE TO 0
FA63	CDD8FD	1493	CALL BYTE ; READ 2 ASCII CHAR REPRESENTING THE RECORD
		1494	; LENGTH AND DECODE THEM INTO 8 BITS BINARY
		1495	; STORING THE RESULT IN A-REG
FA66	CA9EFA	1496	JZ RED3 ; JUMP IF ZERO RECORD LENGTH BECAUSE THIS
		1497	; MEANS IT'S AN EOF RECORD SO WE'RE DONE
FA69	5F	1498	MOV E,A ; E := RECORD LENGTH

LOC	OBJ	SEQ	SOURCE	STATEMENT	
FA6A	CDDDFD	1499	CALL	BYTE	; GET MSB OF LOAD ADDRESS
FA6D	67	1500	MOV	H,A	; H := MSB OF LOAD ADDRESS
FA6E	CDDDFD	1501	CALL	BYTE	; GET LSB OF LOAD ADDRESS
FA71	6F	1502	MOV	L,A	; L := LSB OF LOAD ADDRESS
FA72	CDDDFD	1503	CALL	BYTE	; GET RECORD TYPE AND IGNORE IT
FA75	4B	1504	MOV	C,E	; C := RECORD LENGTH
FA76	E5	1505	PUSH	H	; STORE LOAD ADDRESS ON THE STACK
FA77	2100FF	1506	LXI	H,-256	; COMPUTE BUFFER POINTER
FA7A	39	1507	DAD	SP	; HL NOW POINTS TO THAT PART OF THE MONITOR
		1508			;     STACK ONE PAGE (256 BYTES) BELOW WHERE
		1509			;     THE SP IS CURRENTLY POINTING
		1510			; WE WILL NOW READ DATA FROM THE FILE RECORD
		1511			; AND STORE THEM TEMPORARILY IN THE MONITOR'S
		1512			; STACK STARTING FROM A LOW MEMORY ADDRESS AND
		1513			; MOVING TOWARD A HIGHER MEMORY ADDRESS (REVERSE
		1514			; OF USUAL PROCEDURE WHERE STACK GROWS DOWN)
		1515	RED1:		
FA7B	CDDDFD	1516	CALL	BYTE	; READ DATA; NOTE: 8 BITS OF MEMORY (DATA)
		1517			;     IS REPRESENTED AS 2 HEX CHAR AND EACH HEX
		1518			;     HEX CHAR IS REPRESENTED AS ONE 8 BIT ASCII CHAR
FA7E	77	1519	MOV	H,A	; PUT DATA IN MONITOR BUFFER
FA7F	23	1520	INX	H	; MOVE "UP" THE STACK
FA80	1D	1521	DCR	E	; DECREMENT RECORD LENGTH COUNT
FA81	C27BFA	1522	JNZ	RED1	; LOOP UNTIL RECORD LENGTH COUNTER IS 0
FA84	CDDDFD	1523	CALL	BYTE	; READ THE CHECKSUM RECORD FRAME --- PRIOR TO
		1524			;     CALL TO BYTE, D-REG CONTAINED SUM OF DATA
		1525			;     RECORDS. THE CHECKSUM FRAME SHOULD CONTAIN
		1526			;     THE NEGATIVE OF THIS SUM. BYTE ADDS D AND A
		1527			;     TOGETHER AND SETS THE ZERO BIT IF D = (-A)
FA87	C247F8	1528	JNZ	ERROR	; CHECKSUM ERROR
FA8A	D1	1529	POP	D	; DE = LOAD ADDRESS; STACK ENTRY POINTED TO BY SP
		1530			;     NOW CONTAINS BASE (BIAS) ADDRESS
FA8B	E3	1531	XTHL		; HL = BIAS ADDRESS; CONTENTS OF STACK ENTRY
		1532			;     POINTED TO BY SP NOW IS ADDRESS ONE ABOVE
		1533			;     WHERE LAST DATA IS STORED IN MONITOR STACK
FA8C	EB	1534	XCHG		; DE = BIAS ADDRESS, HL = LOAD ADDRESS
FA8D	19	1535	DAD	D	; HL = BIAS + LA
FA8E	0600	1536	MVI	B,0	; BC = RECORD LENGTH (RL)
FA90	09	1537	DAD	B	; HL = BIAS + LA + RL
FA91	EB	1538	XCHG		; DE = BIAS + LA + RL, HL = BIAS
FA92	E3	1539	XTHL		; HL POINTS TO ADDRESS 1 GREATER THAN WHERE LAST
		1540			;     DATA IS STORED IN MONITOR STACK
		1541			; -----
		1542	RED2:		; LOAD INTO PROPER AREA IN RAM BUT IN
		1543			;     REVERSE ORDER
FA93	2B	1544	DCX	H	; DECREMENT STACK BUFFER POINTER
FA94	7E	1545	MOV	A,H	; A := DATA
FA95	1B	1546	DCX	D	; DECREMENT MEMORY POINTER
FA96	12	1547	STAX	D	; PUT DATA IN DESIGNATED ADDRESS
FA97	0D	1548	DCR	C	; KEEP DOING THIS UNTIL RECORD LENGTH
FA98	C293FA	1549	JNZ	RED2	;     COUNT IS EXHAUSTED
FA9B	C359FA	1550	JMP	REDO	; DONE WITH ONE RECORD, GO GET ANOTHER
		1551			; -----
		1552	RED3:		; EOF RECORD - ENTIRE FILE HAS BEEN READ IN
FA9E	C5	1553	PUSH	B	; SAVE B,C

```

LOC  OBJ          SEQ          SOURCE STATEMENT
FA9F  CDD8FD      1554          CALL    BYTE          ; GET MSB OF LOAD ADDRESS OF EOF RECORD ---
                               1555          ; THIS IS THE (START ADDRESS) SPECIFIED IN
                               1556          ; THE 'E' COMMAND. IF IT IS ZERO, DO NOT
                               1557          ; MODIFY THE USER'S STORED PC IN EXIT TEMPLATE
FAA2  47          1558          MOV     B,A           ; B := MSB OF START ADDRESS
FAA3  CDD8FD      1559          CALL    BYTE          ; GET LSB OF START ADDRESS
FAA6  4F          1560          MOV     C,A           ; C := LSB OF START ADDRESS
FAA7  80          1561          ORA    B              ; SEE IF START ADDRESS IS 0000
FAA8  CAB3FA      1562          JZ     RED4           ; JUMP IF IT IS (DON'T SET NEW PC)
FAAB  2A0400      1563          LHLD   MENTOP
FAAE  2EE1        1564          MVI    L,PLOC AND OFFH ; HL POINTS TO PLOC IN EXIT CODE IN TOP OF RAM
FAB0  70          1565          MOV     M,B           ; STORE MSB OF START ADDRESS
FAB1  28          1566          DCX   H              ; HL POINTS TO PLOC - 1 OF EXIT CODE
FAB2  71          1567          MOV     M,C           ; STORE LSB OF START ADDRESS
                               1568 RED4:          ; FINISH PROCESSING EOF RECORD
FAB3  C1          1569          POP    B              ; RESTORE B,C
FAB4  CDD8FD      1570          CALL    BYTE          ; GET RECORD TYPE AND IGNORE IT
FAB7  CDD8FD      1571          CALL    BYTE          ; GET CHECKSUM
FABA  C247F8      1572          JNZ   ERROR          ; JUMP IF CHECKSUM ERROR
FABD  E1          1573          POP    H              ; CUT BACK STACK POINTER
FABE  C9          1574          RET
                               1575 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*-----*
                               1576 ;
                               1577 ; 'S' COMMAND - SUBSTITUTE MEMORY
                               1578 ;
                               1579 ; THIS ROUTINE EXPECTS ONE PARAMETER FROM THE LOCAL CONSOLE, FOLLOWED
                               1580 ; BY A SPACE. THE PARAMETER IS INTERPRETED AS A MEMORY LOCATION
                               1581 ; AND THE ROUTINE WILL DISPLAY THE CONTENTS OF THAT LOCATION,
                               1582 ; FOLLOWED BY A DASH (-). TO MODIFY MEMORY, TYPE IN THE NEW DATA
                               1583 ; FOLLOWED BY A SPACE OR A CARRIAGE RETURN. IF NO MODIFICATION
                               1584 ; OF THE LOCATION IS REQUIRED, TYPE ONLY A SPACE OR CARRIAGE RETURN.
                               1585 ; IF A SPACE WAS LAST TYPED, THE NEXT MEMORY LOCATION WILL BE DISPLAYED
                               1586 ; AND MODIFICATION OF IT IS ALLOWED. IF A CARRIAGE RETURN WAS ENTERED,
                               1587 ; THE COMMAND IS TERMINATED.
                               1588 ;
                               1589 SUBS:
FABF  CD74FE      1590          CALL    PARAM          ; GET MEMORY ADDRESS
FAC2  D8          1591          RC                      ; ONLY CR ENTERED SO RETURN TO MAIN COMMAND LOOP
                               1592 SUD:
FAC3  7E          1593          MOV     A,M           ; HL HAS REQUESTED MEMORY ADDRESS
FAC4  CD5BFE      1594          CALL    LBYTE         ; DISPLAY CONTENTS OF THAT ADDRESS
FAC7  CDDEFC      1595          CALL    CONC          ; OUTPUT PROMPT CHARACTER
FACA  2D          1596          DB     '-'
FACB  CDC5FE      1597          CALL    PCHK
FACE  D8          1598          RC                      ; CR ENTERED, RETURN TO COMMAND MODE
FACF  CAD9FA      1599          JZ     SUI            ; SPACE ENTERED, SPACE BY
FAD2  EB          1600          XCHG          ; SAVE MEMORY ADDRESS
FAD3  CD7AFE      1601          CALL    PAD          ; GET NEW VALUE
FAD6  EB          1602          XCHG          ; E = VALUE
FAD7  73          1603          MOV     M,E           ; STORE NEW VALUE
FADB  D8          1604          RC                      ; CR ENTERED AFTER VALUE, RETURN
                               1605 SUI:
FAD9  23          1606          INX   H              ; HL POINTS TO NEXT MEMORY LOCATION
FADA  C3C3FA      1607          JMP    SUD
                               1608 ;-----*-----*-----*-----*-----*-----*-----*-----*-----*-----*
    
```

LOC	OBJ	SEQ	SOURCE STATEMENT
		1609	;
		1610	; 'W' COMMAND - WRITE HEXADECIMAL FILE
		1611	;
		1612	; THIS ROUTINE EXPECTS TWO HEXADECIMAL PARAMETERS WHICH ARE
		1613	; INTERPRETED AS THE BOUNDS OF A MEMORY AREA TO BE ENCODED
		1614	; INTO HEXADECIMAL FORMAT AND PUNCHED ON THE ASSIGNED PUNCH
		1615	; DEVICE.
		1616	WRITE:
FADD	CD39FE	1617	CALL    EXPR            ; GET ADDRESS RANGE
FAE0	CDFEFD	1618	CALL    CRLF           ; NEW LINE
FAE3	D1	1619	POP     D              ; DE := HIGH ADDRESS
FAE4	E1	1620	POP     H              ; HL := LOW ADDRESS
		1621	WRO:
FAE5	CDE5FC	1622	CALL    POC            ; EMIT RECORD MARK
FAE8	3A	1623	DB     ' '              ;
FAE9	011000	1624	LXI    B,16            ; INITIALIZE B := 0, C := AH (DECIMAL 16)
		1625	;
FAEC	E5	1626	PUSH    H              ; SAVE HL
		1627	WR1:
FAED	04	1628	INR     B              ; INCREMENT RECORD LENGTH
FAEE	0D	1629	DCR     C              ;
FAEF	CAF8FA	1630	JZ      WR2            ; TERMINATE ON COUNT OF 16 BYTES
FAF2	CD4CFE	1631	CALL    HILD           ; OR END OF RANGE
FAF5	D2EDFA	1632	JNC    WR1            ; WHICHEVER OCCURS FIRST
		1633	;
		1634	WR2:
FAF8	E1	1635	POP     H              ; RESTORE HL := LOW ADDRESS
FAF9	05	1636	PUSH    D              ; SAVE HIGH ADDRESS
FAFA	1600	1637	MVI    D,0            ; INITIALIZE CHECKSUM D := 0
FAFC	78	1638	MOV    A,B            ; A := RECORD LENGTH
FAFD	CDAFFE	1639	CALL    PBYTE         ; EMIT RECORD LENGTH
FB00	CDAAFE	1640	CALL    PADR           ; EMIT HL := LOW ADDRESS
FB03	AF	1641	XRA     A              ;
FB04	CDAFFE	1642	CALL    PBYTE         ; EMIT RECORD TYPE = 1
		1643	;
		1644	WR3:
FB07	7E	1645	MOV    A,M            ; FETCH DATA
FB08	CDAFFE	1646	CALL    PBYTE         ; EMIT IT
FB0B	23	1647	INX     H              ; INCREMENT MEMORY ADDRESS
FB0C	05	1648	DCR     B              ; DECREMENT COUNT
FB0D	C207FB	1649	JNZ    WR3            ; LOOP UNTIL ENTIRE RECORD HAS BEEN OUTPUT
FB10	AF	1650	XRA     A              ;
FB11	92	1651	SUB     D              ; D CONTAINS RUNNING CHECKSUM
FB12	CDAFFE	1652	CALL    PBYTE         ; EMIT CHECKSUM := -D
FB15	D1	1653	POP     D              ; RESTORE DE := HIGH ADDRESS
FB16	2B	1654	DCX     H              ; BACKUP MEMORY POINTER
		1655	; NOW PUNCH CR,LF --- IGNORED BY THE 'R'
		1656	;        COMMAND BUT HANDY IF LISTING PUNCHED
		1657	;        TAPE ON THE TTY
FB17	CDE5FC	1658	CALL    POC            ; PUNCH CARRIAGE RETURN
FB1A	0D	1659	DB     CR              ;
FB1B	CDE5FC	1660	CALL    POC            ; PUNCH LINE FEED CHARACTER
FB1E	0A	1661	DB     LF              ;
FB1F	CD4CFE	1662	CALL    HILD           ; TEST FOR TERMINATION
FB22	D2E5FA	1663	JNC    WRO            ; IF NOT DONE, FORM NEXT RECORD AND OUTPUT IT

LOC	OBJ	SEQ	SOURCE STATEMENT
FB25	C9	1664	RET
		1665	-----
		1666	
		1667	; 'X' COMMAND - EXAMINE AND MODIFY CPU REGISTERS
		1668	
		1669	; THIS ROUTINE ALLOWS THE OPERATOR TO EXAMINE AND/OR MODIFY
		1670	; THE CONTENTS OF THE USER PROGRAM'S REGISTERS. THE REGISTER
		1671	; VALUES WERE STORED AS A RESULT OF A PREVIOUS BREAKPOINT AND
		1672	; WILL BE RESTORED TO THE USER PROGRAM DURING A SUBSEQUENT 'G'
		1673	; COMMAND.
		1674	X1:
FB26	2181FB	1675	LXI    H,ACTBL       ; POINT TO ACCESS TABLE
FB29	CDC5FE	1676	CALL   PCHK         ; GET REGISTER IDENTIFIER
FB2C	DA6AFB	1677	JC     X5           ; IF CARRY = 1, CR ENTERED
FB2F	0E0C	1678	MVI    C,HREGS
		1679	X0:
FB31	8E	1680	CMP    H
FB32	CA3FFB	1681	JZ     X1           ; MATCHED REGISTER IDENTIFIER
FB35	23	1682	INX    H           ; POINT TO NEXT TABLE ENTRY
FB36	23	1683	INX    H
FB37	23	1684	INX    H
FB38	0D	1685	DCR    C           ; DECREMENT REGISTER COUNTER
FB39	C231FB	1686	JNZ    X0           ; TRY AGAIN
FB3C	C347FB	1687	JNP    ERROR       ; NOT IN TABLE, ERROR
		1688	X1:
FB3F	CD93FC	1689	CALL   BLK
		1690	X2:
FB42	CD25FE	1691	CALL   DREG         ; DISPLAY THE REGISTER
FB45	CDDEFC	1692	CALL   CONC
FB48	2D	1693	DB     '-'         ; TYPE PROMPT
FB49	CDC5FE	1694	CALL   PCHK         ; SKIP IF NULL ENTRY
FB4C	D8	1695	RC     ; CR ENTERED, RETURN TO COMMAND MODE
FB4D	CA60FB	1696	JZ     X4
FB50	E5	1697	PUSH   H           ; SAVE POINTER TO ACTBL
FB51	C5	1698	PUSH   B           ; SAVE PRECISION
FB52	CD7AFE	1699	CALL   PAQ          ; GET NEW REG VALUE
FB55	7D	1700	MOV    A,L
FB56	12	1701	STAX   D           ; STORE LSB IN REGISTER AREA
FB57	F1	1702	POP    PSW         ; RETRIEVE PRECISION (A)
FB58	87	1703	ORA    A           ; SET SIGN
FB59	FA5FFB	1704	JN     X3           ; 8 BITS ONLY
FB5C	13	1705	INX    D
FB5D	7C	1706	MOV    A,H
FB5E	12	1707	STAX   D           ; STORE MSB IN REGISTER AREA
		1708	X3:
FB5F	E1	1709	POP    H           ; RETRIEVE ACTBL POINTER
		1710	X4:
FB60	AF	1711	XRA    A
FB61	86	1712	ORA    H
FB62	F8	1713	RN     ; END OF TABLE, RETURN TO COMMAND MODE
FB63	78	1714	MOV    A,B         ; TEST DELIMITER
FB64	FE0D	1715	CPI    CR
FB66	C8	1716	RZ     ; CR ENTERED, RETURN TO COMMAND MODE
FB67	C342FB	1717	JMP    X2
		1718	-----



LOC	OBJ	SEQ	SOURCE STATEMENT
		1719 X5:	
FB6A	CDFEFD	1720	CALL    CRLF                    ; DISPLAY ALL THE REGISTER VALUES
		1721 X6:	
FB6D	CD93FC	1722	CALL    BLK                    ; OUTPUT A SPACE
FB70	AF	1723	XRA    A                        ; CLEAR A
FB71	86	1724	ORA    M                        ; SET CONDITION CODES
FB72	FB	1725	RN                               ; ALL DONE, RETURN TO COMMAND MODE
FB73	4E	1726	MOV    C,M                      ; C CONTAINS A REGISTER IDENTIFIER (A,B,C,D...)
FB74	CD95FC	1727	CALL    COM                      ; PRINT CHARACTER
FB77	CDDEFC	1728	CALL    COMC                    ; PRINT EQUAL SIGN
FB7A	3D	1729	DB    ' = '                      ;
FB7B	CD25FE	1730	CALL    DREG                    ; DISPLAY REGISTER CONTENTS
FB7E	C36DFB	1731	JMP    X6                        ; CONTINUE
		1732 ;	
		1733 ;	TABLE FOR ACCESSING REGISTERS
		1734 ;	TABLE CONTAINS:
		1735 ;	(1) REGISTER IDENTIFIER
		1736 ;	(2) LOCATION ON STORAGE PAGE
		1737 ;	(3) PRECISION
		1738 ;	
		1739 ACTBL:	
FB31	41	1740	DB    'A',            ALOC AND HMSK, 0
FB32	CF		
FB33	00		
FB34	42	1741	DB    'B',            BLOC AND HMSK, 0
FB35	CB		
FB36	00		
FB37	43	1742	DB    'C',            CLOC AND HMSK, 0
FB38	CA		
FB39	30		
FB3A	44	1743	DB    'D',            DLOC AND HMSK, 0
FB3B	C9		
FB3C	00		
FB3D	45	1744	DB    'E',            ELOC AND HMSK, 0
FB3E	C8		
FB3F	00		
FB40	46	1745	DB    'F',            FLOC AND HMSK, 0
FB41	CE		
FB42	00		
FB43	48	1746	DB    'H',            HLOC AND HMSK, 0
FB44	3D		
FB45	30		
FB46	49	1747	DB    'I',            ILOC AND HMSK, 0
FB47	CD		
FB48	00		
FB49	4C	1748	DB    'L',            LLOC AND HMSK, 0
FB4A	DC		
FB4B	00		
FB4C	4D	1749	DB    'M',            MLOC AND HMSK, 1
FB4D	DD		
FB4E	01		
FB4F	50	1750	DB    'P',            PLOC AND HMSK, 1
FB40	E1		
FB41	01		
FB42	53	1751	DB    'S',            SLOC AND HMSK, 1

```

LOC  OBJ      SER      SOURCE STATEMENT
FBA3  D1
FBA4  01
FBA5  FF      1752          DB          -1
000C      1753  MREGS    EQU      (%ACTBL)/3      ; LENGTH OF ACCESS TABLE
1754  ;-----*-----*-----*-----*-----*-----*-----*-----*-----*-----*
1755  ;
1756  ; 'Z' COMMAND - TRANSFER CONTROL TO DIAGNOSTIC PROGRAM IN PROM
1757  ; THIS ROUTINE EXPECTS A '$' AT WHICH POINT IT WILL CALL THE DIAGNOSTIC PROGRAM.
1758  Z:
FBA6  CD61FF   1759          CALL     TI          ; GET A CHARACTER FROM THE CONSOLE
FBA9  FE24     1760          CPI      '$'         ; IS IT A '$'?
FBA8  C247F8   1761          JNZ     ERRD        ; ERROR IF IT ISN'T
FBAE  CD61FF   1762          CALL     TI          ; GET A CHARACTER FROM THE CONSOLE
FBB1  FE0D     1763          CPI      CR         ; EXPECT A CARRIAGE RETURN
FBB3  C247F8   1764          JNZ     ERRD        ; ERROR IF IT ISN'T
FBB6  3E0C     1765          MVI     A,BTDGON    ; TURN ON THE BDDT/DIAGNOSTIC PROM
FBB8  03FF     1766          OUT     CPUC
FBBA  CD00EB   1767          CALL     DIAGN      ; CALL THE DIAGNOSTIC PROGRAM
FBBD  C9       1768          RET             ; RETURN TO MAIN COMMAND LOOP
1769  ;*****
1770  ;*
1771  ;*  END OF MONITOR COMMANDS, BEGINNING OF I/O ROUTINES
1772  ;*
1773  ;*****
1774  ;*****
1775  ; 'CI' - EXTERNALLY REFERENCED ROUTINE
1776  ;     ENTERED VIA CALL FROM 'TI' ROUTINE
1777  ; PROCESS: LOCAL CONSOLE INPUT CODE
1778  ; INPUT:
1779  ; OUTPUT: CHARACTER RETURNED IN A-REG
1780  ; MODIFIED: A, FLAGS
1781  ; STACK USAGE: 2 BYTES
1782  ; EXPLANATION: BASED ON I/O STATUS BYTE (IOBYT), DECIDE IF CONSOLE INPUT
1783  ;     DEVICE IS TTY, CRT, BATCH, OR USER-DEFINED DEVICE. IF IT IS TTY OR CRT
1784  ;     LOOP UNTIL READ, INPUT THE CHARACTER, THEN RETURN. IF IT IS BATCH,
1785  ;     JUMP TO 'RI' ROUTINE. IF IT IS USER-DEFINED DEVICE, JUMP TO @USER.
1786  ;*****
1787  CI:
FBBE  3A0300   1788          LDA     IOBYT      ; GET STATUS BYTE
FBC1  E603     1789          ANI     NOT CMK    ; LOOK AT ONLY CONSOLE FIELD
FBC3  C2D0FB   1790          JNZ     CIO        ; JUMP IF CONSOLE IS NOT TTY
1791  ;-----*-----*-----*-----*-----*-----*-----*-----*-----*-----*
1792  ; CONSOLE = TTY
1793  TTYIN:
FBC6  DBF5     1794          IN      TTYS      ; TTY STATUS PORT
FBC8  E602     1795          ANI     RRDY      ; CHECK FOR RECEIVE BUFFER READY
FBCA  CAC6FB   1796          JZ      TTYIN     ; LOOP UNTIL IT IS READY
FBCD  DBF4     1797          IN      TTYI     ; INPUT CHARACTER FROM TTY
FBCF  C9       1798          RET             ; RETURN; CHARACTER IN A-REG
1799  ;-----*-----*-----*-----*-----*-----*-----*-----*-----*-----*
1800  ; CONSOLE = CRT, BATCH, OR USER-DEFINED
1801  CID:
FBD0  FE01     1802          CPI      CCRT     ; LOCAL CONSOLE = CRT?
FBD2  C2FDFB   1803          JNZ     CI4       ; JUMP IF CONSOLE IS NOT CRT
FBD5  E5       1804          PUSH    H         ; SAVE HL
    
```

```

LOC  OBJ          SEQ          SOURCE STATEMENT
-----
FBDB 2A0400      1805          LHLD   MENTOP
FBD9 2ECC        1806          MVI    L,ILOC-1 AND OFFH; HL NOW POINTS TO CONFIGURATION BYTE STORED
1807                                     ; IN EXIT TEMPLATE IN TOP PAGE OF RAM
FBDB 7E         1808          MOV    A,M           ; A := CONFIGURATION BYTE
FBDC E1         1809          POP    H             ; RESTORE HL
FBDD 3F         1810          RRC                    ; ROTATE BIT 0 INTO CARRY BIT, THUS CARRY = 1
1811                                     ; MEANS RUNNING ON SYSTEM WITHOUT INTEGRATED
1812                                     ; CRT
FBDE D2EBFB     1813          JNC    C12           ; JUMP IF INTEGRATED CRT IS PRESENT
1814 ; -----
1815 ; CONSOLE = SERIAL CRT
1816 C11:
FBEB DBF7       1817          IN     USCG          ; INPUT CRT STATUS
FBEB E602       1818          ANI   RRDY          ; CHECK FOR RECEIVER BUFFER READY
FBEB CAE1FB     1819          JZ    C11           ; LOOP UNTIL IT IS READY
FBEB DBF6       1820          IN     USCI          ; GET CHARACTER FROM THE CRT
FBEB C9        1821          RET                    ; RETURN; CHARACTER IS IN A-REG
1822 ; -----
1823 ; CONSOLE = INTEGRATED CRT
1824 C12:
FBEB C5        1825          PUSH  B             ; SAVE B,C
1826 C13:
FBEB 0613       1827          MVI   B,KSTS        ; LOAD KEYBOARD STATUS COMMAND
FBEB CD7FFF     1828          CALL IOCDRI         ; INPUT KEYBOARD STATUS FROM IOC
FBEB E601       1829          ANI   KRDY          ; IS THE KEYBOARD READY?
FBEB CAECFB     1830          JZ    C13           ; LOOP UNTIL IT IS
FBEB 0612       1831          MVI   B,KEYC        ; LOAD INPUT DATA COMMAND
FBEB CD7FFF     1832          CALL IOCDRI         ; INPUT DATA FROM THE KEYBOARD
FBEB C1         1833          POP   B             ; RESTORE B,C
FBEB C9        1834          RET                    ; RETURN; CHARACTER IS IN A-REG
1835 ; -----
1836 ; CONSOLE IS BATCH OR USER-DEFINED DEVICE
1837 C14:
FBEB FE02       1838          CPI   BATCH         ;
FBEB CADFFC     1839          JZ    RI             ; BATCH MODE, INPUT = READER
FC02 3EE8       1840          MVI   A,CILDC AND HNSK; USER DEFINE LOCAL CONSOLE INPUT
FC04 C38CFC     1841          JMP   USER          ;
1842 ; ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
1843 ; 'BREAK' - ENTERED VIA CALLS FROM 'BLK','CON','LDM' ROUTINES
1844 ; PROCESS: TEST FOR OPERATOR INTERRUPTION OF COMMAND (I.E. DID OPERATOR
1845 ; DEPRESS THE "BREAK" KEY)
1846 ; INPUT:
1847 ; OUTPUT:
1848 ; MODIFIED: A,FLAGS
1849 ; STACK USAGE: 4 BYTES
1850 BREAK:
FC07 CD44FD     1851          CALL  CSTS          ; SEE IF A KEY WAS DEPRESSED
FC0A B7         1852          ORA   A             ;
FC0B C8         1853          RZ                    ; NO CHARACTER READY
FC0C C361FF     1854          JMP   TI             ; GET THE CHARACTER
1855 ; ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
1856 ; 'RI' - EXTERNALLY REFERENCED ROUTINE
1857 ; ENTERED VIA CALLS FROM 'CI','RIX' ROUTINES
1858 ; PROCESS: READER INPUT CODE
1859 ; INPUT:
  
```



```

LOC  OBJ          SEQ          SOURCE STATEMENT

1915 RIS:
FC58 FE04          1916          CPI      RPTR          ; IS READER THE PAPER TAPE READER?
FC5A C282FC        1917          JNZ      R18          ; JUMP IF IT ISN'T
1918 ;-----
1919 ; READER = PAPER TAPE READER
FC5D C5            1920          PUSH     B            ; SAVE BC
FC5E 0650          1921          MVI     B,RDRRC OR PTRADV; LOAD READER ADVANCE 1 FRAME COMMAND
FC60 CDE4FF        1922          CALL    PIOCMB        ; OUTPUT THE COMMAND
FC63 26FA          1923          MVI     H,TOUT        ; 250 MS. TIMEOUT COUNTER
1924 RI6:
FC65 0611          1925          MVI     B,RSTC        ; LOAD READER STATUS COMMAND
FC67 CDB5FF        1926          CALL    PIDDRI        ; READ STATUS
FC6A E601          1927          ANI     PTRDY         ; IS THE READER READY?
FC6C C279FC        1928          JNZ     R17           ; JUMP IF IT IS
FC6F CD1EFE        1929          CALL    DELAY         ; STALL FOR 1 MS.
FC72 25            1930          DCR     H             ; 250 MS. TIMEOUT LOOP
FC73 C265FC        1931          JNZ     R16           ;
FC76 C347FC        1932          JMP     R13           ; 250 MS. ARE UP; RETURN WITH CARRY = 1 (EOF COND)
1933 RI7:
FC79 0610          1934          MVI     B,RDRRC        ; LOAD READER COMMAND
FC7B CDB5FF        1935          CALL    PIDDRI        ; READ A CHARACTER FROM THE PAPER TAPE READER
FC7E B7            1936          ORA     A             ; RESET CARRY BIT
FC7F C1            1937          POP     B             ; RESTORE BC
FC80 E1            1938          POP     H             ;
FC81 C9            1939          RET                    ; RETURN SUCCESSFULLY WITH CARRY = 0
1940 ;-----
1941 ; READER IS USER-DEFINED DEVICE 1 OR DEVICE 2
1942 RI8:
FC82 E1            1943          POP     H             ;
FC83 FE08          1944          CPI     RUSE1         ;
FC85 3EEE          1945          MVI     A,R1LOC AND HNSK
FC87 CA8CFC        1946          JZ      0USER         ; READER = USER-DEFINED DEVICE 1
FC8A 3EF1          1947          MVI     A,R2LOC AND HNSK
1948 ;*****JMP     0USER         ; READER = USER-DEFINED DEVICE 2
1949 ;-----
1950 ; '0USER' - ENTERED VIA JUMPS FROM 'LO','LON','RI','CI','BLK','COM',
1951 ; 'CO','POC','PO','CSTS' ROUTINES
1952 ; ENTERED VIA FALL-THRU FROM 'RI' ROUTINE
1953 ; PROCESS: USER-DEFINED I/O ENTRY POINT TRANSFER LOGIC
1954 ; INPUT: A-REG CONTAINS LSB ADDRESS PTR INTO USER-DEFINED ENTRY POINT TABLE (XTBL)
1955 ; OUTPUT:
1956 ; MODIFIED:
1957 ; STACK USAGE:
1958 0USER:
FC8C E5            1959          PUSH     H            ; SAVE HL, CREATE A STACK ENTRY
FC8D 2A0400        1960          LHLD   HENTOP        ;
FC90 6F            1961          MOV     L,A          ; HL NOW POINTS TO PROPER USER ENTRY POINT IN
1962 ; XTBL IN EXIT TEMPLATE IN TOP PAGE OF RAM
FC91 E3            1963          XTHL                    ; RESTORE HL; SP NOW POINTS TO USER ENTRY POINT
FC92 C9            1964          RET                    ; BEGIN EXECUTING AT THIS ENTRY POINT
1965 ;-----
1966 ; 'CO' - EXTERNALLY REFERENCED ROUTINE
1967 ; ENTERED VIA CALL FROM 'TI' ROUTINE
1968 ; 'BLK' - ENTERED VIA CALLS FROM 'H','K' COMMANDS
1969 ; 'COM' - ENTERED VIA CALLS FROM 'Q','X' COMMANDS
    
```

```

LOC  OBJ      SEQ      SOURCE STATEMENT
1970 ;          ENTERED VIA JUMPS FROM 'CONC', 'HXD' ROUTINES ;
1971 ; 'TTYOUT' - ENTERED VIA JUMPS FROM 'LOM', 'LO', 'POC', 'PO' ROUTINES ;
1972 ; 'CRTOUT' - ENTERED VIA JUMPS FROM 'LOM', 'LO' ROUTINES ;
1973 ;          ENTERED VIA CALL FROM BOOTSTRAP PROGRAM ;
1974 ; PROCESS: LOCAL CONSOLE OUTPUT CODE ;
1975 ; INPUT: VALUE IN C-REG ;
1976 ; OUTPUT: DATA OUTPUT TO APPROPRIATE DEVICE ;
1977 ; MODIFIED: A, FLAGS, C ;
1978 ; STACK USAGE: 2 BYTES ;
1979 ; ~~~~~ ;
1980 BLK:          ; PRINT A BLANK
FC93 OE20        1981      MVI      C, ' '
1982 COM:          ; LOCAL CONSOLE OUTPUT
FC95 3A0300      1983      LDA      IOBYT   ; GET STATUS BYTE
FC98 E603        1984      ANI      NOT CMSK  ; LOOK ONLY AT CONSOLE FIELD
FC9A FE02        1985      CPI      BATCH   ; IS CONSOLE = BATCH?
FC9C C407FC      1986      CNZ      BREAK   ; IF SO, DO NOT HONOR BREAK KEY IN BATCH MODE
1987            ; IF IT ISN'T, THEN TEST FOR BREAK KEY
1988 CO:          ; EXTERNAL ENTRY POINT
FC9F 3A0300      1989      LDA      IOBYT   ; GET STATUS BYTE
FCA2 E603        1990      ANI      NOT CMSK  ; LOOK ONLY AT CONSOLE FIELD
FCA4 C2B2FC      1991      JNZ      COO      ; JUMP IF CONSOLE IS NOT TTY
1992 ; ~~~~~
1993 ; CONSOLE = TTY
1994 TTYOUT:
FCA7 DBF5        1995      IN       TTYS     ; LOCAL CONSOLE = TTY; GET TTY STATUS
FCA9 E601        1996      ANI      TRDY     ; IS IT READY?
FCAB CA07FC      1997      JZ       TTYOUT   ; LOOP UNTIL IT IS
FCAE 79          1998      MOV      A,C      ; LOAD CHARACTER TO BE OUTPUT
FCAF D3F4        1999      OUT      TTYO     ; OUTPUT CHARACTER
FCB1 C9          2000      RET
2001 ; ~~~~~
2002 ; CONSOLE IS CRT, BATCH, OR USER-DEFINED
2003 COO:
FCB2 FE02        2004      CPI      BATCH   ; CONSOLE = BATCH?
FCB4 CA1EFD      2005      JZ       LO       ; JUMP TO LIST OUTPUT IF IT IS
FCB7 FE01        2006      CPI      CCRT   ; LOCAL CONSOLE = CRT?
FCB9 3EEB        2007      MVI      A,COLOC AND OFFH
FCBB C28CFC      2008      JNZ      @USER   ; JUMP IF IT ISN'T, I.E. CONSOLE IS
2009            ; USER DEFINED LOCAL CONSOLE OUTPUT
2010 ; ~~~~~
2011 ; CONSOLE = CRT
2012 CRTOUT:
FCBE E5          2013      PUSH     H           ; SAVE H,L
FCBF 2A0400      2014      LHLD    MEMTOP
FCC2 2ECC        2015      MVI      L,ILOC-1 AND OFFH ; HL NOW POINTS TO CONFIGURATION BYTE IN EXIT TEMPLATE
FCC4 7E          2016      MOV      A,M      ; A NOW CONTAINS THIS CONFIGURATION BYTE
FCC5 E1          2017      POP     H           ; RESTORE H,L
FCC6 0F          2018      RRC
2019            ; ROTATE BIT 0 INTO CARRY BIT; THUS CARRY
2020            ; = 1 IF INTEGRATED CRT NOT PRESENT
FCC7 D2D5FC      2020      JNC     CRTDT2   ; JUMP IF INTEGRATED CRT
2021 ; ~~~~~
2022 ; CONSOLE = SERIAL CRT
2023 CRTOT1:
FCCA DBF7        2024      IN       USCS     ; INTELLEC WITH SERIALLY CONNECTED CRT
                ; INPUT CRT STATUS

```







LOC	OBJ	SEQ	SOURCE STATEMENT
FD35	CDB5FF	2135	CALL P1ODR1 ; READ STATUS
FD38	E601	2136	ANI LPTRY ; IS IT READY?
FD3A	CA33FD	2137	JZ LPO ; LOOP UNTIL IT IS
FD3D	0614	2138	MVI B,LPTC ; LOAD LINE PRINTER PRINT COMMAND
FD3F	CDCEFF	2139	CALL P1ODR3 ; OUTPUT CHARACTER CONTAINED IN C-REG
FD42	C1	2140	POP B ; RESTORE BC
FD43	C9	2141	RET
		2142	;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
		2143	; 'CSTS' - EXTERNALLY REFERENCED ROUTINE ;
		2144	; ENTERED VIA CALL FROM 'BREAK' ROUTINE ;
		2145	; PROCESS: LOCAL CONSOLE INPUT STATUS ;
		2146	; INPUT: ;
		2147	; OUTPUT: A-REG CONTAINS 00 IF NO KEY HAS BEEN DEPRESSED. ;
		2148	; A-REG CONTAINS FFH IF A KEY HAS BEEN DEPRESSED ;
		2149	; MODIFIED: A, FLAGS ;
		2150	; STACK USAGE: 2 BYTES ;
		2151	;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
		2152	CSTS: ; LOCAL CONSOLE INPUT STATUS
FD44	3A0300	2153	LDA IOBYT ; GET STATUS BYTE
FD47	E603	2154	ANI NOT CHSK ; LOOK ONLY AT CONSOLE FIELD OF IOBYT
FD49	C253FD	2155	JNZ CS0 ; JUMP IF CONSOLE IS NOT TTY
		2156	-----
		2157	; CONSOLE = TTY
FD4C	DBF5	2158	IN TTYS ; GET TTY STATUS
FD4E	E602	2159	ANI RRDY ; IS RECEIVE BUFFER READY? (IF TTY KEY WAS
		2160	; DEPRESSED, ZERO BIT WILL BE RESET)
FD50	C374FD	2161	JNP CS2
		2162	-----
		2163	; CONSOLE = CRT, BATCH, OR USER-DEFINED
		2164	CS0:
FD53	FE01	2165	CPI CCRT ; CONSOLE = CRT?
FD55	C279FD	2166	JNZ CS3 ; JUMP IF CONSOLE IS NOT CRT
FD58	E5	2167	PUSH H ; SAVE H,L
FD59	2A0400	2168	LHLD HENTOP
FD5C	2ECC	2169	MVI L,ILOC-1 AND OFFH; HL POINTS TO CONFIGURATION BYTE IN EXIT TEMPLATE
FD5E	7E	2170	MOV A,M ; A CONTAINS THIS CONFIGURATION BYTE
FD5F	E1	2171	POP H ; RESTORE H,L
FD60	0F	2172	RRC ; ROTATE BIT 0 INTO CARRY; THUS CARRY = 1
		2173	; MEANS INTEGRATED CRT NOT PRESENT
FD61	D268FD	2174	JNC CS1 ; JUMP IF INTEGRATED CRT PRESENT
		2175	-----
		2176	; CONSOLE = SERIAL CRT
FD64	DBF7	2177	IN USCS ; GET CRT STATUS
FD66	E602	2178	ANI RRDY ; IS RECEIVE BUFFER READY? (IF KEY HAS BEEN
		2179	; DEPRESSED, ZERO BIT WILL BE RESET)
FD68	C374FD	2180	JNP CS2
		2181	-----
		2182	; CONSOLE = INTEGRATED CRT
		2183	CS1: ; INTELLEC WITH INTEGRATED CRT
FD6B	C5	2184	PUSH B ; SAVE B,C
FD6C	0613	2185	MVI B,KSTB ; LOAD CRT STATUS COMMAND
FD6E	CD7FFF	2186	CALL IOCDR1 ; GET CRT STATUS
FD71	E601	2187	ANI KRDY ; IS RECEIVE BUFFER READY? (IF KEY HAS BEEN
		2188	; DEPRESSED, ZERO BIT WILL BE RESET)
FD73	C1	2189	POP B ; RESTORE B,C



LOC	OBJ	SEQ	SOURCE STATEMENT
		2245	; 'IODEF' - EXTERNALLY REFERENCED ROUTINE ;
		2246	; PROCESS: DEFINE USER I/O ENTRY POINTS ;
		2247	; INPUT: SELECTION CODE IN C-REG, USER ENTRY POINT ADDRESS IN D,E ;
		2248	; OUTPUT: ;
		2249	; MODIFIED: A, FLAGS ;
		2250	; STACK USAGE: 8 BYTES ;
		2251	; EXPLANATION: POINT HL TO TABLE OF USER ENTRY POINTS IN TOP OF RAM; ;
		2252	;     SUBSTITUTE IN THERE THE ADDRESS GIVEN BY THE USER IN DE REGISTERS. ;
		2253	;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; ;
		2254	IODEF: ;
FD94	E5	2255	PUSH    H                    ; SAVE H & L
FD95	C5	2256	PUSH    B                    ; SAVE B & C
FD96	2A0400	2257	LHLD    MENTOP               ; GET XTBL+1
FD99	2EE9	2258	MVI     L,XTBL+1 AND OFFH; HL NOW POINTS TO XTBL+1 IN TOP PAGE OF RAM
FD9B	79	2259	MOV     A,C                  ; A := LOGICAL DEVICE CATEGORY
FD9C	FE08	2260	CPI     UCS+1
FD9E	D247F8	2261	JNC     ERROR               ; INVALID SELECTION CODE
FDA1	81	2262	ADD     C                    ; DOUBLE INDEX
FDA2	81	2263	ADD     C                    ; TRIPLE INDEX
FDA3	4F	2264	MOV     C,A
FDA4	0600	2265	MVI     B,0
FDA6	09	2266	DAD     B                    ; COMPUTE PROPER INDEX INTO XTBL
FDA7	73	2267	MOV     M,E                 ; STORE BRANCH OPERAND IN INSTRUCTION
FDA8	23	2268	INX     H
FDA9	72	2269	MOV     H,D                 ; STORE THE USER-DEFINED I/O ENTRY ROUTINE
		2270	;     ADDRESS IN THE PROPER PLACE IN XTBL,
		2271	;     SO IT LOOKS LIKE:
		2272	;     JMP <USER-DEFINED ADDRESS>
FDAA	C1	2273	POP     B                    ; RESTORE B & C
FDA8	E1	2274	POP     H                    ; RESTORE H & L
FDAC	C9	2275	RET
		2276	;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; ;
		2277	; 'UI' - EXTERNALLY REFERENCED ROUTINE ;
		2278	; PROCESS: INPUT A CHARACTER FROM THE UPP ;
		2279	; INPUT: B CONTAINS MSB OF PROM ADDRESS ;
		2280	;     C CONTAINS LSB OF PROM ADDRESS ;
		2281	; OUTPUT: DATA IN A-REG ;
		2282	; MODIFIED:A,FLAGS ;
		2283	; STACK USAGE: 6 BYTES ;
		2284	;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; ;
		2285	UI: ;
		2286	; IT IS ASSUMED THE 'UPPS' ROUTINE HAS BEEN
		2287	; CALLED AND THAT THE UPP UNIT IS READY
FDAE	C5	2288	PUSH    B                    ; SAVE B,C
FDAE	D617	2289	MVI     B,RPPC               ; LOAD THE READ PROM COMMAND
		2290	; C CONTAINS PROM LOW ADDRESS
FDB0	CDCEFF	2291	CALL    P1ODR3               ; OUTPUT READ PROM COMMAND
		2292	; OUTPUT PROM LOW ADDRESS
FDB3	C1	2293	POP     B                    ; RESTORE B,C; B CONTAINS PROM HIGH ADDRESS
FDB4	C5	2294	PUSH    B                    ; SAVE B,C
FDB5	48	2295	MOV     C,B                 ; C CONTAINS PROM HIGH ADDRESS
FDB6	CD1FF	2296	CALL    P1ODR4               ; OUTPUT PROM HIGH ADDRESS
FDB9	C1	2297	POP     B                    ; RESTORE B,C
FDBA	CD80FF	2298	CALL    P1ODR2               ; INPUT PROM DATA
FDBD	C9	2299	RET

```

LOC OBJ      SEQ      SOURCE STATEMENT
2300 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
2301 ; 'UO' - EXTERNALLY REFERENCED ROUTINE
2302 ; PROCESS: OUTPUT A CHARACTER TO THE UPP
2303 ; INPUT: C CONTAINS THE CHARACTER TO BE WRITTEN INTO THE PROM
2304 ;          D CONTAINS THE MSB OF THE PROM ADDRESS
2305 ;          E CONTAINS THE LSB OF THE PROM ADDRESS
2306 ; OUTPUT:
2307 ; MODIFIED: A, FLAGS
2308 ; STACK USAGE: 8 BYTES
2309 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
2310 UO:
2311 ; IT IS ASSUMED THE 'UPPS' ROUTINE HAS BEEN
2312 ; CALLED AND THAT THE UPP UNIT IS READY
FDBE C5      2313      PUSH      B          ; SAVE B,C
FDBF 0616    2314      MVI      B,UPPC    ; LOAD WRITE PROM COMMAND
FDC1 4B      2315      MOV      C,E      ; LOAD PROM LOW ADDRESS
FDC2 CDCEFF  2316      CALL     P1DDR3   ; OUTPUT WRITE PROM COMMAND
2317 ; OUTPUT PROM LOW ADDRESS
FDC5 4A      2318      MOV      C,D      ; LOAD PROM HIGH ADDRESS
FDC6 CDD1FF  2319      CALL     P1DDR4   ; OUTPUT PROM HIGH ADDRESS
FDC9 C1      2320      POP      B          ; RESTORE B,C; C CONTAINS THE DATA TO BE
2321 ; WRITTEN TO THE PROM
FDCA CDD1FF  2322      CALL     P1DDR4   ; OUTPUT DATA TO PROM
FDCD C9      2323      RET
2324 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
2325 ; 'UPPS' - EXTERNALLY REFERENCED ROUTINE
2326 ; PROCESS: INPUT THE UPP STATUS BYTE
2327 ; INPUT:
2328 ; OUTPUT: A-REG CONTAINS THE UPP STATUS BYTE
2329 ; MODIFIED:
2330 ; STACK USAGE: 8 BYTES
2331 ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
2332 UPPS:
FDC E C5     2333      PUSH     B          ; SAVE BC
FDCF 0618    2334      MVI     B,RPSTC    ; B CONTAINS STATUS COMMAND
FDD1 CDB5FF  2335      CALL    P1DDR1    ; GET UPP STATUS BYTE
FDD4 F5     2336      PUSH     PSW        ; SAVE IT ON THE STACK
FDD5 CDBBFF  2337      CALL    P1DDR2    ; GET PIO DEVICE STATUS BYTE AND IGNORE IT
FDD8 F1     2338      POP     PSW        ; A NOW CONTAINS UPP STATUS BYTE
FDD9 C1     2339      POP     B          ; RESTORE BC
FDDA C9     2340      RET
2341 ;*****
2342 ;*
2343 ;* END OF I/O SUBROUTINES, BEGINNING OF MONITOR SUBROUTINES
2344 ;*
2345 ;*****
2346 ;- - - - -
2347 ; 'BYTE' - ENTERED VIA CALL FROM 'R' COMMAND
2348 ; PROCESS: READ TWO 8-BIT ASCII CHARACTERS, DECODE INTO ONE 8-BIT BINARY WORD
2349 ; INPUT: D CONTAINS RUNNING CHECKSUM
2350 ; OUTPUT: DECODED BYTE IN A-REG, RUNNING CHECKSUM IN D-REG, ZERO BIT SET OR RESET
2351 ; MODIFIED: A,F,C,D
2352 ; STACK USAGE:
2353 BYTE:
FDD B C5     2354      PUSH     B          ; SAVE B,C

```

LOC	OBJ	SEQ	SOURCE STATEMENT
FDDC	CD58FF	2355	CALL RIX ; READ ONE ASCII CHAR FROM TAPE, PUT IN A-REG
FDDF	CD98FE	2356	CALL NIBBLE ; CONVERT 8-BIT ASCII TO 4-BIT HEXADECIMAL VALUE
FDE2	07	2357	RLC ; SHIFT FOUR PLACES TO THE LEFT
FDE3	07	2358	RLC
FDE4	07	2359	RLC
FDE5	07	2360	RLC ; MOVE HEX CHAR TO 4 MSB OF A-REG
FDE6	4F	2361	MOV C,A ; STORE TEMPORARILY IN C
FDE7	CD58FF	2362	CALL RIX ; GET ANOTHER ASCII CHAR FROM READER
FDEA	CD98FE	2363	CALL NIBBLE ; CONVERT TO 4 BIT HEX; NOW LSB OF A-REG
FDED	81	2364	ORA C ; ASSEMBLE IT ALL TOGETHER
FDEE	4F	2365	MOV C,A ; STORE IT TEMPORARILY IN C
FDEF	82	2366	ADD D ; UPDATE CHECKSUM (ZERO BIT IS SET/RESET)
FDF0	37	2367	MOV D,A ; D CONTAINS UPDATED CHECKSUM
FDF1	79	2368	MOV A,C ; LOAD THE CONVERTED WORD
FDF2	C1	2369	POP B
FDF3	C9	2370	RET ; RETURN
		2371	; //////////////////////////////////////
		2372	; 'CONV' - ENTERED VIA CALLS FROM 'DBYTE', 'MXD', 'PBYTE' ROUTINES
		2373	; PROCESS: CONVERT 4 BIT HEX VALUE TO ASCII CHARACTER
		2374	; INPUT: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E, OR F IN HEX IN A-REG
		2375	; OUTPUT: 30H,...,39H,41H,...,46H IN C-REG
		2376	; MODIFIED: A, FLAGS, C
		2377	; STACK USAGE:
		2378	;
		2379	CONV:
FDF4	E60F	2380	ANI 0FH ; ONLY 4 LSB ARE SIGNIFICANT, SO MASK 4 MSB
FDF6	C690	2381	ADI 90H ; SET UP A-REG SO THAT A-F CAUSE CARRY
FDF8	27	2382	DAA
FDF9	CE40	2383	ACI 40H ; ADD IN CARRY AND ADJUST UPPER NIBBLE
FDFB	27	2384	DAA
FDFC	4F	2385	MOV C,A ; STORE CONVERTED RESULT IN C-REG
FDFD	C9	2386	RET ; RETURN
		2387	; //////////////////////////////////////
		2388	; 'CRLF' - ENTERED VIA CALLS FROM 'G', 'H', 'Q', 'R', 'U', 'X' COMMANDS AND
		2389	; 'START' ROUTINE
		2390	; PROCESS: TYPE CARRIAGE RETURN AND LINE FEED ON LOCAL CONSOLE
		2391	; INPUT:
		2392	; OUTPUT:
		2393	; MODIFIED:
		2394	; STACK USAGE:
		2395	CRLF:
FDFE	CDDEFC	2396	CALL CONC ; OUTPUT <CR> ON CONSOLE
FE01	0D	2397	DB CR
FE02	CDDEFC	2398	CALL CONC ; OUTPUT <LF> ON CONSOLE
FE05	0A	2399	DB LF
FE06	C9	2400	RET
		2401	; //////////////////////////////////////
		2402	; 'DADR' - ENTERED VIA CALL FROM 'D' COMMAND
		2403	; PROCESS: PRINT CONTENTS OF HL IN HEX FORMAT ON LIST DEVICE
		2404	; INPUT: HL CONTAINS <LOW ADDRESS> OF 'D' COMMAND
		2405	; OUTPUT:
		2406	; MODIFIED: A
		2407	; STACK USAGE:
		2408	DADR:
FE07	7C	2409	MOV A,H ; PRINT MSB OF LOW ADDRESS

```

LOC  OBJ          SEQ      SOURCE STATEMENT
FE08 C00CFE      2410      CALL   DBYTE
FE08 7D          2411      MOV    A,L           ; PRINT LSB OF LOW ADDRESS
                2412 ;*****JMP   DBYTE
                2413 ;////////////////////////////////////
                2414 ; 'DBYTE' - ENTERED VIA CALLS FROM 'D' COMMAND AND 'DADR' ROUTINE
                2415 ;           ENTERED VIA FALL-THRU FROM 'DADR' ROUTINE
                2416 ; PROCESS: LIST A BYTE ON THE LIST DEVICE AS TWO ASCII CHARACTERS
                2417 ; INPUT: A CONTAINS THE BYTE TO BE LISTED
                2418 ; OUTPUT:
                2419 ; MODIFIED:
                2420 ; STACK USAGE:
                2421 DBYTE:
FE0C F5          2422      PUSH   PSM           ; SAVE A COPY OF A-REG
FE0D OF          2423      RRC
FE0E OF          2424      RRC
FE0F OF          2425      RRC
FE10 OF          2426      RRC           ; WANT TO LOOK ONLY AT BITS 4-7 OF A-REG
FE11 CDF4FD      2427      CALL   CONV          ; CONVERT 4 MSB OF ORIGINAL A-REG TO 1 ASCII CHAR
FE14 CD14FD      2428      CALL   LOM           ; OUTPUT ON LIST DEVICE
FE17 F1          2429      POP    PSM          ; RETRIEVE ORIGINAL VALUE
FE18 CDF4FD      2430      CALL   CONV          ; CONVERT 4 LSB OF ORIGINAL A-REG TO 1 ASCII CHAR
FE1B C314FD      2431      JMP    LOM           ; OUTPUT ON LIST DEVICE
                2432 ;////////////////////////////////////
                2433 ; 'DELAY' - ENTERED VIA CALL FROM 'RI' ROUTINE
                2434 ; PROCESS: 1.0 MS. DELAY
                2435 ; INPUT: ONEMS
                2436 ; OUTPUT: ROUTINE IDLES FOR 1.0 MS.
                2437 ; MODIFIED: C, FLAGS
                2438 ; STACK USAGE: 2 BYTES
                2439 DELAY:
FE1E 0E70        2440      MVI    C,ONEMS      ; LOAD 1 MS.CONSTANT (USE 3BH IN ICE ENVIRONMENT)
                2441 DLY1:
FE20 0D          2442      DCR    C           ; DECREMENT COUNTER
FE21 C220FE      2443      JNZ    DLY1         ; JUMP IF NOT EXPIRED
FE24 C9          2444      RET              ; RETURN
                2445 ;////////////////////////////////////
                2446 ; 'DREG' - ENTERED VIA CALL FROM 'X' COMMAND
                2447 ; PROCESS: DISPLAY THE CONTENTS OF A USER REGISTER
                2448 ; INPUT: HL POINTS TO CHARACTER IN ACTBL OF 'X' COMMAND
                2449 ; OUTPUT: HL POINTS TO NEXT CHARACTER IN ACTBL,
                2450 ;           DE CONTAINS ADDRESS OF REGISTER LOCATION
                2451 ;           B CONTAINS REGISTER PRECISION
                2452 ; MODIFIED:
                2453 ; STACK USAGE:
                2454 DREG:
FE25 23          2455      INX    H           ; HL POINTS TO LOCATION ENTRY IN ACTBL OF 'X' COMMAND
FE26 5E          2456      MOV    E,M         ; INCREMENT HL TO POINT AT DISPLACEMENT
FE27 3A0500      2457      LDA    MEMTOP+1
FE2A 57          2458      MOV    D,A         ; D := MSB OF ADDRESS OF TOP PAGE OF MEMORY
                2459 ; DE POINTS TO THAT PART OF THE EXIT TEMPLATE
                2460 ;           CONTAINING SAVED REGISTER VALUES
FE2B 23          2461      INX    H           ; HL POINTS TO PRECISION IN ACTBL
FE2C 46          2462      MOV    B,M         ; PRECISION, 0=8 BITS, 1=16 BITS
FE2D 23          2463      INX    H           ; POINT AT NEXT REGISTER IDENTIFIER
FE2E 1A          2464      LDAX  D             ; 8/16 BIT DISPLAY AND MODIFICATION
    
```

LOC	OBJ	SER	SOURCE STATEMENT
FE2F	CD58FE	2465	CALL    LBYTE                    ; MSB OF 16 BIT REG, ALL OF 8 BIT REG
FE32	05	2466	DCR     B                        ; TEST PRECISION
FE33	FB	2467	RM                                ; 8 BIT DISPLAY, RETURN
FE34	1B	2468	DCK     D
FE35	1A	2469	LDAX    D
FE36	C358FE	2470	JMP     LBYTE                    ; LSB OF 16 BIT REG
		2471	////////////////////////////////////
		2472	; 'EXPR' - ENTERED VIA CALLS FROM 'D','E','F','H','M','R','W' COMMANDS
		2473	; PROCESS: EVALUATE EXPRESSION "<EXPR>,<EXPR>,<EXPR>"
		2474	; INPUT: C-REG CONTAINS THE NUMBER OF PARAMETERS REQUIRED (1,2, OR 3)
		2475	; OUTPUT: STACK CONTAINS THE PARAMETERS IN REVERSE ORDER
		2476	; MODIFIED: F,C,H,L,SP
		2477	; STACK USAGE:
		2478	EXPR:
FE39	CD74FE	2479	CALL    PARAM                    ; GET A HEXADECIMAL PARAMETER, RETURNED IN HL
FE3C	E3	2480	XTHL                              ; PUT THE PARAMETER IN THE STACK; HL NOW
		2481	;                    CONTAINS RETURN ADDRESS OF CALL TO 'EXPR'
FE3D	E5	2482	PUSH    H                        ; PUT RETURN ADDRESS ON TOP OF STACK
FE3E	0D	2483	DCR     C                        ; DECREMENT PARAMETER COUNT; CARRY BIT UNAFFECTED
FE3F	D246FE	2484	JNC     EXD                      ; JUMP IF COMMA ENTERED (PARAM CALLS PCHK)
FE42	C247F8	2485	JNZ     ERROR                    ; INCORRECT PARAM COUNT
FE45	C9	2486	RET
		2487	EXD:
FE46	C239FE	2488	JNZ     EXPR                    ; GET ANOTHER PARAMETER
FE49	C347F8	2489	JMP     ERROR                    ; NOT TERMINATED WITH CR
		2490	////////////////////////////////////
		2491	; 'HILO' - ENTERED VIA CALLS FROM 'D','F','M','W' COMMANDS
		2492	; PROCESS: COMPARE HL WITH DE
		2493	; INPUT: ADDRESS VALUES IN HL AND DE
		2494	; OUTPUT: IF HL <= DE THEN CARRY = 0;
		2495	;                    IF HL > DE THEN CARRY = 1
		2496	; MODIFIED: HL,A,F
		2497	; STACK USAGE:
		2498	HILO:
FE4C	23	2499	INX     H                        ; INCREMENT HL ADDRESS
FE4D	7C	2500	MOV     A,H                      ; TEST FOR HL = 0
FE4E	B5	2501	ORA     L                        ; ZERO BIT SET IF H=L=00, I.E. HL MUST
		2502	;                    HAVE BEEN FFFFH
FE4F	37	2503	STC                                ; CARRY := 1
FE50	C8	2504	RZ
FE51	7B	2505	MOV     A,E                      ; DE - HL, SET/RESET CARRY
FE52	95	2506	SUB     L                        ; (LSB OF HIGH ADDR) - (MSB OF LOW ADDR)
FE53	7A	2507	MOV     A,D                      ;
FE54	9C	2508	SBB     H                        ; (MSB OF HIGH ADDR) - (MSB OF LOW ADDR)
FE55	C9	2509	RET                                ; RETURN
		2510	////////////////////////////////////
		2511	; 'LADR' - ENTERED VIA CALLS FROM 'H' COMMAND AND 'RESTART' ROUTINE
		2512	; PROCESS: PRINT CONTENTS OF HL IN HEX ON LOCAL CONSOLE DEVICE
		2513	; INPUT: HL CONTAINS THE HEX VALUE TO BE OUTPUT(16 BITS)
		2514	; OUTPUT:
		2515	; MODIFIED: H,L,A
		2516	; STACK USAGE:
		2517	LADR:
FE56	7C	2518	MOV     A,H
FE57	CD58FE	2519	CALL    LBYTE                    ; PRINT 8 MSB OF HEX VALUE ON CONSOLE

LOC	OBJ	SEQ	SOURCE STATEMENT
FE5A	7D	2520	MOV    A,L
		2521	*****JMP    LBYTE                    ; PRINT 8 LSB OF HEX VALUE ON CONSOLE
		2522	////////////////////////////////////
		2523	;'LBYTE' - ENTERED VIA CALLS FROM 'S' COMMAND AND 'DREG', 'LADR' ROUTINES
		2524	;                    ENTERED VIA JUMP FROM 'DREG' ROUTINE
		2525	;                    ENTERED VIA FALL-THRU FROM 'LADR' ROUTINE
		2526	; PROCESS: LIST A BYTE AS TWO ASCII CHARACTERS
		2527	; INPUT: A-REG CONTAINS THE 8 BITS TO BE CONVERTED TO ASCII
		2528	; OUTPUT:
		2529	; MODIFIED: A,F
		2530	; STACK USAGE: 6 BYTES
		2531	LBYTE:
FE5B	F5	2532	PUSH    PSW                    ; SAVE A-REG
FE5C	0F	2533	RRC
FE5D	0F	2534	RRC
FE5E	0F	2535	RRC
FE5F	0F	2536	RRC                    ; LOOK ONLY AT 4 MSB OF THE BYTE VALUE
FE60	CD64FE	2537	CALL    HKD                    ; CONVERT IT TO ONE ASCII CHAR AND OUTPUT IT
FE63	F1	2538	PDP    PSW                    ; RETRIEVE ORIGINAL VALUE
		2539	*****JMP    HKD                    ; CONVERT 4 LSB OF BYTE TO ASCII AND OUTPUT IT
		2540	////////////////////////////////////
		2541	;'HKD' - ENTERED VIA CALL FROM 'LBYTE' ROUTINE
		2542	;                    ENTERED VIA FALL-THRU FROM 'LBYTE' ROUTINE
		2543	; PROCESS: CONVERT 4 LSB IN A-REG INTO ONE ASCII CHAR IN A-REG, PRINT IT
		2544	;                    ON LOCAL CONSOLE DEVICE
		2545	; INPUT: NIBBLE TO BE CONVERTED IS IN BITS 0-3 OF A-REG
		2546	; OUTPUT:
		2547	; MODIFIED: A-REG
		2548	; STACK USAGE:
		2549	HKD:
FE64	CDF4FD	2550	CALL    CONV                    ; CONVERT 4 BITS TO ONE 8-BIT ASCII CHAR
FE67	C395FC	2551	JMP    CON                    ; OUTPUT ON LOCAL CONSOLE
		2552	////////////////////////////////////
		2553	;'LCRLF' - ENTERED VIA CALL FROM 'D' COMMAND
		2554	; PROCESS: PRINT <CR>,<LF> ON LIST DEVICE
		2555	; INPUT:
		2556	; OUTPUT:
		2557	; MODIFIED: C
		2558	; STACK USAGE: 4 BYTES
		2559	LCRLF:
FE6A	0E0D	2560	MVI    C,CR
FE6C	CD14FD	2561	CALL    LOM                    ; OUTPUT <CR> TO LIST DEVICE
FE6F	0E0A	2562	MVI    C,LF
FE71	C314FD	2563	JMP    LOM                    ; OUTPUT <LF> TO LIST DEVICE
		2564	////////////////////////////////////
		2565	;'PARAM' - ENTERED VIA CALLS FROM 'G','S' COMMANDS AND 'EXPR' ROUTINE
		2566	;'PAD' - ENTERED VIA CALLS FROM 'G','S','X' COMMANDS
		2567	; PROCESS: COLLECT A HEXADECIMAL PARAMETER
		2568	; INPUT:
		2569	; OUTPUT: HEXADECIMAL PARAMETER IN HL
		2570	; MODIFIED: A,F,B,H,L
		2571	; STACK USAGE:
		2572	PARAM:
FE74	CDC5FE	2573	CALL    PCHK                    ; GET FIRST CHARACTER
FE77	CA47FB	2574	JZ    ERROR                    ; DISALLOW NULL PARAMETERS



LOC	OBJ	SEQ	SOURCE STATEMENT
		2575	PA0:
FE7A	210000	2576	LXI    H,0                    ; INITIALIZE HL := 0000
		2577	PA1:
FE7D	47	2578	MOV    B,A                    ; SAVE CHAR IN CASE IT'S A DELIMITER
FE7E	CD98FE	2579	CALL   NIBBLE                ; CONVERT THE ASCII CHARACTER TO HEX; MUST BE
		2580	;    0-9,A-F; IF NOT THE CARRY BIT IS SET
FE81	DA90FE	2581	JC     PA2                    ; NOT LEGAL CHAR, TREAT AS DELIMITER
FE84	29	2582	DAD    H                     ; *2
FE85	29	2583	DAD    H                     ; *4
FE86	29	2584	DAD    H                     ; *8
FE87	29	2585	DAD    H                     ; *16 --- SHIFT THE OLD HEX VALUES 4 PLACES TO LEFT
FE88	B5	2586	ORA    L                     ; PUT NEW HEX VALUE IN 4 LSB OF L-REG
FE89	6F	2587	MOV    L,A
FE8A	CD61FF	2588	CALL   TI                    ; GET SUBSEQUENT CHARACTERS
FE8D	C37DFE	2589	JMP    PA1                    ; DECODE NEXT CHARACTER
		2590	PA2:
FE9D	78	2591	MOV    A,B                    ; A := B := DELIMITER CHARACTER
FE91	CDC8FE	2592	CALL   P2C                    ; IS IT A VALID DELIMITER?
FE94	C247F8	2593	JNZ    ERROR                 ; JUMP TO ERROR IF IT ISN'T
FE97	C9	2594	RET
		2595	////////////////////////////////////
		2596	; 'NIBBLE' - ENTERED VIA CALLS FROM 'BYTE','PARAM','PAD' ROUTINES
		2597	; PROCESS: DECODE 8-BIT ASCII CHAR IN A-REG INTO 4-BIT HEX DIGIT IN A-REG.
		2598	;            FILTER OUT ALL CHARACTERS NOT IN THE ASCII CODING SEQUENCE
		2599	;            0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F.
		2600	; INPUT: 8-BIT ASCII CHAR IN A-REG
		2601	; OUTPUT: VALID HEX EQUIVALENT IN A-REG AND CARRY = 0, OTHERWISE
		2602	;            GARBAGE IN A-REG AND CARRY = 1 (INDICATING ILLEGAL CHARACTER)
		2603	; MODIFIED: A, FLAGS
		2604	; STACK USAGE: 2 BYTES
		2605	NIBBLE:
FE98	D630	2606	SUI    '0'                    ; IF THE ASCII CHAR IS BETWEEN 00 AND 2FH,
FE9A	D8	2607	RC                            ;    THEN RETURN WITH CARRY = 1
FE9B	C6E9	2608	ADI    '0' - 'G'              ; IF THE ASCII CHAR IS GREATER THAN 46H,
FE9D	D8	2609	RC                            ;    THEN RETURN WITH CARRY = 1
FE9E	C606	2610	ADI    6                     ; ORIGINAL ASCII CHAR WAS BETWEEN 30H AND 46H INCL.
FEA0	F2A6FE	2611	JP     M10                    ; JUMP IF IT WAS 41H THRU 46H (I.E. A-F)
FEA3	C607	2612	ADI    7                     ; ORIGINAL ASCII CHAR WAS BETWEEN 30H AND 40H INCL.
FEA5	D8	2613	RC                            ; RETURN WITH CARRY = 1 IF ASCII CHAR WAS
		2614	;    BETWEEN 3AH AND 40H INCLUSIVE
		2615	M10:                         ; VALID VALUE: 30H-39H,41H-46H
FEA6	C60A	2616	ADI    10                    ; A-REG NOW CONTAINS HEX EQUIV.(0-9,A-F)
FEA8	B7	2617	ORA    A                     ; CLEAR ERROR FLAG (I.E. RESET CARRY BIT)
FEA9	C9	2618	RET                          ; RETURN WITH VALID VALUE
		2619	////////////////////////////////////
		2620	; 'PADR' - ENTERED VIA CALLS FROM 'E','M' COMMANDS
		2621	; PROCESS: PUNCH CONTENTS OF HL IN HEX ON PUNCH DEVICE
		2622	; INPUT: HL CONTAINS 8-BIT LOAD ADDRESS
		2623	; OUTPUT:
		2624	; MODIFIED: A
		2625	; STACK USAGE: 4 BYTES
		2626	PADR:
FEAA	7C	2627	MOV    A,H                    ; A := MSB OF LOAD ADDRESS
FEAB	CDAFFE	2628	CALL   PBYTE                 ; EMIT FRAMES 3 & 4
FEAE	7D	2629	MOV    A,L                    ; A := LSB OF LOAD ADDRESS

```

LOC OBJ      SEQ      SOURCE STATEMENT
2630 ;*****JMP      PBYTE      ; EMIT FRAMES 5 & 6
2631 ;////////////////////
2632 ; 'PBYTE' - ENTERED VIA CALLS FROM 'E','M' COMMANDS AND 'PADR' ROUTINE
2633 ;          ENTERED VIA FALL-THRU FROM 'PADR' ROUTINE
2634 ; PROCESS: PUNCH A BYTE AS 2 ASCII CHARACTERS
2635 ; INPUT: A-REG CONTAINS BYTE TO BE CONVERTED, D CONTAINS RUNNING CHECKSUM
2636 ; OUTPUT: D CONTAINS UPDATED CHECKSUM
2637 ; MODIFIED: A,F,D,E
2638 ; STACK USAGE:
2639 PBYTE:
FEAF 5F      2640      MOV      E,A          ; SAVE BYTE TO BE CONVERTED IN E-REG
FEB0 0F      2641      RRC
FEB1 0F      2642      RRC
FEB2 0F      2643      RRC
FEB3 0F      2644      RRC
FEB4 CDF4FD  2645      CALL    CDNV          ; LOOK ONLY AT 4 MSB OF THE BYTE
FEB7 CDE9FC  2646      CALL    PD           ; CONVERT IT TO 1 ASCII CHARACTER
FEB8 78      2647      MOV      A,E          ; PUNCH IT
FEBB CDF4FD  2648      CALL    CDNV          ; NOW LOOK ONLY AT 4 LSB OF BYTE
FEBE CDE9FC  2649      CALL    PD           ; CONVERT IT TO ONE ASCII CHAR
FEC1 78      2650      MOV      A,E          ; PUNCH IT
FEC2 82      2651      ADD      D            ; UPDATE THE RUNNING CHECKSUM
FEC3 57      2652      MOV      D,A          ; STORE IT BACK IN THE D-REG
FEC4 C9      2653      RET              ; RETURN
2654 ;////////////////////
2655 ; 'PCHK' - ENTERED VIA CALLS FROM 'G','S','X' COMMANDS AND 'PARAM' ROUTINE
2656 ; 'P2C' - ENTERED VIA CALLS FROM 'PARAM','PAD' ROUTINES
2657 ; PROCESS: TEST FOR NULL INPUT PARAMETER (LOOK FOR SPACE, COMMA, OR (CR))
2658 ; INPUT:
2659 ; OUTPUT: CHARACTER IN A-REG
2660 ;          IF SPACE OR COMMA, THEN ZERO = 1 AND CARRY = 0
2661 ;          IF (CR),          THEN ZERO = 1 AND CARRY = 1
2662 ;          IF NONE OF ABOVE, THEN ZERO = 0 AND CARRY = 0
2663 ; MODIFIED: A, FLAGS
2664 ; STACK USAGE: 4 BYTES
2665 PCHK:
FEC5 CD61FF  2666      CALL    TI           ; GET A CHARACTER
2667 P2C:
FEC8 FE20    2668      CPI      ' '
FECA CB      2669      RZ              ; IF SPACE, THEN ZERO = 1 & CARRY = 0
FECB FE2C    2670      CPI      ','
FECD CB      2671      RZ              ; IF COMMA, THEN ZERO = 1 & CARRY = 0
FECE FE0D    2672      CPI      CR
FED0 37      2673      STC
FED1 CB      2674      RZ              ; IF (CR), THEN ZERO = 1 & CARRY = 1
FED2 3F      2675      CMC
FED3 C9      2676      RET              ; IF NONE OF THE THREE, THEN ZERO=CARRY=0
2677 ;////////////////////
2678 ;// 'RESTART' - ENTERED VIA JUMP FROM LOCATION 0 /
2679 ;// PROCESS: BREAKPOINT/INTERRUPT/RESTART PROCESSING /
2680 ;// INPUT: /
2681 ;// OUTPUT: /
2682 ;// MODIFIED: /
2683 ;// EXPLANATION: /
2684 ;// THIS ROUTINE IS ENTERED VIA A RESTART 0 (RST 0) INSTRUCTION. THE /
    
```

LOC	OBJ	SEQ	SOURCE STATEMENT
2685	;/		INSTRUCTION IS ENCOUNTERED EITHER IN THE USER PROGRAM (AS A BREAKPOINT)
2686	;/		OR IS INPUT VIA A LOCAL CONSOLE INTERRUPT (I.E. USER HAS ACTIVATED THE
2687	;/		INTERRUPT 0 SWITCH). THIS ROUTINE SAVES THE STATE OF THE CALLING
2688	;/		PROCESS AND TURNS CONTROL OVER TO THE MONITOR. THIS IS DONE IN THE
2689	;/		FOLLOWING MANNER:
2690	;/	1.	THE USER ENVIRONMENT IS SAVED BY PUSHING THE REGISTERS ON TOP
2691	;/		OF THE USER'S OWN WORK STACK.
2692	;/	2.	PROGRAM THE 0259 WITH THE MONITOR'S OWN INTERRUPT MASK REGISTER.
2693	;/	3.	THE MONITOR'S EXIT TEMPLATE IS FOUND AND THE REGISTER VALUES
2694	;/		REPRESENTING THE USER'S STATE ARE POPPED OFF THE USER WORK STACK
2695	;/		AND STORED IN THE APPROPRIATE PLACES IN THE EXIT TEMPLATE.
2696	;/	4.	TEST TO SEE IF THE POINT AT WHICH USER PROGRAM INTERRUPTION
2697	;/		OCCURRED (VALUE OF PROGRAM COUNTER) COINCIDES WITH A BREAKPOINT
2698	;/		ADDRESS.
2699	;/	A.	IF IT DOESN'T, THEN RESTART CODE WAS ENTERED VIA A CONSOLE
2700	;/		INTERRUPT SO SEND EDI TO THE 0259.
2701	;/	B.	IF IT DOES, THEN PROGRAM THE EXIT CODE TO 1) LOAD THE CORRECT
2702	;/		H AND L VALUES AND TO 2) JUMP TO THE ADDRESS INDICATED BY THE PC
2703	;/		(PUSHED ON STACK AT TIME OF RST 0 INSTRUCTION OR WHEN CONSOLE
2704	;/		INTERRUPT). ALSO, RESTORE THE TRAP VALUES AT THE PROPER
2705	;/		TRAP ADDRESSES.
2706	;/	5.	RETURN CONTROL TO THE MONITOR (BY JUMPING TO START).
2707	;/		
2708	;/		////////////////////////////////////
2709	;/		RESTART:
FED4	F3	2710	DI ; DISABLE IF SOFTWARE TRAP
		2711	; SAVE USER'S ENVIRONMENT
FED5	E5	2712	PUSH H ; SAVE H,L
FED6	D5	2713	PUSH D ; SAVE D,E
FED7	C5	2714	PUSH B ; SAVE B,C
FED8	F5	2715	PUSH PSM ; SAVE A,FLAGS
FED9	D1	2716	PDP D ; TEMPORARILY SAVE PSW IN D & E
FEDA	E5	2717	PUSH H ; DUNNY PUSH TO RESERVE SPACE IN STACK FOR
		2718	; CURRENT INTERRUPT MASK AND CONFIGURATION
		2719	; BYTE
FEDB	2A0400	2720	LHLD MENTOP
FEDE	2ECC	2721	MVI L,ILOC-1 AND OFFH; HL NOW POINTS TO CONFIGURATION BYTE IN
		2722	; EXIT CODE IN TOP PAGE OF RAM
FEE0	6E	2723	MOV L,M ; L NOW CONTAINS THIS CONFIGURATION BYTE
FEE1	DBFC	2724	IN SOCP1 ; INPUT CURRENT INTERRUPT MASK REGISTER ---
		2725	; THIS MASK IS THE USER'S, SO SAVE IT
FEE3	67	2726	MOV H,A ; H NOW CONTAINS THIS INTERRUPT MASK
FEE4	E3	2727	XTHL ; THE INTERRUPT MASK AND CONFIGURATION BYTE
		2728	; ARE NOW ON TOP OF THE USER STACK
FEE5	D5	2729	PUSH D ; NOW PUT THE ORIGINAL PSW ON TOP OF THE STACK
FEE6	3EFE	2730	MVI A,NDT INTO ; SET MONITOR'S DEFAULT INTERRUPT MASK
FEEB	D3FC	2731	OUT SOCP1 ; OUTPUT NEW MASK
FEEA	2A0400	2732	LHLD MENTOP
FEED	2ED2	2733	MVI L,EXIT AND OFFH ; HL NOW POINTS TO EXIT CODE AT TOP OF RAM
FEEF	EB	2734	XCHG ; SO NOW DE POINTS TO EXIT CODE AT TOP OF RAM
FEF0	210C00	2735	LXI H,12 ; H := DD, L := 0C (DECIMAL VALUE 12)
FEF3	39	2736	DAD SP ; EFFECT OF THIS IS TO CUT BACK THE USER'S
		2737	; STACK TO WHAT IT WAS BEFORE ENTERING
		2738	; THIS RESTART ROUTINE AND BEFORE THE PC
		2739	; WAS PUSHED ON BY RST 0 OR INTERRUPT.

LOC	OBJ	SEQ	SOURCE STATEMENT
		2740	
FEF4	0605	2741	MVI B,5 ; HL CONTAINS THIS 'OLD' STACK ADDRESS.
		2742	; COUNT FOR TRANSFER OF MACHINE STATE
FEF6	EB	2743	XCHG ; TO EXIT TEMPLATE STORAGE (MOVE THE STACK)
		2744	; HL NOW POINTS TO EXIT CODE AT TOP OF RAM
		2745	; DE NOW POINTS TO USER STACK AS IT WAS
		2746	; PRIOR TO RST 0 OR CONSOLE INTERRUPT.
		2747	RST0 ; -----
		2748	; MOVE THE MACHINE STATE FROM THE USER'S STACK
		2749	; TO THE RESERVED AREA IN THE EXIT TEMPLATE
		2750	; IN TOP PAGE OF RAM.
		2751	; B=5 ! B=4 ! B=3 ! B=2 ! B=1
FEF7	2B	2752	DCX H ; -----
FEF8	72	2753	MOV H,D ; SLOC=MSB(SP)!ALOC=A !ILOC=INT!BLOC=B!DLOC=D
FEF9	2B	2754	DCX H ;
FEFA	73	2755	MOV H,E ; =LSB(SP)!FLOC=FLG! =FLG!CLOC=C!ELOC=E
FEFB	D1	2756	POP D ; DE=AF !DE=INT,F!DE=BC !DE=DE !DE=HL
FEFC	05	2757	DCR B ; B=4 !B=3 !B=2 !B=1 !B=0
FEFD	C2F7FE	2758	JNZ RST0 ; -----
		2759	; AT THIS POINT, HL POINTS TO THE BASE OF
		2760	; THE MONITOR STACK (TOS) IN TOP PAGE OF
		2761	; RAM. DE CONTAINS THE H & L VALUES THE
		2762	; USER HAD PRIOR TO ENTERING THE RESTART
		2763	; ROUTINE.
FF00	C1	2765	POP B ; BC = OLD PC (PUSHED ON USER STACK BY
		2766	; RST 0 OR INTERRUPT)
FF01	0B	2767	DCX B ; DECREMENT TO POINT AT TRAPPED CODE
FF02	F9	2768	SPHL ; SP NOW POINTS TO TOS (BASE OF MONITOR STACK)
FF03	2A0400	2769	LHLD HENTOP
FF06	2EE2	2770	MVI L,TLOC AND OFFH ; HL NOW POINTS TO TLOC IN TOP PAGE OF RAM
		2771	; I.E. LSB OF TRAP 1 ADDRESS
FF08	7E	2772	MOV A,M ; TEST IF THIS IS A PROGRAMMED RESTART OR A
FF09	91	2773	SUB C ; LOCAL CONSOLE INTERRUPT BY COMPARING THE
		2774	; PC VALUE WITH TRAP 1 ADDRESS
		2775	; A := LSB OF TRAP 1 ADDRESS
FF0A	23	2776	INX H ; HL POINTS TO MSB OF TRAP 1 ADDRESS
FF0B	C213FF	2777	JNZ RSTA ; PC DID NOT MATCH TRAP 1 ADDRESS
FF0E	7E	2778	MOV A,M ; A := MSB OF TRAP 1 ADDRESS
FF0F	98	2779	SBB B
FF10	CA25FF	2780	JZ RST1 ; PC MATCHES TRAP 1 --- A PROGRAMMED RESTART
		2781	RSTA ; REPEAT SAME STEPS AS ABOVE BUT SEE IF PC
		2782	; MATCHES 2ND BREAKPOINT (TRAP 2 ADDRESS)
FF13	23	2783	INX H ; HL POINTS TO TRAP 1 OPCODE VALUE
FF14	23	2784	INX H ; HL POINTS TO LSB OF TRAP 2 ADDRESS
FF15	7E	2785	MOV A,M ; A := LSB OF TRAP 2 ADDRESS
FF16	91	2786	SUB C
FF17	23	2787	INX H ; HL POINTS TO MSB OF TRAP 2 ADDRESS
FF18	C220FF	2788	JNZ RSTB ; PC DID NOT MATCH TRAP 2 ADDRESS
FF1B	7E	2789	MOV A,M ; A := MSB OF TRAP 2 ADDRESS
FF1C	98	2790	SBB B
FF1D	CA25FF	2791	JZ RST1 ; PC MATCHES TRAP 2 --- A PROGRAMMED RESTART
		2792	RSTB ; NOT A PROGRAMMED RESTART, BUT A
FF20	3E20	2793	MVI A,EDI ; CONSOLE INTERRUPT SO SEND EOI TO 8259
FF22	D3FD	2794	OUT SOCP0

LOC	OBJ	SEQ	SOURCE STATEMENT
FF24	03	2795	INX    B                    ; ADJUST PC FOR LOCAL CONSOLE RESTART
		2796	; I.E. GET READY TO POINT PC TO
		2797	; RESUMPTION POINT IN CODE IT WAS
		2798	; EXECUTING WHEN INTERRUPTED
		2799	; BC POINTS TO NEXT INSTR TO BE EXECUTED
		2800	; WHEN CONTROL IS RETURNED TO USER PROGRAM
		2801	RST1:                        ; PROGRAMMED RESTART AT A BREAKPOINT (TRAP)
		2802	; ALSO FALLTHROUGH FROM CONSOLE INTERRUPT
FF25	2A0400	2803	LHLD    MENTOP
FF28	2EDC	2804	MVI    L,LLOC AND OFFH ; HL NOW POINTS TO LLOC IN EXIT CODE IN TOP OF RAM
FF2A	73	2805	MOV    M,E                ; USER'S L VALUE PRIOR TO RESTART IS STORED IN LLOC
FF2B	23	2806	INX    H
FF2C	72	2807	MOV    M,D                ; USER'S H VALUE PRIOR TO RESTART IS STORED IN MLOC
		2808	;
FF2D	2EE0	2809	MVI    L,PLOC-1 AND OFFH; HL POINTS TO LSB OF JMP INSTR IN EXIT CODE
FF2F	71	2810	MOV    M,C                ; SAVE LSB OF USER'S PC
FF30	23	2811	INX    H
FF31	70	2812	MOV    M,B                ; SAVE MSB OF USER'S PC. EFFECT IS TO LOAD THE
		2813	; PROPER ADDRESS INTO THE EXIT TEMPLATE FOR THE
		2814	; JUMP BACK TO THE USER'S PROGRAM.
		2815	;
FR32	C5	2816	PUSH    B
FR33	CDDEFC	2817	CALL    CONC
FR36	23	2818	DB    '0'
FR37	E1	2819	POP    H                 ; RETRIEVE OLD PC FOR DISPLAY
FR38	CD56FE	2820	CALL    LADR              ; DISPLAY PC
		2821	;
		2822	; CLEAR TRAPS
FR3B	2A0400	2823	LHLD    MENTOP
FR3E	2EE2	2824	MVI    L,TLOC AND OFFH ; HL NOW POINTS TO TLOC IN TOP PAGE OF RAM
FF40	1602	2825	MVI    D,2                ; SET COUNT FOR TWO TRAPS
		2826	RST2:
FF42	4E	2827	MOV    C,M                ; C := LSB OF TRAP ADDRESS
FF43	AF	2828	XRA    A
FF44	77	2829	MOV    M,A                ; ZERO OUT LSB OF TRAP ADDRESS
FF45	23	2830	INX    H
FF46	46	2831	MOV    B,M                ; B := MSB OF TRAP ADDRESS
FF47	77	2832	MOV    M,A                ; ZERO OUT MSB OF TRAP ADDRESS
FF48	23	2833	INX    H                 ; HL NOW POINTS TO TRAP VALUE
FF49	79	2834	MOV    A,C                ; BC CONTAINS THE TRAP ADDRESS
FF4A	8D	2835	ORA    B                 ; TEST FOR VALID TRAP
FF4B	CA50FF	2836	JZ    RST3                ; TRAP ADDRESS IS 0, SO NO TRAP TO RESTORE
FF4E	7E	2837	MOV    A,M                ; GET OPCODE BYTE, I.E. TRAP VALUE
FF4F	02	2838	STAX    B                ; PUT IT BACK IN CORRECT PLACE IN USER PROGRAM,
		2839	; I.E. REPLACE THE RST 0 INSTR WITH ORIGINAL
		2840	; OPCODE.
		2841	RST3:
FF50	23	2842	INX    H                 ; POINT TO TRAP 2 ADDRESS IF D=2
FF51	15	2843	DCR    D
FF52	C242FF	2844	JNZ    RST2               ; REPEAT FOR TRAP 2
FF55	C355F8	2845	JMP    START             ; ENTER MONITOR (INTERRUPTS STILL DISABLED)
		2846	////////////////////////////////////
		2847	; 'RIX' - ENTERED VIA CALLS FROM 'R' COMMAND AND 'BYTE' ROUTINE
		2848	; PROCESS: GET A CHARACTER FROM READER, MASK OFF PARITY BIT
		2849	; INPUT:



LOC	OBJ	SER	SOURCE STATEMENT
		2905	; 'IOCDR1' - ENTERED VIA CALLS FROM 'CI','CSTS' ROUTINES
		2906	; PROCESS: GET DEVICE STATUS OR GET DATA FROM PERIPHERAL
		2907	; INPUT: B CONTAINS THE COMMAND (STATUS REQUEST OR INPUT DATA REQUEST)
		2908	; OUTPUT: A CONTAINS THE REQUESTED INFORMATION
		2909	; MODIFIED: A,FLAGS,B
		2910	; STACK USAGE:
		2911	IOCDR1:
FF7F	CDA6FF	2912	CALL IOCCOM ; OUTPUT 'GET DEVICE STATUS COMMAND' OR
		2913	; 'INPUT DATA COMMAND' TO IOC CONTROL
		2914	; PORT
		2915	IOCXXX:
FF82	DBC1	2916	IN IOCS ; INPUT DBB STATUS
FF84	E607	2917	ANI IBF OR OBF OR FD; MASK OFF STATUS FLAGS
FF86	FE01	2918	CPI OBF ; TEST FOR SLAVE DONE; SOMETHING FOR THE MASTER
FF88	C282FF	2919	JNZ IOCXXX ; IF NOT, CONTINUE TO LOOP
FF8B	DBCD	2920	IN IOCI ; OTHERWISE, INPUT THE DATA FROM THE DBB
FF8D	F5	2921	PUSH PSM ; SAVE A-REG
FF8E	3E05	2922	MVI A,ENABL ; ENABLE INTERRUPTS
FF90	D3FF	2923	OUT CPUC
FF92	F1	2924	PDP PSM ; RESTORE A-REG
FF93	C9	2925	RET
		2926	-----
		2927	; 'IOCDR2' - ENTERED VIA CALLS FROM 'BLK','COM','CO','CRTOUT' ROUTINES
		2928	; PROCESS: OUTPUT DATA TO THE PERIPHERAL DEVICE
		2929	; INPUT: B CONTAINS THE COMMAND TO OUTPUT THE DATA
		2930	; C CONTAINS THE DATA TO BE OUTPUT
		2931	; OUTPUT:
		2932	; MODIFIED: A,FLAGS,B,C
		2933	; STACK USAGE:
		2934	IOCDR2:
FF94	CDA6FF	2935	CALL IOCCOM ; OUTPUT 'OUTPUT DATA COMMAND' TO IOC
		2936	; CONTROL PORT
		2937	IOCYYY:
FF97	DBC1	2938	IN IOCS ; INPUT DBB STATUS
FF99	E607	2939	ANI IBF OR FD OR OBF; TEST FOR SLAVE PROCESSOR READY
FF9B	C297FF	2940	JNZ IOCYYY ; CONTINUE TO LOOP UNTIL IT IS READY
FF9E	79	2941	MOV A,C ; LOAD DATA TO BE WRITTEN
FF9F	D3C0	2942	OUT IOCD ; OUTPUT DATA TO THE DBB
FA01	3E05	2943	MVI A,ENABL ; ENABLE INTERRUPTS
FA03	D3FF	2944	OUT CPUC
FA05	C9	2945	RET
		2946	-----
		2947	; 'IOCCOM' - COMMON ROUTINE TO IOC DRIVERS
		2948	; ENTERED VIA CALLS FROM 'IOCDR1' AND 'IOCDR2'
		2949	; PROCESS: OUTPUT COMMAND TO THE IOC
		2950	; INPUT: B CONTAINS THE COMMAND
		2951	; OUTPUT:
		2952	; MODIFIED: A,FLAGS
		2953	; STACK USAGE:
		2954	IOCCOM:
FA06	3E0D	2955	MVI A,DISABL ; BLOCK ALL INTERRUPTS
FA08	D3FF	2956	OUT CPUC
		2957	IOCZZZ:
FA0A	DBC1	2958	IN IOCS ; INPUT DBB STATUS
FA0C	E607	2959	ANI FD OR IBF OR OBF; TEST FOR SLAVE PROCESSOR IDLE

```

LOC  OBJ          SEQ      SOURCE STATEMENT

FFAE C2AFF        2960      JNZ      IOCZZZ      ; LOOP UNTIL IT IS IDLE
FFB1 78           2961      MOV      A,B        ; LOAD COMMAND
FFB2 D3C1        2962      OUT     IOC        ; OUTPUT COMMAND TO IOC CONTROL PORT
FFB4 C9           2963      RET

2964 ;-----*
2965 ;*
2966 ;*              PARALLEL I/O INTERFACE DRIVERS
2967 ;*
2968 ;-----*
2969 ; 'PIODR1' - ENTERED VIA CALLS FROM 'RI','PO','PC','LO','UPPS'
2970 ; 'PIODR2' - ENTERED VIA CALLS FROM 'UI','UPPS' ROUTINES
2971 ; PROCESS: GET DEVICE STATUS OR GET DATA FROM A PERIPHERAL
2972 ; INPUT: B CONTAINS THE COMMAND (STATUS REQUEST OR INPUT DATA REQUEST)
2973 ; OUTPUT: A CONTAINS THE REQUESTED INFORMATION
2974 ; MODIFIED: A, FLAGS, B
2975 ; STACK USAGE:
2976 PIODR1:
FFB5 CDE4FF      2977      CALL     PIOC0M      ; OUTPUT 'GET DEVICE STATUS COMMAND' OR
2978 ; 'INPUT DATA COMMAND' OR OTHER SUCH
2979 ; COMMAND TO THE PIO CONTROL PORT
2980 PIODR2:
FFB8 3E0D        2981      MVI     A,DISABL    ; BLOCK ALL INTERRUPTS
FFBA D3FF        2982      OUT     CPUC
FFBC DBF9        2983      IN      PIOS        ; INPUT DBB STATUS
FFBE E6D7        2984      ANI     FO OR IBF OR OBF ; MASK OFF STATUS FLAGS
FFC0 FE01        2985      CPI     OBF        ; TEST FOR SLAVE DONE; SOMETHING FOR THE MASTER
FFC2 C2B8FF      2986      JNZ     PIODR2      ; LOOP UNTIL SLAVE IS READY
FFC5 DBF8        2987      IN      PIOC        ; OTHERWISE INPUT THE DATA FROM THE DBB
FFC7 F5          2988      PUSH   PSW        ; SAVE A-REG
FFC8 3E05        2989      MVI     A,ENABL    ; ENABLE INTERRUPTS
FFCA D3FF        2990      OUT     CPUC
FFCC F1          2991      POP     PSW        ; RESTORE A-REG
FFCD C9          2992      RET

2993 ;-----*
2994 ; 'PIODR3' - ENTERED VIA CALLS FROM 'PC','PO','LO','UI','UD' ROUTINES
2995 ; 'PIODR4' - ENTERED VIA CALLS FROM 'UI','UD'
2996 ; PROCESS: OUTPUT DATA TO A PERIPHERAL DEVICE
2997 ; INPUT: B CONTAINS THE COMMAND TO OUTPUT THE DATA
2998 ; C CONTAINS THE DATA TO BE OUTPUT
2999 ; OUTPUT:
3000 ; MODIFIED: A,FLAGS,B, C
3001 ; STACK USAGE:
3002 PIODR3:
FFCE CDE4FF      3003      CALL     PIOC0M      ; OUTPUT 'OUTPUT DATA COMMAND' TO PIO
3004 PIODR4:
FFD1 3E0D        3005      MVI     A,DISABL    ; BLOCK ALL INTERRUPTS
FFD3 D3FF        3006      OUT     CPUC
FFD5 DBF9        3007      IN      PIOS        ; INPUT DBB STATUS
FFD7 E6D7        3008      ANI     FO OR IBF OR OBF ; TEST FOR SLAVE PROCESSOR READY
FFD9 C2D1FF      3009      JNZ     PIODR4      ; LOOP UNTIL IT IS READY
FFDC 79          3010      MOV     A,C        ; LOAD DATA TO BE WRITTEN
FFDD D3F8        3011      OUT     PIOC        ; OUTPUT DATA TO THE DBB
FFDF 3E05        3012      MVI     A,ENABL    ; ENABLE INTERRUPTS
FFE1 D3FF        3013      OUT     CPUC
FFE3 C9          3014      RET
    
```



```

LOC  OBJ          SEQ          SOURCE STATEMENT
-----
3015 ; -----
3016 ; 'PIOCOM' - COMMON ROUTINE OF PIO DRIVERS
3017 ;          ENTERED VIA CALLS FROM 'PIODR1', 'PIODR3', 'RI' ROUTINES
3018 ; INPUT: B CONTAINS THE COMMAND
3019 ; OUTPUT:
3020 ; MODIFIED: A, FLAGS
3021 ; STACK USAGE:
3022 PIOCOM:
FFE4 3E0D      3023      MVI      A,DISABL      ; BLOCK ALL INTERRUPTS
FFE6 D3FF      3024      OUT      CPUC
3025 PIOZZZ:
FFE8 DBF9      3026      IN       PIOS          ; INPUT DBB STATUS
FFEA E607      3027      ANI      FD DR IBF OR OBF; TEST FOR SLAVE PROCESSOR IDLE
FFEC C2E8FF    3028      JNZ      PIOZZZ        ; LOOP UNTIL IT IS IDLE
FFEF 78        3029      MOV      A,B           ; LOAD THE COMMAND
FFF0 D3F9      3030      OUT      PIOC          ; OUTPUT THE COMMAND TO THE PIO CONTROL PORT
FFF2 3E05      3031      MVI      A,ENABL      ; ENABLE INTERRUPTS
FFF4 D3FF      3032      OUT      CPUC
FFF6 C9        3033      RET
3034 ; -----
FFF8          3035      ORG      OFFFDH
FFF8 6D        3036 MNCKSH: DB      06DH      ; CHKSUM MONITOR TO 01EH
FFFE 00        3037      DB      00           ; UNUSED BYTE
FFFF 01        3038      DB      01           ; 0, IF SERIES I MONITOR
3039          3039          ; 1, IF SERIES II MONITOR
3040 ; -----
3041 ;
3042 ; END OF PROGRAM
3043 ;
3044 ; -----
3045      END
    
```

PUBLIC SYMBOLS

EXTERNAL SYMBOLS

USER SYMBOLS

USER	A FCBC	ACHRM	A 007F	ACT	A F913	ACTBL	A FB81	ALOC	A EACF	ALT	A F92B	ALUP1	A F8D3
ALUP2	A F8DB	ALUP3	A F8F2	APT	A F923	ART	A F91B	ASD	A FB8E	AS1	A F8CD	AS2	A F8E5
AS3	A F8F2	ASSIGN	A F8B6	BO110	A 02BA	B2400	A 0020	B9600	A 0007	BASE	A F800	BATCH	A 0002
BBASE	A E800	BCDC	A 0001	BDLY	A EA07	BDLY1	A EA09	BEGIN	A F800	BLK	A FC93	BLOC	A EACB
BOVROF	A 0001	BOVRON	A 0009	BREAK	A FC07	BS0	A E806	BS1	A E83B	BS10	A E8FF	BS11	A E90C
BS12	A E93A	BS13	A E945	BS14	A E967	BS2	A E84E	BS3	A E856	BS4	A E898	BS5	A E8AC
BS6	A E8B2	BS7	A E8C8	BS8	A E8CE	BS9	A E8D6	BSK1	A E97A	BSX10	A E9FC	BSX2	A E990
BSX3	A E99E	BSX4	A E9A9	BSX5	A E9B0	BSK6	A E9CE	BSX8	A E9E2	BSX9	A E9F1	BTCKSH	A EA48
BTDGOF	A 0004	BTDCON	A 000C	BYTE	A FDD8	CCRT	A 0001	CI	A FB8E	C10	A F8D0	C11	A FBE1
C12	A FBEB	C13	A FBEC	C14	A FBFD	C1LOC	A EAEB	CL5	A 0000	CL6	A 0004	CL7	A 0008
CL8	A 000C	CLERR	A 0010	CLOC	A EACA	CMSK	A 00FC	CNOTD	A 0008	CO	A FC9F	COO	A FCB2
COLOC	A EAEB	COM	A FC95	COMC	A FCDE	COND	A 0025	CONC	A 00C1	CONI	A 00C0	CONO	A 00C0
CONS	A 00C1	CONV	A FDF4	COP	A F809	CPUC	A 00FF	CPUS	A 00FE	CR	A 0000	CRLF	A FDFE
CRTC	A 0010	CRTOT1	A FCCA	CRTOT2	A FCD5	CRTOUT	A FCBE	CRTS	A 0011	CSO	A FD53	CS1	A FD6B
CS2	A FD74	CS3	A FD79	CSLOC	A EAFD	CSMEN	A 0008	CSTS	A FD44	CTBL	A F882	CTROP	A 00F0
CTROS	A 0000	CTR1P	A 00F1	CTR1S	A 0040	CTR2P	A 00F2	CTR2S	A 0080	CTTY	A 0000	CUSE	A 0003

DADR	A	FE07	DATE	A	0401	DBYTE	A	FE0C	DECHO	A	0007	DELAY	A	FE1E	DIO	A	F938	DI1	A	F93E
D12	A	F956	DIAGBT	A	E803	DIAGNM	A	E800	DISABL	A	0000	DISAXP	A	0000	DISP	A	F933	DLOC	A	EAC9
DLY1	A	FE20	DPRNT	A	0008	DREG	A	FE25	DSR	A	0080	DSTAT	A	0003	DSTS	A	0078	DTR	A	0002
EL3C	A	EAC8	ENABL	A	0005	ENAXP	A	0008	ENDX	A	E800	ENHM	A	0080	EOF	A	F95F	EDI	A	0020
ERESET	A	0001	ERMSG	A	EA3A	ERROR	A	F847	ETX	A	0003	EXO	A	FE46	EXIT	A	EAD2	EXPR	A	FE39
FO	A	0004	FALSE	A	0000	FBOC	A	0004	FIO	A	F984	FILL	A	F97D	FLOC	A	EACE	FRDY	A	0001
FS1OP	A	00E7	FSTP	A	00F7	GOO	A	F9A4	GO1	A	F9AA	GO2	A	F9BA	GO3	A	F9C2	GO4	A	F9D1
G0T0	A	F98C	HEXN	A	F9D5	HI	A	007A	HILO	A	FE4C	HLOC	A	EADD	HMSK	A	00FF	HXD	A	FE64
ISF	A	0002	ICFG	A	0041	ICNP	A	0001	ICRTI	A	0020	ICRTO	A	0010	ICW1	A	0012	ICW2	A	0000
IICPO	A	00FB	IICP1	A	00FA	ILOC	A	EACD	ILPT	A	0040	INIT	A	E803	INIT10	A	0006	INT0	A	0001
INT1	A	0002	INT2	A	0004	INT3	A	0008	INT4	A	0010	INT5	A	0020	INT6	A	0040	INT7	A	0080
INTA	A	0000	IOBYT	A	0003	IOCC	A	00C1	IOCCOM	A	FFA6	IOCDP1	A	F821	IOCDP2	A	F844	IOCDR1	A	FF7F
IOCDR2	A	FF94	IOCHK	A	FD83	IOCI	A	00C0	IOCO	A	00C0	IOCP0	A	00FB	IOCP1	A	00FA	IOCS	A	00C1
IOCXKX	A	FF82	IOCYYY	A	FF97	IOGZZZ	A	FFAA	IODEF	A	FD94	IOPB	A	EA18	IOSET	A	FD87	IPTP	A	0004
IFTR	A	0008	ITCP	A	00F3	ITIND	A	00FF	ITYYI	A	0002	ITTYO	A	0001	KEYC	A	0012	KINT	A	0014
KRDY	A	0001	KSTS	A	0013	L1LOC	A	EAF4	LADR	A	FE56	LBK	A	00FF	LBYTE	A	FE5B	LCRLF	A	FE6A
LCRT	A	0040	LCT	A	001A	LCTR	A	0000	LERM	A	000E	LF	A	000A	LLOC	A	EADC	LLPT	A	0080
LMSK	A	003F	LO	A	FD1E	LDM	A	FD14	LOWM	A	0079	LPO	A	F033	LPTC	A	0014	LPTRY	A	0001
LSTC	A	0015	LSTE	A	0040	LTBL	A	F903	LTTY	A	0000	LUSE	A	00C0	LVER	A	001B	MEMCHK	A	FD8C
MENTOP	A	0004	MENB	A	0080	NLP	A	E8EE	MNCKSM	A	FFFD	MODE0	A	0000	MODE1	A	0002	MODE2	A	0004
MODE3	A	0006	MODE4	A	0008	MODE5	A	000A	MOVBDT	A	0002	MOVE	A	F9F0	NVO	A	F9F7	NIO	A	FEA6
NIBBLE	A	FE98	NLEADX	A	FA0B	NREGS	A	000C	NUD	A	FA09	NULL	A	FA01	0BF	A	0001	OCW3	A	000B
QHEMS	A	0070	OPCPL	A	0004	P1LOC	A	0010	P2C	A	FEC8	P2LOC	A	EAF7	PAO	A	FE7A	PA1	A	FE7D
PA2	A	FE90	PACIFY	A	0000	PADR	A	FEAA	PARAM	A	FE74	PARML	A	0004	PBYTE	A	FEAF	PAK	A	FEC5
PCOMP	A	0002	PENB	A	0010	PEVEN	A	0020	PGRDY	A	0001	PIDC	A	00F9	PIOCOM	A	FFE4	PIODR1	A	FFB5
PIODR2	A	FFB8	PIODR3	A	FFCE	PIODR4	A	FFD1	PIOI	A	00F8	PI00	A	00F8	PI05	A	00F9	PI0ZZZ	A	FFEB
PLOC	A	EAE1	PMSK	A	00CF	PNIB	A	0010	PO	A	FCE9	PO0	A	FCF7	PO1	A	FD08	PDC	A	FCE5
PFTP	A	0010	PRTM	A	EAOE	PSOCK	A	0020	PSTC	A	0013	PTPRY	A	0001	PTRADV	A	0040	PTRDY	A	0001
PTRREV	A	0060	PTTY	A	0000	PUNC	A	0012	PUSE1	A	0020	PUSE2	A	0030	Q0	A	FA21	Q1	A	FA3D
Q2	A	FA48	QUERY	A	FA14	R16X	A	0002	R1LOC	A	EAE1	R1X	A	0001	R2LOC	A	EAF1	R64X	A	0003
RADCT	A	0028	RDBC	A	0019	RDBCC	A	001A	RDRC	A	0010	RDSTS	A	001C	READ	A	FA52	REDD	A	FA59
RED1	A	FA7B	RED2	A	FA93	RED3	A	FA9E	RED4	A	FAB3	RESET	A	0000	RESTAR	A	FED4	RFR	A	0020
RI	A	FC0F	RIO	A	FC1F	RI1	A	FC2C	RI2	A	FC39	R13	A	FC47	R14	A	FC4C	R14B	A	FC4F
RI5	A	FC58	RI6	A	FC65	RI7	A	FC79	RIB	A	FC82	RIK	A	FF58	RLLB	A	0010	RLLN	A	0030
RLMB	A	0020	RMSK	A	00F3	ROY	A	0010	RPAR	A	0008	RPPC	A	0017	RPSTC	A	0018	RPTR	A	0004
RADY	A	0002	RRSTS	A	0018	RST0	A	FEF7	RST1	A	FF25	RST2	A	FF42	RST3	A	FF50	RSTA	A	FF13
RSTB	A	FF20	RSTC	A	0011	RSTS	A	0078	RTCC	A	04CD	RTOCT	A	00FA	RTS	A	0020	RTTY	A	0000
RUSE1	A	0008	RUSE2	A	000C	RKEN	A	0004	SBCH	A	0008	SICPO	A	00FD	SICP1	A	00FC	SINT	A	000A
SLOC	A	EAD1	SOCPO	A	00FD	SOCPI	A	00FC	SRR	A	0006	SRACK	A	0005	SRQDAK	A	0004	ST1	A	0040
ST15	A	0080	ST2	A	00C0	START	A	F855	START0	A	F851	SU0	A	FAC3	SU1	A	FAD9	SUBS	A	FABF
SYNC	A	0000	SYND	A	0040	SYSTAT	A	0002	TADV	A	0027	TI	A	FF61	TLOC	A	EAE2	TOS	A	EAC8
TOUT	A	00FA	TRAN	A	0009	TRDY	A	0001	TRKO	A	3000	TRKL	A	0000	TRUE	A	FFFF	TTYC	A	00F5
TTY1	A	00F4	TTYIN	A	FBC6	TTYO	A	00F4	TTYOUT	A	FCA7	TTYS	A	00F5	TXBE	A	0004	TKEN	A	0001
UC	A	FF76	UCI	A	0000	UCD	A	0001	UCS	A	0007	UI	A	FDAD	UL1	A	0006	UD	A	FD8E
UP1	A	0004	UP2	A	0005	UPPS	A	FDCE	UR1	A	0002	UR2	A	0003	USCC	A	00F7	USCI	A	00F6
USCO	A	00FA	USCS	A	00F7	USER	A	EAC0	USRST	A	0040	VER	A	000C	VERN	A	0012	VERS	A	EA1F
WDBC	A	0017	WDBCC	A	0018	WPBC	A	0015	WPBCC	A	0016	WPPC	A	0016	WRO	A	FAE5	WR1	A	FAED
WR2	A	FAFB	WR3	A	FB07	WRITE	A	FADD	X	A	FB26	X0	A	FB31	X1	A	FB3F	X2	A	FB42
X3	A	F85F	X4	A	FB60	X5	A	FB6A	X6	A	FB6D	XTBL	A	EAE8	Z	A	FBA6			

ASSEMBLY COMPLETE, NO ERRORS

	1841	1946	1958#	2008	2100	2102	2129	2203
0USER	1841							
ACHRM	384#							
ACT	1146	1160#						
ACTBL	1675	1739#	1753					
ALOC	847#	1740						
ALT	1152	1175#						
ALUP1	1104#	1107						
ALUP2	1109#	1112						
ALUP3	1125#	1128						
APT	1150	1170#						
ART	1148	1165#						
AS0	1084#	1092						
AS1	1087	1095#						
AS2	1115#	1121						
AS3	1118	1124#						
ASSIGN	1045	1079#						
B0110	212#	528						
B2400	211#	517						
B9600	210#							
BASE	927#	928						
BATCH	341#	1163	1838	1985	2004	2117	2198	
BBASE	414#	415	835					
BCDC	209#							
BDLY	546	547	560	561	788#			
BDLY1	790#	792						
BEGIN	775	937#						
BLK	1365	1689	1722	1980#				
BLOC	571	624	689	841#	1741			
BOVROF	87#	439						
BOVRON	88#							
BREAK	1850#	1986	2118					
BS0	416	435#						
BS1	475#	483						
BS10	610#	613						
BS11	587	622#						
BS12	647#	652						
BS13	654#	657						
BS14	670#	674	681					
BS2	487	489#						
BS3	494#	499						
BS4	541#	550						
BS5	545	551#						
BS6	555#	564						
BS7	559	565#						
BS8	549	563	569#					
BS9	568	574#						
BSX1	590	618	627	631	636	662	687#	
BSX10	738	751	779#					
BSX2	692	701#	713	718				
BSX3	706	709#						
BSX4	708	715#						
BSX5	698	721#						
BSX6	735	746#						
BSX8	745	756#						
BSX9	741	748	755	768#	783			
BTKSM	828#							
BTDOGF	89#	761	1002					









PENB	145#					
PEVEN	146#					
PGRDY	127#					
PIOC	260#	3030				
PIOCOM	1922	2977	3003	3022#		
PIODR1	1926	1935	2088	2135	2335	2976#
PIODR2	2298	2337	2980#	2986		
PIODR3	2092	2139	2291	2316	3002#	
PIODR4	2296	2319	2322	3004#	3009	
PIOI	257#	2987				
PIOO	258#	3011				
PIOS	259#	2983	3007	3026		
PIOZZZ	3025#	3028				
PLOC	868#	1312	1564	1750	2809	
PNBK	336#	1149	2079			
PNIB	129#					
PO	942	2077#	2646	2649		
POO	2086#	2090				
POI	2082	2097#				
POC	1229	1411	1622	1658	1660	2070#
PPIP	351#	1172	2081			
PRTH	771	782	798#	803		
PSOCK	128#					
PSIC	269#	2087				
PTPRY	280#	2089				
PTRADV	266#	1921				
PYRDY	279#	1927				
PTRREV	265#					
PTTY	350#	1171				
PUMC	268#	2091				
PUSE1	352#	1173	2098			
PUSE2	353#	1174				
QO	1431#	1464				
Q1	1451#	1458				
Q2	1455	1459#				
QUERY	1061	1424#				
R16X	138#	522	533			
R1LOC	880#	898	1945			
RIX	139#					
R2LOC	882#	899	1947			
R64X	137#					
RADCT	175#	1883				
RDBC	242#	665				
RDBCC	243#					
RDBC	264#	1921	1934			
RDSTS	245#	628	632	653	752	
READ	1062	1483#				
REDO	1487#	1490	1550			
RED1	1515#	1522				
RED2	1542#	1549				
RED3	1496	1552#				
RED4	1562	1568#				
RESET	402#	505	507			
RESTAR	504	506	2709#			
RFR	169#					
RI	940	1839	1865#	2854		
RIO	1877#	1880				







XTBL	873#	896	897	898	899	900	901	902	903	2258
Z	1070	1758#								

CROSS REFERENCE COMPLETE



INTEL CORPORATION, 3065 Bowers Avenue, Santa Clara, California 95051 • (408) 987-8080  
A117/1079/10K NCG