```
                        ;*******8080 CONSOLE FOR DIGITAL EQUIPMENT CORP. KS10*******

                        .SBTTL *CONSOLE ERROR CODES DEFINED*
```

```
                        ;LISTING CONTROLS
                        .NLIST BEX
                        .LIST MD,MC,MEB
                        ;SYMBOL ASSIGNMENTS

020000          RAMST   =       8192.           ;FIRST "RAM" MEMORY ADDRESS
003244          KPAINI  =       1700.           ;VALUE FOR .44 SEC BETWEEN WORD31 CHECKS
000043          KATIMX  =       35.             ;NUMBER OF TIME IN A ROW KA HAS TO BE THE SAME
                                                ;KATIMX .EQ. 14. IS 6 SECONDS, .EQ.35. IS 15 SEC
000300          RUNFP   =       ^O300           ;I/O REG CONTAING CPU "RUN FLOP"
000300          TTYSW   =       ^O300           ;I/O REG FOR TTY FRONT PANEL SWITCHES
000301          SMSTS   =       ^O301           ;I/O REG FOR KS10 STATUS BITS
000301          BOOTSW  =       ^O301           ;I/O REG FOR HARDWARE "BOOT" SWITCH
000302          CARRIER =       ^O302           ;I/O REG FOR READING KLINIK CARRIER
000100          SMPAR   =       ^O100           ;I/O REG FOR ENABLING KS10 PARITY DETECT
000100          RESET   =       ^O100           ;I/O REG FOR ISSUEING KS10 BUS RESET
000101          REFERR  =       ^O101           ;I/O REG FOR READING MOS MEM REFRESH ERR FLAG
000101          LIGHTS  =       ^O101           ;I/O REG FOR WRITING THE PANEL LIGHTS
000101          DTR     =       ^O101           ;I/O REG FOR WRITING THE DATA TERMINAL READY
000102          R.BUSY  =       ^O102           ;I/O REG FOR READING "MEM BUSY" OR "I/O BUSY"
000102          R.RESET =       ^O102           ;I/O REG FOR READY IF RESET WENT TRUE(VIA AC LO)
000204          CRMCTL  =       ^O204           ;I/O REG FOR ACCESSING CRAM
000205          DIAG    =       ^O205           ;I/O REG FOR DIAG FUNCTION BITS
000206          CLKCTL  =       ^O206           ;I/O REG FOR SETTING KS10 CLK CTL BITS
000210          BUSCTL  =       ^O210           ;I/O REG FOR KS10 BUS COMM.
000212          CPUCTL  =       ^O212           ;I/O REG FOR KS10 "RUN,EXECUTE,CONTINUE"
000201          CTYCTL  =       ^O201           ;I/O REG FOR UART CONTROL/STATUS
000200          CTYDAT  =       ^O200           ;I/O REG FOR UART DATA BUFFER
000203          REMCTL  =       ^O203           ;REMOTE UART CONTROL REGISTER
000202          REMDAT  =       ^O202           ;REMOTE UART DATA REGISTER
000114          DTARB   =       ^O114           ;I/O REG FOR DATA CYCLE OF BUS ARB..
000115          BUSARB  =       ^O115           ;I/O REG FOR BUS ARBITRATION SIGNALS
000116          INT2KS  =       ^O116           ;I/O REG FOR INTERRUPTING KS-10
000000          D2835   =       ^O00            ;R DATA BITS 28-35
000001          D2027   =       ^O01            ;R DATA BITS 20-27
000002          D1219   =       ^O02            ;R DATA BITS 12-19
000003          D0411   =       ^O03            ;R DATA BITS 04-11
000103          D0003   =       ^O103           ;R DATA BITS 0-3
000103          A2835   =       ^O103
000105          A2027   =       ^O105
000107          A1219   =       ^O107
000111          A0411   =       ^O111
000113          A0003   =       ^O113
000102          W2835   =       ^O102
000104          W2027   =       ^O104
000106          W1219   =       ^O106
000110          W0411   =       ^O110
000112          W0003   =       ^O112
```

```
                        .SBTTL****NUMERICAL VALUE ASSIGNMENTS
002000          BPI16   =       ^O2000          ;'DENSITY/SLAVE' DEFAULT FOR 1600 BPI TAPES
001000          BPI8    =       ^O1000          ;'DENSITY/SLAVE' DEFAULT FOR 800 BPI TAPES
000377          EOLCH   =       ^O377           ;END-OF-LINE CODE
000020          ARBRESP =       ^O20            ;BIT FOR 'BUS REQ'
000001          DATACK  =       01              ;BIT FOR 'DATA ACKNOWLEDGE'
000100          NONXMEM =       ^O100           ;BIT FOR 'NON-EXISTANT MEMORY'
000020          TRPDEF  =       ^O20            ;BIT SAYS DEFAULT TRAP ENABLE ON
000174          DEFLTE  =       ^O174           ;BITS SAY DEFAULT: CACHE ENABLE, 1MSEC CLK ENABL
                                                ;ALL PARITY ON
100000          BIT15   =       ^O100000        ;THIS IS BIT15(IF U COUNT RIGHT TO LEFT)
000001          BT.BIT  =       1               ;BIT SAYS DOING BOOTSTRAP FROM DISK
000002          MT.BIT  =       2               ;BIT SAYS DOING BOOTSTRAP FROM MAGTAPE
000001          .MODE0  =       1               ;FLAG SAYS KLINIK MODE 0
000002          .MODE1  =       2               ;FLAG SAYS KLINIK MODE 1
000004          .MODE2  =       4               ;FLAG SAYS KLINIK MODE 2
000010          .MODE3  =       ^O10            ;FLAG SAYS KLINIK MODE 3
000020          .MODE4  =       ^O20            ;FLAG SAYS KLINIK MODE 4 (MANUFACTURING MODE)
000003          CARRLOSS =      ^O3             ;INTERRUPT CODE FOR 'KLINIK CARRIER LOSS'
000002          KL.ACTIVE =     ^O2             ;INTERRUPT CODE FOR WHEN KLINIK BECOMES ACTIVE
000373          STMSK   =       ^O373           ;MASK FOR THE STATE LIGHT
000004          STBIT   =       ^O4             ;STATE BIT
000333          .IN     =       ^O333           ;THE 'IN' INSTRUCTION
000323          .OUT    =       ^O323           ;THE 'OUT' INSTRUCTION
000311          .RET    =       ^O311           ;THE 'RET' INSTRUCTION
```

```
                        .SBTTL****ASCII CHARACTER VALUE ASSIGNMENTS
000000          Q.OUT   =       00              ;NULL CHAR..STANDS FOR 'QUICK.OUT'
000001          SYNC    =       01              ;SYNC CHAR FOR APT MESSAGES
000015          CRCHR   =       ^O15            ;CARRIAGE RETURN
000012          LFCHR   =       ^O12            ;LINE FEED
000054          COMMA   =       ^O54            ;ASCII FOR COMMA
000034          CNBCK   =       ^O34            ;CONTROL-BACKSLASH
000007          BELL    =       ^O7             ;A BELL FOR WHEN HOST SYSTEMS GIVE ME GRIEF
000052          STAR    =       ^O52            ;A STAR *
000042          QUO     =       ^O42            ;DOUBLE QUOTES "
000043          LBSIGN  =       ^O43            ;POUND SIGN #
000051          RPAREN  =       ^O51            ;RIGHT PAREN )
000033          ALT     =       ^O33            ;ALT MODE
000032          CNTLZ   =       ^O32            ;CONTROL-Z
000031          CNTLY   =       ^O31            ;CONTROL-Y ENDS MODE 4
000025          CNTLU   =       ^O25            ;CONTROL-U
000023          CNTLS   =       ^O23            ;CONTROL-S
000021          CNTLQ   =       ^O21            ;CONTROL-Q
000017          CNTLO   =       ^O17            ;CONTROL-O
000003          CNTLC   =       ^O3             ;CONTROL-C
000177          RBOUT   =       ^O177           ;RUB-OUT
000103          CCHR    =       ^O103           ;'C'
000104          DCHR    =       ^O104           ;'D'
000011          TAB     =       ^O11            ;HORIZONTAL TAB
000040          SPACE   =       ^O40            ;SPACE
000057          SLASH   =       ^O57            ;SLASH '/'
000134          BSLASH  =       ^O134           ;BACKSLASH '\'
000044          DOLLAH  =       ^O44            ;DOLLAR SIGN '$'
000045          PERCNT  =       ^O45            ;'%'
000077          QUES    =       ^O77            ;'?'
000076          ARROW   =       ^O76            ;'^'
000125          UCHR    =       ^O125           ;'U'
000117          OCHR    =       ^O117           ;'O'
000136          UPARR   =       ^O136           ;'^'
000001          ONE     =       ^O01            ;'1'
000002          TWO     =       ^O02            ;'2'
000003          THREE   =       ^O03            ;'3'
000004          FOUR    =       ^O04            ;'4'
000005          FIVE    =       ^O05            ;'5'
000006          SIX     =       ^O06            ;'6'
000007          SEVEN   =       ^O07            ;'7'
000070          EIGHT   =       ^O70            ;'8' THE ASCII NUMBER
000071          NINE    =       ^O71            ;'9' THE ASCII NUMBER
000012          TEN     =       ^O12            ;'10'
```

```
                        ;NUMERICAL ASSIGNMENTS TO GENERATE THE OFFSETS USED IN COMPUTING
                        ;DEVICE ADDRESSES FROM ANY RH BASE ADDRESS
                                .RADIX 8
000000                  P.00    =       0
000002                  P.02    =       2
000004                  P.04    =       4
000006                  P.06    =       6
000010                  P.10    =       10
000012                  P.12    =       12
000014                  P.14    =       14
000016                  P.16    =       16
000020                  P.20    =       20
000022                  P.22    =       22
000024                  P.24    =       24
000026                  P.26    =       26
000030                  P.30    =       30
000032                  P.32    =       32
000034                  P.34    =       34
000036                  P.36    =       36
000040                  P.40    =       40
000042                  P.42    =       42
000044                  P.44    =       44
000046                  P.46    =       46
000776                  D776    =       776
000772                  D772    =       772
                                .RADIX 10
                        ;DRIVE CONTROLLER REGISTER IS 776440.,FOLLOWING COMMANDS APPLY
                                        ;       7=REWIND
                                        ;       11=DRIVE CLEAR
                                        ;       25=ERASE
                                        ;       27=WRITE TAPE MARK
000031                   SKP.TAPE=^D31  ;       31=SPACE FORWARD(SKIP A FILE)
                                        ;       33=SPACE REVERSE(SKIP A FILE, MOVING TAPE IN REVERSE)
                                        ;       51=WRITE CHECK FORWARD
                                        ;       57=WRITE CHECK REVERSE
                                        ;       61=WRITE FORWARD
000071                   READ.TAPE=^D71 ;       71=READ FORWARD(GO!)
                                        ;       77=READ REVERSE(GO!)
                        ;DEFAULT VALUES FOR CHECKSUM COUNTS, SO THAT ASSEMBLIES WITH 3 PROMS WILL WORK
000000                          CHKSM0  =       0
000000                          CHKSM1  =       0
000000                          CHKSM2  =       0
000000                          CHKSM3  =       0
```

```
                        .SBTTL****MACROS FOR OFTEN USED CODE, OR CODE SUBJECT TO REPEATED CHANGES

                        ;PCHAR---
                        ; PRINTS A SINGLE CHARACTER, WHICH IT FINDS IN THE TRAILING BYTE
                        ; CLOBBERS ACCUM
                        .MACRO  PCHAR           XCHR
                                RST     1       ;GO PRINT CHAR IN TRAILING BYTE
                                .BYTE   XCHR    ;CHAR TO PRINT

                                .ENDM

                        ;KCHAR---
                        ; PRINT A SINGLE CHARACTER ON THE KLINIK LINE ONLY. CHAR TO BE PRINTED PASS
                        ; AS A TRAILING ARG
                        .MACRO  KCHAR   XCHR
                                CALL    KCHR    ;GO PRINT THE CHARACTER
                                .BYTE   XCHR

                                .ENDM

                        ;PLINE---
                        ; PRINTS A LINE OF CHARACTERS, POINTER PASSED AS TRAILING ARG
                        ; AND END OF LINE SIGNALED BY A '0' BYTE
                        .MACRO  PLINE    XMS
                                RST     3       ;PRINT LINE OF CHARS
                                .ADDR   XMS     ;BUFF TO PRINT

                                .ENDM
                        ;KLINE---
                        ; PRINTS A LINE OF CHARACTERS, POINTER PASSED AS TRAILING ARG
                        ; AND END OF LINE SIGNALED BY A '0' BYTE
                        .MACRO  KLINE    XMS
                                CALL    KLINE   ;PRINT LINE OF CHARS
                                .ADDR   XMS     ;BUFF TO PRINT

                                .ENDM
                        ;INTOFF---
                        ; EXECUTES THE OLD 'INTERNAL MODE OFF' SUBROUTINE.., BUT BY USING THE
                        ; RESTART INSTRUCTION CALL, WE SAVE 30.+ BYTES OVER 'CALL INTOFF'
                        .MACRO  INTOFF
                                RST     6       ;GO EXIT FROM INTERNAL MODE

                                .ENDM
                        ;INTON---
                        ; EXECUTES THE OLD 'INTERNAL MODE ON' SUBROUTINE.., BUT BY USING THE
                        ; RESTART INSTRUCTION CALL, WE SAVE 30.+ BYTES OVER 'CALL INTON'
                        .MACRO  INTON
                                RST     2       ;GO SET INTERNAL MODE

                                .ENDM
```

```
                        ;CLINE---
                        ; PRINTS A LINE OF CHARACTERS, POINTER PASSED IN (H,L),
                        ; AND END OF LINE SIGNALED BY A '0' BYTE
                        .MACRO  CLINE    XMS
                               LXI     H,XMS   ;PASS POINTER TO THE CHARACTERS
                               CALL    CLINE   ;PRINT LINE OF CHARS

                               .ENDM

                        ;PCRLF---
                        ; PRINTS CARRIAGE RETURN-LINE FEED
                        ; LEAVES ALL REGISTERS IN TACT
                        .MACRO  PCRLF
                               CRLF            ;GO PRINT CARRIAGE RETURN LINE FEED

                               .ENDM
                        ;SSCLK---
                        ; ISSUES A SINGLE KS10 CLOCK,,CLOBBERS THE ACCUMULATOR
                        .MACRO  SSCLK
                               MVI     A,02    ;BIT TO SET 'SINGLE CLK' TO KS10
                               OUT     CLKCTL  ;ISSUE THE SINGLE CLOCK

                               .ENDM
                        ;PTAB---
                        ; PRINTS A TAB
                        ; CLOBBERS ACCUM
                        .MACRO  PTAB
                               PCHAR   TAB     ;GO PRINT A TAB

                               .ENDM
                        ;PSPACE---
                        ; PRINTS A ' ' SPACE
                        ; CLOBBERS ACCUM
                        .MACRO  PSPACE
                               PCHAR   SPACE   ;GO PRINT A SPACE

                               .ENDM
```

```
                        ;PSLSH---
                        ; PRINTS A "/" SLASH
                        ; CLOBBERS ACCUM
                        .MACRO  PSLASH
                                PCHAR   SLASH   ;GO PRINT A SLASH

                                .ENDM
                        ;ENDCMD---MACRO TO DO THE RIGHT STUFF
                        ; AT THE END OF A COMMAND..COMMON CODE FOR FINISHING ALL INSTRS
                        .MACRO  ENDCMD
                                RET             ;RETURN TO CALLER

                                .ENDM
                        ;BUSRESP--- MACRO TO READ I/O REG 301Q
                        ; IN ORDER TO CHECK ANY OF THE BITS WHICH
                        ; ARE READABLE IN THAT REGISTER.. THE BIT
                        ; OR BITS TO BE CHECKED IS PASSED AS A
                        ; TRAILING ARGUEMENT 'DB' TO A RST INSTR.
                        ;   ;THE EXECUTION OF THE RESTART ACCOUNTS
                        ; FOR THE 2 'NOP' TIME REQUIRED FOR THE
                        ; BUS ARBITRATOR TO GRANT THE BUS. THIS
                        ; MACRO RETURNS MAINLINE WITH THE Z-BIT SET
                        ; IF THE REG 301 DOES !NOT! MATCH THE TRAILING
                        ; ARG THAT WAS PASSED AND WITH THE Z-BIT CLEAR
                        ; IF THERE IS A 'TRUE' IN ANY OF THE BIT POSITIONS
                        ; PASSED...
                        .MACRO  BUSRESP  XXX
                                CALL    BUSRESP ;DO A CALL TO EXECUTE THIS CODE
                                .BYTE   XXX     ;BITS TO BE CHECKED

                                .ENDM
                        .MACRO  D       ARG,ARG1,NUL,ARG2,ARG3
                                .NLIST

                                .RADIX  8

                                .BYTE   ARG3 & 377
                                Q.1 =   <<ARG3/400> & 1> ! <<ARG2*2> & 376>
                                Q.2 =   <<ARG2/200> & 3> ! <<ARG1*4> & 374>
                                Q.3 =   <<ARG1/100> & 7> ! <<ARG*10> & 370>
                                Q.4 =   <ARG/40> & 17
                                .BYTE   Q.1
                                .BYTE   Q.2
                                .BYTE   Q.3
                                .BYTE   Q.4

                                .RADIX 10
                                .LIST

                        .ENDM
```

```
                        .MACRO  LI.     ARG2,ARG3
                                .RADIX  8


                        ARG     =       0
                        ARG1    =       0
                        NUL     =       0


                                Q.1 =   <<ARG3/400> & 1> ! <<ARG2*2> & 376>
                                Q.2 =   <<ARG2/200> & 3> ! <<ARG1*4> & 374> ! <2*4>
                                .BYTE   Q.1
                                .BYTE   ARG3 & 377
                                .BYTE   Q.2
                                .RADIX  10

                        .ENDM


                        .MACRO  EI.     ARG2,ARG3
                                .RADIX 8


                        ARG     =       0
                        ARG1    =       0
                        NUL     =       0

                                Q.1 =   <<ARG3/400> & 1> ! <<ARG2*2> & 376>
                                Q.2 =   <<ARG2/200> & 3> ! <<ARG1*4> & 374> ! <4*4>
                                .BYTE   Q.1
                                .BYTE   ARG3 & 377
                                .BYTE   Q.2
                                .RADIX 10

                        .ENDM
```

```
                        .MACRO  DI,     ARG2,ARG3
                                .RADIX 8

                        ARG     =       0
                        ARG1    =       0
                        NUL     =       0

                                .BYTE   ARG3 & 377
                                Q.1 =   <<ARG3/400> & 1> ! <<ARG2*2> & 376>
                                Q.2 =   <<ARG2/200> & 3> ! <<ARG1*4> & 374>
                                .BYTE   Q.1
                                .BYTE   Q.2

                                .RADIX  10
                        .ENDM


                        .MACRO  DI,INDIRECT ARG3
                                .RADIX  8
                                .ADDR   ARG3
                                .BYTE   ^O200

                                .RADIX 10
                        .ENDM

                        .MACRO  TWAIT ARG3
                                .RADIX  8
                                .ADDR   ARG3
                                .BYTE   <14*4>

                                .RADIX 10
                        .ENDM

                        .MACRO  WAIT ARG3
                                .RADIX  8
                                .ADDR   ARG3
                                .BYTE   <6*4>

                                .RADIX 10
                        .ENDM
```

```
                        .MACRO  ERRTST ARG3
                                .RADIX  8
                                .ADDR   ARG3
                                .BYTE   <10*4>

                                .RADIX 10
                        .ENDM
                        .MACRO  UBA.    ARG2,ARG3
                                .RADIX  8


                        ARG     =       0
                        ARG1    =       0
                        NUL     =       0


                                .BYTE   ARG3 & 377
                                Q.1 =   <<ARG3/400> & 1> ! <<ARG2*2> & 376>
                                Q.2 =   <<ARG2/200> & 3> ! <<ARG1*4> & 374> ! <16*4>
                                .BYTE   Q.1
                                .BYTE   Q.2
                                .RADIX  10

                        .ENDM

                        .MACRO  ENDLST
                                .RADIX 8

                                .BYTE   0
                                .BYTE   0
                                .BYTE   <12*4>
                                .RADIX 10

                        .ENDM
```

```
                        .MACRO  MOV5B
                                RST     4
                                .BYTE   0
                        .ENDM

                        .MACRO  CRLF
                                RST     4
                                .BYTE   2
                        .ENDM

                        .MACRO  ARG16
                                RST     4
                                .BYTE   4
                        .ENDM

                        .MACRO  RUN..
                                RST     4
                                .BYTE   6
                        .ENDM

                        .MACRO  ARG36
                                RST     4
                                .BYTE   8
                        .ENDM

                        .MACRO  CLRRM   FOO
                                RST     4
                                .BYTE   10.
                                .ADDR   FOO+5
                        .ENDM
```

```
                        .MACRO  EXAM FOO
                                .RADIX 8
                                CALL    EXAMSH  ;AND DO EXAM ASSUMING SHORT ADDRESS
                                .ADDR   FOO     ;ADDR TO BE ZAPPED PASSED AS TRAILING ARG
                                .RADIX 10
                        .ENDM

                        .MACRO  DEPOS FOO
                                .RADIX 8
                                ANA     A       ;CLR "C-BIT" FOR USE BY COMMON CODE
                                CALL    DEPSHT  ;AND DO THE DEPOSIT ASSUMING SHORT ADDR
                                .ADDR   FOO     ;ADDR TO BE ZAPPED PASSED AS TRAILING ARG
                                .RADIX 10
                        .ENDM

                        .MACRO  RD.REG  FOO
                                IN      FOO     ;READ AN I/O REG
                                CMA             ;AND ALWAYS COMPLEMENT
                        .ENDM

                        .MACRO  SETZM   FOO
                                CLRRM   DMDAT
                                DEPOS   FOO
                        .ENDM

                        .MACRO  TSTRDY  FOO
                                .RADIX 8
                                IN      FOO     ;READ DESIRED UART STATUS
                                ANI     01      ;CHECK IF SET; ZBIT=0 IF READY; ZBIT=1 IF NOT YET READY
                                .RADIX 10
                        .ENDM

                        .MACRO  LONG.DELAY      ARG
                                LXI     H,200. * ARG ;SET UP THE TIMING COUNT IN H,L REG
                                CALL    LTLOOP  ;AND GO DELAY ABOUT 1 SECOND FOR EACH (300 COUNT)
                        .ENDM
```

```
                        ;MACRO TO SAVE SPACE ON OPERATIONS THAT WANT TO CLEAR
                        ;A LOCATION IN THE RAM.. ELIMINATES EVERY XRA/STA PAIR, AND
                        ;INSERTS INSTEAD AN RST/BYTE PAIR.. AT BEST, AN XRA AND AN STA
                        ;ARE ELIMINATED. AT WORST, A 3 BYTE STA IS REPLACED WITH
                        ;A 2 BYTE RST/BYTE.
                        .MACRO  CLRB    XX
                                .NLIST
                                OFFSET= XX - ^O20000
                        .IF     LE, ^O400-OFFSET
                        .IFF
                                RST     5
                                .BYTE   OFFSET
                        .IFT
                                .ERROR OFFSET,;BYTE OUT OF RANGE ON CLRB MACRO
                        .ENDC
                        .LIST
                        .ENDM

        120052                  CHKSM0=-^O57726
        140617                  CHKSM1=-^O37161
        160157                  CHKSM2=-^O17621
        003216                  CHKSM3=-^O174562
```

```
                        .SBTTL**** KS10 CONSOLE HARDCORE ****
                        .SBTTL  **'RST' INSTRUCTION CODE**

                        ;AT POWER UP, ALWAYS BEGIN AT 0000

     000000                     .=      0000
000000  000                     NOP             ;NO-OPS FOR 2 INSTR CYCLES
000001  000                     NOP             ; LETS 8080 SETTLE DOWN

000002  363                     DI              ;GUARANTEE INTERRUPTS DISABLED
000003  303  100  000           JMP     PWRUP   ;BEGIN WITH INITIALIZATION CODE

                        ;*** RESERVE 'RESTART' MEMORY BLOCKS FOR INTERRUPTS
                        ;*** AND SPECIAL PURPOSE 'RST' INSTRUCTIONS
                        ;BEGIN CODE FOR 'PCHAR' WITH A 'RST 1'
     000010                     .=      ^O10

000010  343                     XTHL            ;GET POINTER TO TRAILING ARG
000011  176                     MOV     A,M     ;CHAR INTO ACCUM
000012  043                     INX     H       ;UPDATE POINTER TO RETURN ADDRESS
000013  343                     XTHL            ;RESTORE RETURN ADDRESS TO STACK
000014  303  010  004           JMP     PCHR    ;AND GO....

                        ;*** RESERVE 'RESTART' MEMORY BLOCKS FOR INTERRUPTS
                        ;*** AND SPECIAL PURPOSE 'RST' INSTRUCTIONS
                        ;SUBROUTINE TO SET INTERNAL STATUS(I.E. DONT TYPE ON TTY)
     000020                     .=      ^O20

000020  345             INTON:  PUSH    H       ;SAVE THE LIL BUGGER
000021  041  120  040           LXI     H,NOPNT ;GET VALUE OF FLAG
000024  064                     INR     M       ;INCREMENT BY ONE
000025  341                     POP     H       ;RESTORE THE REG WE SAVED
000026  311                     RET             ;AND RETURN
```

```
                        ;*** RESERVE 'RESTART' MEMORY BLOCKS FOR INTERRUPTS
                        ;*** AND SPECIAL PURPOSE 'RST' INSTRUCTIONS
                        ;BEGIN CODE FOR 'PLINE'
      000030                    .=      ^O30

000030  343                     XTHL            ;GET PNTR TO TRAILING ARG
000031  136                     MOV     E,M     ;GET LO ORDER PIECE
000032  043                     INX     H       ;UPDATE PNTR
000033  126                     MOV     D,M     ;GET HI ORDER PIECE
000034  043                     INX     H       ;UPDATE PNTR
000035  303  156  004           JMP     PLNE    ;AND GO TO ACTUAL ROUTINE

                        ;*** RESERVE 'RESTART' MEMORY BLOCKS FOR INTERRUPTS
                        ;*** AND SPECIAL PURPOSE 'RST' INSTRUCTIONS
                        ;BEGIN CODE FOR 'TRAP' OR 'UUO' TYPE SUBROUTINE CALLS
      000040                    .=      ^O40

000040  343                     XTHL            ;SAVE 'H,L' WHILE GETTING PC
000041  176                     MOV     A,M     ;GET INDEX INTO SUBROUTINE DISPATCH LIST
000042  043                     INX     H       ;UPDATE 'PC' TO POINT AT RETURN
000043  343                     XTHL            ;PUT BACK RETURN AND RESTORE 'H,L'
000044  345                     PUSH    H       ;NOW SAVE 'H,L' WHILE WE SET UP DISPATCH
000045  303  075  032           JMP     RTNDIS  ;GO CALCULATE SUBROUTINE TO DISPATCH TO..

                        ;*** RESERVE 'RESTART' MEMORY BLOCKS FOR INTERRUPTS
                        ;*** AND SPECIAL PURPOSE 'RST' INSTRUCTIONS
                        ;THIS CODE CLEARS A SINGLE BYTE OF RAM SPACE, IN THE RANGE
                        ;FROM 20000-20377
      000050                    .=      ^O50

000050  343                     XTHL            ;SAVE 'H,L' WHILE GETTING PC
000051  176                     MOV     A,M     ;GET INDEX INTO SUBROUTINE DISPATCH LIST
000052  043                     INX     H       ;UPDATE 'PC' TO POINT AT RETURN
000053  343                     XTHL            ;PUT BACK RETURN AND RESTORE 'H,L'
000054  345                     PUSH    H       ;NOW SAVE 'H,L' WHILE WE SET UP DISPATCH
000055  303  133  032           JMP     CLRBYT  ;GO CALCULATE SUBROUTINE TO DISPATCH TO..

                        ;*** RESERVE 'RESTART' MEMORY BLOCKS FOR INTERRUPTS
                        ;*** AND SPECIAL PURPOSE 'RST' INSTRUCTIONS
                        ;THIS CODE EXECUTES THE 'INTERNAL MODE OFF' FUNCTION, TURNING OFF
                        ;THE PRINTING OF 8080 FUNCTIONS EXECUTED INTERNALLY. 'INTOFF' IS
                        ;CALLED AT LEAST 15 TIMES, SO THIS RESTART CODE SAVES AT LEAST
                        ;30 BYTES OVER 'CALLS INTOFF'
      000060                    .=      ^O60

000060  345                     PUSH    H       ;SAVE THE LIL BUGGER
000061  041  120  040           LXI     H,NOPNT ;CLEAR ACCUM
000064  065                     DCR     M       ;DECREMENT PRINT FLAG
000065  341                     POP     H       ;RESTORE THE REG WE SAVED
000066  311                     RET             ;AND RETURN

                        ;*** RESERVE 'RESTART' MEMORY BLOCKS FOR INTERRUPTS
                        ;*** AND SPECIAL PURPOSE 'RST' INSTRUCTIONS
                        ;BEGIN CODE FOR HANDLING INTERRUPTS
```

```
  603       000070                    .=      ^O70
  604
  605  000070  363                    DI              ;GUARANTEE INTERRUPTS DISABLED
  606  000071  365                    PUSH    PSW     ;SAVE
  607  000072  305                    PUSH    B       ; EVERYTHING
  608  000073  325                    PUSH    D       ;  ONTO THE STACK
  609  000074  345                    PUSH    H
  610
  611  000075  303  262  005          JMP     INTRP   ;AND GO TO PROCESS THE INTERRUPT...
```

```
  613                           ;MAINLINE CODE BEGINS HERE
  614       000100                      .=      ^O100
  615
  616                           .SBTTL  **POWER UP START LOCATON**
  617                           ;BEGIN BY SETTING UP 8080 STACK POINTER
  618  000100  061  000  044   PWRUP: LXI     SP,RAMST+^O2000 ;LOAD STACK PNTR WITH TOP RAM LOC
  619
  620                           ;*********************************************
  621                           ;WILL BE TAKEN OUT WHEN WE PUT IN THE POWER UP SELF TEST
  622
  623                           ;MUST SET DISPATCH LIST AND CLEAR RAM
  624  000103  041  000  040           LXI     H,RAMST ;SET 'H,L' REGISTER TO START  RAM ADDR
  625  000106  021  000  004           LXI     D,^O2000 ;COUNTER TO FILL REST OF RAM W 0'S
  626
  627  000111  066  000        CLRLP: MVI     M,00    ;CLEAR A RAM LOC
  628  000113  043                    INX     H       ;NEXT RAM LOCATION
  629  000114  033                    DCX     D       ;DOWN COUNTER
  630  000115  173                    MOV     A,E     ;LO ORDER PIECE TO ACCUM
  631  000116  262                    ORA     D       ;THROW IN THE HI ORDER PIECE
  632  000117  302  111  000           JNZ     CLRLP   ;CONTINUE TILL DONE
  633                           ;*************************************************
```

```
                        ;BEGIN PROCEEDURE FOR INITIALIZING KS10
                        ;MUST INDIVIDUALLY CLEAR KS10 FLOPS 'RUN','EXECUTE',& 'CONTINUE'
                        ; BECAUSE THEY ARE NOT RESET BY KS10 BUS RESET
000122  257                     XRA     A       ;SET ACCUM=0
                        ;SET 0'S TO 'RUN,EXECUTE,CONT'
000123  323  212                OUT     CPUCTL  ;***** I/O WRT 212/0 *****

                        ;ISSUE KS10 BUS RESET
000125  315  233  005          CALL    MRINT   ;DOES RESET AND SETS DEFAULT PARITY & TRAPS
```

```
                        ,SBTTL  **INITIALIZE UARTS**
                        ;RAM BUFFERS HAVE BEEN INITIALIZED.. NOW MUST READ FRONT
                        ; PANEL SWITCHES & INITIALIZE 'UART'
000130  333  300               IN      TTYSW   ;***** I/O READ 300Q *****
000132  057                   CMA             ;FIX INVERSION

000133  001  200  004          LXI     B,^O2200 ;THIS SETS B=^O4 AND C=^O200
000136  147                   MOV     H,A     ;H WILL HOLD CTY STOP BIT FOR US.@BIT POS 7
000137  037                   RAR             ;BIT 4 MOVES TO BIT 3
000140  037                   RAR             ;BIT MOVES TO BIT 2
000141  137                   MOV     E,A     ;E WILL HOLD KLINIK LENGTH BIT, @BIT POS 2
000142  037                   RAR             ;BIT 6 IS AT 4 FROM OTHER SHIFTS, NOW TO BIT 3
000143  037                   RAR             ;AND TO BIT 2
000144  157                   MOV     L,A     ;L WILL HOLD CTY LENGTH BIT, @BIT POS 2
000145  174                   MOV     A,H     ;ORIGINAL BACK TO ACCUM
000146  027                   RAL             ;BIT 5 TO BIT 6
000147  027                   RAL             ;AND TO BIT 7
000150  127                   MOV     D,A     ;D WILL HOLD KLINIK STOP BIT, @BIT POS 7

                        ;NOW BEGIN MASKING OPERATIONS TO ISOLATE THE DESIRED UART BIT SETTINGS
000151  171                   MOV     A,C     ;MASK OF OCTAL 200 INTO ACCUM
000152  244                   ANA     H       ;ACCUM HAS ONLY CTY STOP BIT
000153  147                   MOV     H,A     ;COPY BACK TO H
000154  171                   MOV     A,C     ;MASK TO ACCUM
000155  242                   ANA     D       ;ACCUM HAS ONLY A STOP BIT
000156  127                   MOV     D,A     ;COPY BACK TO D

                        ;NOW MASK FOR THE LENGTH BITS
000157  170                   MOV     A,B     ;MASK OF OCTAL 4 INTO ACCUM
000160  245                   ANA     L       ;ACCUM NOW HAS CTY LENGTH BIT ONLY
000161  264                   ORA     H       ;THROW IN THE CTY STOP BIT
000162  366  112               ORI     ^B01001010 ;ADD IN THE CONSTANT BITS
000164  323  201               OUT     CTYCTL  ;SET THE UART MODE.........

                        ;NOW SET MODE FOR THE KLINIK STUFF
000166  170                   MOV     A,B     ;MASK OF OCTAL 4 INTO ACCUM
000167  243                   ANA     E       ;ACCUM NOW HAS KLINIK LENGTH BIT ONLY
000170  262                   ORA     D       ;THROW IN THE KLINIK STOP BIT
000171  366  112               ORI     ^B01001010 ;ADD IN THE CONSTANT BITS
000173  323  203               OUT     REMCTL  ;SET THE KLINIK UART MODE
```

```
                        ;NOW ENABLE THE UART TO RECEIVE AND TRANSMIT
000175  076  025              MVI     A,^B00010101 ;BITS FOR UART CONTROL
                        ;SET UART TO RECEIVE AND TRANSMIT
000177  323  201              OUT     CTYCTL  ;***** I/O WRT 200/025 *****

000201  076  020              MVI     A,^O20  ;ONE BIT TO SAY RESET THE UART
000203  323  203              OUT     REMCTL  ;RESET THE KLINIK UART BUT DO NOT ENABLE IT!!

000205  333  200              IN      CTYDAT  ;***** I/O RD 201***** READ 1 DUMMY CHAR OUT OF UART
000207  333  202              IN      REMDAT  ;***** I/O RD 203***** READ 1 DUMMY CHAR OUT OF UART

000211  315  160  010          CALL    BFRST   ;INIT TTY INPUT BUFFER
```

```
                        .SBTTL  8080 PROM CHECKSUMMER
                        ;CODE TO COMPUTE A CHECKSUM FOR EACH OF THE 2K 8080 PROM PIECES
                        ;FIRST COMPUTE THE CHECKSUMS FOR EACH PROM, THEN DO SIMPLE TEST ON THE 8080
                        ;RAM..BEGIN WITH THE PROMS
000214  041  000  000          LXI     H,00    ;START AT PROM ADDRESS 0
000217  257              LCHKR: XRA     A       ;CLEAR ACCUM
000220  117                     MOV     C,A     ;CLEAR B,C PAIR
000221  107                     MOV     B,A     ;'B,C' TO BE USED AS ADDEND
000222  137                     MOV     E,A     ;CLEAR D,E PAIR
000223  127                     MOV     D,A     ;'D,E' TO HOLD CURRENT COUNT OF CHECKSUM

000224  116              A256:  MOV     C,M     ;GET A BYTE
000225  043                     INX     H       ;UPDATE MEM POINTER
000226  353                     XCHG            ;PUT CURRENT CHECKSUM SUB TOTAL INTO H,L
000227  011                     DAD     B       ;THROW IN AN ADDEND
000230  353                     XCHG            ;RETURN H,L TO RIGHTFUL PLACE

                        ;NOW QUICK CHECK FOR THE END OF PROM
000231  175                     MOV     A,L     ;GET LO ORDER OF CURRENT PROM ADDRESS
000232  247                     ANA     A       ;SET CONDITION CODES
000233  302  224  000          JNZ     A256    ;IF .EQ. 0,WE'VE DONE 256 LOCS, IF .NE. 0, DO MORE

                        ;FALL TO HERE WHEN DONE A CHUNK OF 256..SEE IF ITS ON A PROM BOUNDARY NOW
000236  174                     MOV     A,H     ;GET HI ORDER PIECE OF ADDRESS
000237  346  007                ANI     ^O7     ;IF THIS .EQ. 0, THEN WE ARE AT A BOUNDARY
000241  302  224  000          JNZ     A256    ;IF .NE 0, THEN KEEP TRYING
```

```
                                ;FALL THRU HERE WHEN WE'VE COMPLETED A PROMS WORTH
000244  174                     MOV     A,H     ;GET HI ORDER, WHICH CORRESPONDS TO "WHICH PROM"
000245  017                     RRC             ;JUSTIFY AT EDGE OF THE ACCUM
000246  017                     RRC
000247  017                     RRC
000250  075                     DCR     A       ;DECREMENT MAKES "0-3" INSTEAD OF "1-4"
000251  207                     ADD     A       ;DOUBLE THIS VALUE TO MAKE IT ON 16 BYTE BOUNDARIES

000252  365                     PUSH    PSW     ;SAVE THIS VALUE FOR A BIT
000253  345                     PUSH    H       ;AND SAVE OUR CURRENT POINTER
000254  302  315  000           JNZ     DEV     ;IF DOING PROM ZERO, ELIMINATE THE ACTUAL CHECKSUM COUNT

                                ;FELL THRU TO HERE TO DO ACTUAL CHECKSUM ELIMINATION
                                ;EXECUTED WHEN CHECKSUMMING PROM 0. IT REMOVES THE ACTUAL CHECKSUM VALUES
                                ;FROM THE COMPUTED CHECKSUM, AS WE CANNOT SOLVE THE CHECKSUM FEEDBACK
                                ;LOOP PROBLEM
000257  365                     PUSH    PSW     ;SAVE ACCUM
000260  041  175  005           LXI     H,CHECKS ;H,L GETS A POINTER TO THE LIST OF CHECKSUMS
000263  076  010                 MVI     A,8.    ;START WITH A COUNT OF 8, FOR THE EIGHT BYTES WE MUST SUB
000265  062  000  040  DEVLP:    STA     T8ODT   ;SAVE COUNT IN RAM, SO WE CAN USE ACCUM
000270  006  377                 MVI     B,^O377 ;IN B,C PAIR, ENSURE UPPER HALF .EQ. -1
000272  176                     MOV     A,M     ;GET CHECKSUM BYTE TO ACCUM
000273  057                     CMA             ;NEGATE IT
000274  117                     MOV     C,A     ;THROW IT INTO A 16 BIT ENTITY
000275  003                     INX     B       ;AND MAKE B,C PAIR 2'S COMPLIMENT

000276  353                     XCHG            ;GET CURRENT COMPUTED CHECKSUM TO H,L
000277  011                     DAD     B       ;"SUBTRACT" THE BYTE WE'VE ASSEMBLED.(2'S COMP ADDITION)
000300  353                     XCHG            ;PUT H,L/D,E PAIR RIGHT
000301  043                     INX     H       ;AND UPDATE THE POINTER INTO THE LIST
000302  072  000  040           LDA     T8ODT   ;GET OUR CURRENT COUNT FROM THE RAM
000305  075                     DCR     A       ;DECREMENT
000306  302  265  000           JNZ     DEVLP   ;CONTINUE IN THE LOOP

000311  001  000  000           LXI     B,00    ;NOW GUARANTEE B,C PAIR IS ALL GONE
                                ;FALL THRU WHEN FINISHED
000314  361                     POP     PSW     ;RESTORE ACCUM
```

```
  764  000315  117            DEV:   MOV    C,A     ;GET CURRENT PROM NUMBER INTO C
  765  000316  041  175  005          LXI    H,CHECKS ;H,L POINTS TO THE TABLE OF CHECKSUM
  766  000321  011                   DAD    B       ;ADD INDEX, AND NOW (M) PNTS TO EXPECTED CHECKSUM
  767  000322  116                   MOV    C,M     ;COPY EXPECTED CHECKSUM INTO C
  768  000323  043                   INX    H
  769  000324  106                   MOV    B,M     ;AND B, MAKES 'B,C' PAIR
  770  000325  043                   INX    H       ;AND KEEP COUNT UP
  771
  772                         ;NOW D,E HAS CALCULATED CHECKSUM AND B,C HAS EXPECTED CHECKSUM
  773  000326  353                   XCHG           ;H,L NOW HAS CALCULATED CHECKSUM
  774  000327  011                   DAD    B       ;IF CHECKSUM OK, RESULT OF THIS SHOULD .EQ. 0
  775  000330  175                   MOV    A,L     ;GET LO PIECE
  776  000331  264                   ORA    H       ;THROW IN HI PIECE..CONDITION CODE FLAGS NOW SET
  777
  778  000332  353                   XCHG           ;D,E NOW HAS THE RESULTS OF THE ADDITION
  779  000333  341                   POP    H       ;BEFORE JUMPING ON CONDITION CODES, FIRST FIX REGISTER
  780  000334  302  356  000          JNZ    CHKERR  ;IF H,L WAS .NE. 0, THEN WE HAD PROM CHECKSUM ERROR...
  781
  782                         ;FALL THRU IF CHECKSUM WAS OK
  783  000337  361                   POP    PSW     ;RESTORE INDEX INTO "WHICH PROM" WE ARE IN
  784  000340  376  006              CPI    6       ;SEE IF DONE ALL
  785  000342  302  217  000          JNZ    LCHKR   ;JUMP BACK TO BEGINNING IF NOT DONE ALL
  786
  787                         ;FALL THRU IF DONE ALL..PROMS CHECKSUM OK
  788  000345  303  373  000          JMP    XXX230  ;SO AVOID THE ERROR PRINTOUT CODE
```

```
  790                           ;YOU JUMPED TO HERE IF YOU ENCOUNTERED A CHECKSUM ERROR.
  791  000350  077  103  110     ROMMSG: .ASCIZ /?CHK /  ;IF PROM CHECKSUM FAILS, USE THIS MESSAGE
  792  000356                    CHKERR: PLINE   ROMMSG  ;PRINT '?CHK '
(1)    000356  337                       RST     3       ;PRINT LINE OF CHARS
(1)    000357  350  000                   .ADDR   ROMMSG  ;BUFF TO PRINT
  793  000361  361                       POP     PSW     ;RETRIEVE PROM NUMBER
  794  000362  017                       RRC             ;DIVIDE DOWN, SINCE WE DOUBLED IT BEFORE
  795  000363  074                       INR     A       ;MAKE PROM TYPE-OUT BE 1-4
  796  000364  366  060                   ORI     ^O60    ;MAKE IT ASCII
  797  000366  315  010  004             CALL    PCHR    ;GO PRINT IT
  798  000371                            PCRLF           ;<CR-LF>
(2)    000371  347                       RST     4
(2)    000372  002                       .BYTE   2
```

```
800                             ;FINAL STEP IS TO ENABLE PARITY DETECTION IN THE KS10
801                             ; DEFAULT IS: PARITY DETECTION ON,,DATA PATH PARITY DETECION ENABLED
802  000373  076  174            XXX230: MVI     A,DEFLTE ;INIT ENABLES TO BE ON
803  000375  323  100                    OUT     SMPAR   ;***** I/O WRT 100/174 *****
804
805                             ;LITTLE ROUTINE TO LOAD ALL DEFAULT CONSTANTS INTO THE 8080 RAM
806                             ;THIS ROUTINE SAVE ABOUT 40. BYTES OF PROM, OVER USING LXI'S,SHLD'S
807                             ;MVI'S & STA'S
808  000377  041  342  040            LXI     H,KATIM1 ;LIST OF DESTINATION LOCATIONS BEGINS HERE
809  000402  021  021  001            LXI     D,PRMLST ;LIST OF DATUMS FOR THE RAM INIT
810  000405  032                    RAMMER: LDAX    D       ;GET BYTE FROM THE DATA LIST
811  000406  376  252                    CPI     ^O252   ;TEST FOR END OF LIST
812  000410  312  062  001            JZ      XXX235  ;CONTINUE PROM INIT
813
814                             ;ELSE MORE RAM TO INITIALIZE
815  000413  167                    MOV     M,A     ;PUT GOOD STUFF IN RAM
816  000414  043                    INX     H       ;UPDATE POINTERS
817  000415  023                    INX     D
818  000416  303  005  001            JMP     RAMMER
819  000421                         PRMLST:
820  000421  244  006                    .ADDR   KPAINI ;KATIM1(2) KEEP ALIVE INITIAL COUNTER
821  000423  377                    .IIF    DF,SCECOD,      .BYTE   -1      ;FOR SCE ADR INIT
822  000424  377                    .IIF    DF,SCECOD,      .BYTE   -1      ;FOR SCE ADR INIT
823  000425  200  010                    .ADDR   MODEO   ;MODDIS(2) INIT THE KLINIK LINE TO MODE 0
824  000427  255  001                    .ADDR   REINI   ;NORMAL INSTR ENDS WILL GO TO REINIT
825  000431  013  042                    .ADDR   ENVBUF  ;ENVPNT(2) APT PNTER FOR ENVELOPES SENT TO THE HOST
826  000433  174                    .BYTE   DEFLTE ;PARBT(1) INIT ENABLES TO BE ON
827  000434  020                    .BYTE   TRPDEF  ;TRAPEN(1) INIT ENABLES THE HARDWARE TRAPS
828  000435  014                    .BYTE   ^O14    ;MTAUBA(1) DEFAULT MAGTAPE UBA NUMBER IS '3'
829  000436  004                    .BYTE   ^O4     ;DSKUBA(1) DEFAULT DISK UBA NUMBER IS '1'
830  000437  010                    .BYTE   ^O10    ;STATE(1) STATE BEGINS WITH DTR TRUE
831  000440  041                    .BYTE   ^O41    ;LSTMSG(1) FIRST APT MSG SHOULD BE 136(NOT OF 41)
832  000441                         D 0,0,,2,0      ;DEN,SLV(5) GET DEFAULT VALUE FOR ALL TAPES
833  000446                         .MTBASE: D 0,0,,772,440 ;MTBASE(5) INITIAL DEFAULT MTA RHBASE ADDRESS
834  000453                         .DSBASE: D 0,0,,776,700 ;DSBASE(5) INITIAL DEFAULT DSK RHBASE ADDRESS
835  000460  377                    .BYTE   -1      ;RPINI(1) SO CAN MAKE IT -1
836  000461  252                    .BYTE   ^O252   ;END OF LIST MARKER
837
838  000462  076  025            XXX235: MVI     A,^O25  ;BIT TO RESET THE UART
839  000464  323  203                    OUT     REMCTL  ;RESET THE KLINIK UART AND ENABLE IT!!
840  000466  076  010                    MVI     A,^O10  ;GET A BIT TO SET 'DATA TERMINAL READY'
841  000470  323  101                    OUT     DTR     ;SET IT.
```

```
  843  000472                        PLINE   INIMS   ;PRINT VERSION AND ID
  (1)  000472  337                   RST     3       ;PRINT LINE OF CHARS
  (1)  000473  205  005              .ADDR   INIMS   ;BUFF TO PRINT
  844  000475                        INTON           ;SET INTERNAL MODE
  (1)  000475  327                   RST     2       ;GO SET INTERNAL MODE
  845  000476  315  032  012         CALL    EM1     ;DO PSUEDO EXAMINE TO SET MEM LATCHES
  846  000501  315  113  011         CALL    EBCMD   ;EXAMINE THE BUS, TO SEE IF ZERO
  847  000504  373                   EI              ;ENABLE INTERRUPTS
  848
  849  000505  315  022  033         CALL    CMP36   ;NOW CHECK RESULTS OF THE 'EB' CMD
  850  000510  012  040              .ADDR   EMBUF   ;THE RESULTS READ FROM THE BUS
  851  000512  364  016              .ADDR   MAD000  ; VERSUS A WORD OF ALL ZEROES
  852
  853  000514  312  163  001         JZ      PWRCHK  ;IF OK, GO TO AUTO BOOT
  854                        ;FALL THRU IF ERRORS ENCOUNTERED
  855  000517                        INTOFF          ;LEAVE INTERNAL MODE
  (1)  000517  367                   RST     6       ;GO EXIT FROM INTERNAL MODE
  856  000520                        PLINE   INIER   ;SAY 'INITIALIZE ERROR'
  (1)  000520  337                   RST     3       ;PRINT LINE OF CHARS
  (1)  000521  013  037              .ADDR   INIER   ;BUFF TO PRINT
  857  000523  303  255  001         JMP     REINI   ;AND GO OUT
  858
```

```
                                ;JUMP TO HERE IF DOING PWR FAIL RESTART
000526  315  263  022  PWR.FAIL: CALL  MICROP ;READ IN POINTERS AND READY TO GO
000531  332  175  032           JC    C.BTERR ;IF BOMBED, GIVE UP
000534  315  043  023         CALL    DMEM2CRAM ;GO READIN THE MICRO-CODE

000537  315  347  021         CALL    BT.GO   ;START MICRO-CODE
000542                       CLRRM   TMPBF2  ;CLEAR A BUFFER
(1) 000542  347              RST     4
(1) 000543  012              .BYTE   10.
(1) 000544  100  040         .ADDR   TMPBF2+5
000546  066  070            MVI     M,^O70  ;POWER FAIL START ADDRESS IS '70'

000550  076  004            MVI     A,4     ;CODE 4 INDICATES PWR FAIL
000552  062  255  040        STA     GOCODE  ;SAVE IN RESTART INDICATOR
000555  315  113  017        CALL    STINT   ;GO START MACHINE, USE INTERNAL MODE
000560  303  255  001        JMP     REINI   ;AND GO LET THINGS HAPPEN

                            ;FIRST THING TO DO IS TO CHECK IF THIS IS A POWER FAIL RESTART
                            ;NOTE: WE ARE STILL IN 'INTERNAL' MODE HERE..THINGS WONT PRINT
000563                      PWRCHK: CLRRM   IOAD    ;CLEAR A BUFFER..WE WILL GENERATE A '100000'
(1) 000563  347              RST     4
(1) 000564  012              .BYTE   10.
(1) 000565  031  040         .ADDR   IOAD+5
000567  043                  INX     H       ;CLRRM PASSES PNTR..WE MAKE IT 'IOAD+1'
000570  066  200            MVI     M,^O200 ;SET BIT THAT MAKES IT '100000'
000572  315  357  012        CALL    EI1     ;GO EXAMINE I/O ADDRESS
000575                       INTOFF          ;CLEAR INTERNAL MODE
(1) 000575  367              RST     6       ;GO EXIT FROM INTERNAL MODE
```

```
  882  000576  016  226        SYSUP:  MVI     C,150.  ;NOW DO AN AUTO BOOT SEQUENCE
  883  000600  041  031  000   SYSUP1: LXI     H,25.   ;WILL DO A DELAY LOOP
  884  000603  315  134  014           CALL    LTLOOP  ;  BY HAND, BY-PASSING USUAL MACRO GENERATION
  885
  886  000606  072  157  040           LDA     RPEND   ;SEE IF A CHAR WAS TYPED TO END THE AUTO BOOT
  887  000611  247                     ANA     A       ;SET CONDITION FLAGS
  888  000612  302  255  001           JNZ     REINI   ;IF A CHAR WAS TYPED, NO AUTO STUFF, GO NULL JOB
  889
  890                           ;WHILE WE HAVEN'T GOT ANYTHING TO DO, MIGHT AS WELL CHECK BOOT SWITCH
  891                           ;TO SEE IF THAT GOT PUSHED WHILE WE WERE SITTING HERE
  892  000615  333  301                IN      BOOTSW  ;***** I/O RD/301 *****
  893  000617  346  002                ANI     2       ;"BOOT" BUTTON PUSHED IS "TRUE .LO."
  894  000621  312  043  002           JZ      NO.5    ;THEREFORE, IF RESULTS .EQ. Z-BIT, THEN BUTTON PUSHED
  895
  896  000624  333  102                IN      ^O102   ;READ AND SEE IF "AC PWR LO" HAPPENED
  897  000626  346  100                ANI     ^O100   ;CHECK THE RESET SIGNAL
  898  000630  312  000  000           JZ      0000    ;ITS TRUE .LO., SO IF TRUE, RESTART THE MACHINE
  899
  900                           ;FALL THROUGH IF NO BOOT BUTTON PUSHED
  901  000633  015                     DCR     C       ;STILL TIMING OUT.. WAIT LONGER
  902  000634  302  200  001           JNZ     SYSUP1  ;BACK WHILE WAITING
  903
  904                           ;NOW SEE IF NEED AUTO BOOT, OR POWER FAIL RECOVERY..
  905  000637  072  014  040           LDA     EMBUF+2 ;FETCH UP BITS 12-19
  906  000642  346  200                ANI     ^O200   ;CHECK THE PWR SAVED BIT
  907  000644  312  126  001           JZ      PWR.FAIL ;IF BIT SET, GO TRY A POWER FAIL RECOVERY
  908
  909                           ;FELL THROUGH IF THIS MUST BE AN AUTO BOOT SEQUENCE
  910  000647                           PLINE   AUTOMS  ;SAY "BT AUTO"
  (1)  000647  337                     RST     3       ;PRINT LINE OF CHARS
  (1)  000650  365  037                .ADDR   AUTOMS  ;BUFF TO PRINT
  911  000652  315  234  021           CALL    BTAUT   ;CALL THE BOOT
```

```
 913                            .SBTTL  **BEGINNING OF BASIC CONSOLE NULL JOB... BEGINS WITH A PROMPT
 914  000655  061  000  044  REINI:  LXI     SP,RAMST+^O2000 ;RESET STACK POINTER TO RE-INIT
 915  000660                         CLRB    EOL     ;GUARANTEE END-OF-LINE CNTR RESET
 916  000662                         CLRB    ERRCD   ;CLEAR CURRENT ERROR CODE
 917  000664                         CLRB    ERRCD+1 ;CLEAR CURRENT ERROR CODE
 918  000666                         CLRB    RPTON   ;ALWAYS CLEAR THE REPEAT FLAG
 919  000670                         CLRB    NOPNT   ;CLR THE 'NO PRINT' FLAG
 920  000672  041  001  041         LXI     H,RPINI ;GET POINTER TO BUFFER WHERE WE SAVE
 921  000675  042  211  040         SHLD    RPLST   ;  THE COMMAND DISPATCH ADDRESSES.USED BY REPEAT
 922  000700  041  255  001         LXI     H,REINI ;IF GOT HERE, SAFE TO GUARANTEE U ALWAYS GET HERE
 923  000703  042  350  040         SHLD    NOREND  ;PASS REINIT LOCATION TO THE NORMAL END DISPATCH
 924  000706  315  160  010         CALL    BFRST   ;RESET TTY INPUT BUFFER
 925  000711  373                   EI              ;MAKE SURE THAT BOMBS RESTORE THE INTERRUPTS
 926
 927  000712  072  156  040         LDA     USRMD   ;GRAB USER MODE FLAG
 928  000715  247                   ANA     A       ;SET CONDITION CODES
 929  000716  302  335  001         JNZ     NULLJ   ;IF USER MODE, NO PROMPTS, NO CR-LF, NOTHING
 930
 931  000721  072  251  040         LDA     MMFLG   ;IF MM MODE WE WILL PRINT NO PROMPTS
 932  000724  247                   ANA     A       ;SET 8080 FLAGS
 933  000725  302  335  001         JNZ     NULLJ   ;SKIP PROMPT IF MM
 934
 935  000730                         PCRLF           ;START FRESH WITH CR-LF
(2)   000730  347                   RST     4
(2)   000731  002                   .BYTE   2
 936  000732                         PLINE   KSPRMT  ;PROMPT IS 'KS10>'
(1)   000732  337                   RST     3       ;PRINT LINE OF CHARS
(1)   000733  053  037               .ADDR   KSPRMT  ;BUFF TO PRINT
```

```
  938                           ;HERE BEGINS THE NULL  STATE LOOP....
  939                           ; CHECK FRONT PANEL BOOT SWITCH
  940  000735  041  222  002    NULLJ:  LXI     H,DCODE ;NULLJ JOB WILL PROCESS COMMANDS ON 'END-OF-LINE'
  941  000740  333  102         NULLW:  IN      ^O102   ;READ AND SEE IF 'AC PWR LO' HAPPENED
  942  000742  346  100                 ANI     ^O100   ;CHECK THE RESET SIGNAL
  943  000744  312  000  000           JZ      0000    ;ITS TRUE .LO.; SO IF TRUE, RESTART THE MACHINE
  944
  945  000747  333  302                 IN      ^O302   ;READ THE STATE OF THE KLINIK SWITCHES
  946  000751  057                      CMA             ;FIX THE HARDWARE INVERSION
  947  000752  117                      MOV     C,A     ;SAVE RESULTS OF READ IN REG 'C'
  948  000753  346  014                 ANI     ^O14    ;OFF ALL BUT THE 2 KLINIK BITS
  949  000755  017                      RRC             ;JUSTIFY 'WORD-WISE'
  950  000756  107                      MOV     B,A     ;SAVE THE KLINIK BITS FOR A SEC.
  951  000757  072  124  040           LDA     KLNKSW  ;GET THE CURRENT KLINIK SWITCH STATE
  952  000762  270                      CMP     B       ;NOW SEE IF KLINIK SWITCH HAS CHANGED SINCE LAST TIME
  953  000763  345                      PUSH    H       ;SAVE H,L  IT HAS OUR DISPATCH ADDRESS
  954  000764  304  327  033           CNZ     KLNKLT  ;IF COMPARE WAS .NE. 0, THEN WE MUST CHANGE LIGHTS
  955  000767  341                      POP     H       ;RETRIEVE H,L SO THAT WE HAVE THE REAL DISPATCH
  956
  957                           ;OK, NOW SEE IF WE ARE WATCHING CARRIER FROM THE KLINIK LINE,
  958                           ;AND IF WE ARE, SEE IF IT HAS GONE AWAY.. IF KLINIK CARRIER GOES
  959                           ;AWAY FOR 2 SECONDS, THEN WE WILL HANGUP THE LINE
  960  000770  171                      MOV     A,C     ;GET STATE OF THE CARRIER INTO ACCUM
  961  000771  346  001                 ANI     1       ;SEE IF IT IS SET
  962  000773  312  004  002           JZ      NO0     ;IF THERE IS NO CARRIER, THEN SEE IF WE CARE
  963
  964                           ;CARRIER WAS TRUE IF WE GOT HERE, SET FLAG SAYING THAT WE MUST WATCH
  965                           ;IF IT DECIDES TO GO AWAY ON US
  966  000776  062  130  040           STA     WATCHC  ;SET FLAG SAYING WATCH THE KLINIK
  967  001001  303  034  002           JMP     NO      ;AND PROCEED WITH THE NULL JOB
  968
  969                           ;GOT HERE IF CARRIER WAS FALSE, SEE IF WE CARE
  970  001004  072  130  040   NO0:    LDA     WATCHC  ;FETCH UP THE 'WATCH' FLAG
  971  001007  247                      ANA     A       ;SET FLAGS
  972  001010  312  034  002           JZ      NO      ;WE DONT CARE, JUMP
  973
  974                           ;CARRIER WENT AWAY ON US. WAIT 2 SECONDS, IF STILL GONE, HANG EM UP
  975  001013  345                      PUSH    H       ;BEST SAVE DISPATCHER TYPE ADDRESS
  976  001014                           LONG,DELAY 2    ;WAIT 2 SECONDS
(1)  001014  041  220  001             LXI     H,200. * 2 ;SET UP THE TIMING COUNT IN H,L REG
(1)  001017  315  134  014             CALL    LTLOOP  ;AND GO DELAY ABOUT 1 SECOND FOR EACH (300 COUNT)
  977  001022  333  302                 IN      CARRIER ;***** I/O RD 302 *****
  978  001024  346  001                 ANI     1       ;SEE IF TRUE(LOW) OR FALSE(HIGH)
  979  001026  304  132  034           CNZ     KILL.KLINIK  ;IT WAS STILL FALSE(I.E HIGH), HANGUP THE LINE
  980  001031  341                      POP     H       ;NOW RESTORE
  981  001032                           CLRB    WATCHC  ;AND SAY TO LEAVE THIS ALONE FOR A WHILE
```

```
  983                           ;ELSE FALL THRU CAUSE ALL OK
  984  001034  333  301          NO:     IN      BOOTSW  ;***** I/O READ 301 *****
  985  001036  346  002                  ANI     2       ;IS BOOT SW SET???(TRUE LO, BECAUSE OF INVERSION)
  986  001040  302  051  002             JNZ     N1      ;SKIP NEXT INSTR IF FALSE(I.E HIGH)
  987
  988  001043  315  231  021  NO.5:     CALL    BOOT    ;OTHERWISE...GO TO BOOT COMMAND
  989  001046  303  335  001             JMP     NULLJ   ;AFTER BOOT, ALL TTY INPUT WAITS ARE KILLED
```

```
  991                           ;CONTINUE NULL STATE STATUS CHECK
  992                           ; CHECK FOR SYSTEM PARITY ERRORS
  993  001051  333  301        N1:     IN      BOOTSW  ;***** I/O RD 301 *****
  994  001053  346  010                ANI     ^D10    ;IS PARITY ERR SIGNAL SET(TRUE ,LO,)
  995  001055  302  072  002           JNZ     NX2     ;SKIP NEXT INSTR IF NOT SET
  996                           ;IF YES SET, MUST CHECK THE PARITY ERROR
  997  001060  072  166  040           LDA     CHKPAR  ;GET FLAG TO SEE IF SHOULD REPORT PARITY ERRORS
  998  001063  247                     ANA     A       ;SET FLAGS
  999  001064  302  035  027           JNZ     RPTPAR  ;IF NOT ZERO, MUST GO REPORT PARITY ERROR
 1000  001067  303  076  002           JMP     N2      ;AND CONTINUE ELSE
 1001
 1002  001072  057              NX2:    CMA             ;IF HERE, SET ACCUM  ,EQ, -1
 1003  001073  062  166  040           STA     CHKPAR  ;AND SET FLAG TO SAY REPORT PARITY ERR
```

```
 1005
 1006                          ;CONTINUE NULL STATE STATUS CHECKS
 1007                          ; CHECK TO SEE IF RUN FLOP HAS DIED
 1008  001076  333  300       N2:     IN      RUNFP   ;;***** I/O READ 300Q *****
 1009  001100  346  010               ANI     ^D10    ;IS HALT LOOP SET??(TRUE ,LO.)
 1010  001102  302  124  002         JNZ     NX3     ;SKIP NEXT INSTR IF NOT SET(CPU MUST BE 'RUNNING')
 1011                          ;FALL THRU IF HALT LOOP SET..
 1012  001105  072  231  040         LDA     CHKHLT  ;GET THE FLAG THAT SAYS IF WE SHLD CHK HALT
 1013  001110  247                   ANA     A       ;SET CONDITION CODES
 1014  001111  365                   PUSH    PSW     ;SAVE FLAGS FOR LATER USE
 1015  001112  304  257  016         CNZ     HLTCM   ;IF FLAG SET, GO REPORT THE HALT
 1016  001115  361                   POP     PSW     ;GET FLAGS BACK
 1017  001116  302  313  002         JNZ     CHKKA   ;IF YOU JUST HALTED, GO SEE IF RELOAD REQUEST
 1018  001121  303  130  002         JMP     N3      ;IF CLR, NO NEED TO REPORT..THEY KNOW
 1019
 1020                          ;THIS AND THE 'CONTINUE' CODE IS THE ONLY CODE
 1021                          ;IN THE CONSOLE THAT SETS THE 'CHECK HALT' FLAG..
 1022                          ;AND U CAN ONLY GET HERE IF THE 'HALT LOOP' FLAG HAS BEEN CLEARED, BY
 1023                          ;ANY MEANS..AND WHEN U GET HERE, THE ACCUM MUST BE ZERO
 1024  001124  057             NX3:    CMA             ;SET ACCUM = -1
 1025  001125  062  231  040         STA     CHKHLT  ;AND ZAP THE CHECK HALT FLAG..
```

```
 1027                           ;CONTINUE NULL STATE STATUS CHECKS
 1028                           ;CHECK TO SEE IF MOS MEMORY REFRESH CYCLE HAS DIED
 1029  001130  333  101        N3:     IN REFERR       ;READ REFRESH ERROR BIT
 1030  001132  346  001                ANI     01      ;REFRESH ERROR TRUE??(TRUE ,LO.)
 1031  001134  302  153  002          JNZ     NX4     ;SKIP FOLLOWING CODE IF NO REFRESH ERROR
 1032
 1033                           ;FALL INTO THIS IF A REFRESH ERROR OCCURS
 1034  001137  072  167  040          LDA     CHKREF  ;GET FLAG THAT SAYS CHECK REFRESH ..SEE IF SHOULD BE REPORTED
 1035  001142  247                    ANA     A       ;SET 8080 FLAGS
 1036  001143  345                    PUSH    H       ;SAVE DISPATCHING ADDRESS
 1037  001144  304  303  032          CNZ     NOREFRESH ;GO REPORT REFRESH ERROR IF NECESSARY
 1038  001147  341                    POP     H       ;RESTORE DISPATCH ADDRESS
 1039  001150  303  157  002          JMP     N4      ;AND CONTINUE AS IF NOTHING HAPPENED
 1040
 1041                           ;THIS IS THE ONLY CODE THAT CAN SET THE "REPORT REFRESH ERROR FLAG"
 1042  001153  057              NX4:    CMA             ;U GOT HERE ON A JZ, SO ACCUM MUST BE ZERO
 1043  001154  062  167  040          STA     CHKREF  ;AND SET FLAG TO -1
```

```
 1045                           ;CONTINUE NULL STATE STATUS CHECK
 1046                           ; CHECK FOR 'END-OF-LINE' OR A RUNNING 10'S 'KEEP-ALIVE'
 1047  001157  072  156  040    N4:     LDA     USRMD   ;IS THIS USER MODE?
 1048  001162  247                    ANA     A       ;SET CONDITION FLAGS
 1049  001163  302  267  002          JNZ     LIVE10  ;IF YES, CHECK 10 INTERRUPTS & KEEP ALIVE
 1050
 1051                           ;BEFORE DOING ANOTHER COMMAND, SEE IF WE ARE ENVELOPING AND MUST SEND
 1052                           ;OUT A PACKET
 1053  001166  072  247  040          LDA     CSLMODE ;ONLY DO ENVELOPE STUFF IF IN MODE 4 OR GREATER
 1054  001171  376  020              CPI     .MODE4  ;CHECK IF MODE 4
 1055  001173  302  203  002          JNZ     EOL.LK  ;NOT MODE 4, PROCEED AS NORMAL
 1056
 1057  001176  345                    PUSH    H       ;SAVE H,L  IT HAS THE DISPATCH ADDRESS
 1058  001177  315  205  034          CALL    DECNET  ;IF SOMETHING THERE, SEND IT OUT
 1059  001202  341                    POP     H       ;NEED H,L AGAIN
 1060
 1061                           ;OTHERWISE, CTY'S EOL FLAG
 1062                           ;BUT FIRST SEE IF THIS IS A REPEAT
 1063  001203  072  213  040    EOL.LK: LDA     RPTON   ;GET THE REPEAT FLAG
 1064  001206  247                    ANA     A       ;SET THE PROCESSOR FLAGS
 1065  001207  302  202  020          JNZ     RPTRTN  ;GO BACK TO THE REPEAT FUNCTION SO THAT HE CAN PROCEED
 1066
 1067  001212  072  204  040          LDA     EOL     ;GET VALUE OF EOL FLAG
 1068  001215  267                    ORA     A       ;SET FLAGS..IS EOL SET??
 1069  001216  312  340  001          JZ      NULLW   ;REMAIN IN NULL JOB LOOP IF NOT SET
 1070
 1071                           ;ELSE GO PROCESS A COMMAND
 1072  001221  351                    PCHL            ;RETURN TO COMMAND IN TTY INPUT, OR DECODE
```

```
 1074                           .SBTTL  **TYPEIN COMMAND DECODER AND DISPATCHER**
 1075  001222  257              DCODE:  XRA     A       ;ZERO OUT THE ACCUM
 1076  001223  107                      MOV     B,A     ;AND ZERO OUT TMP LOCATION
 1077
 1078  001224  021  210  004           LXI     D,CMDLST ;'D,E' NOW POINTS TO COMMAND LIST CHARACTER PAIRS
 1079  001227  052  227  040           LHLD    FIRST   ;'H,L' NOW POINTS TO FIRST CHARACTER TYPED AS COMMAND
 1080  001232  315  220  021           CALL    FNDARG  ;FIRST GO CLR ANY LEADING SPACES OR TABS FROM COMMAND LINE
 1081  001235  332  344  003           JC      NORML   ;IF AT END-OF-COMMAND, ITS A NULL COMMAND
 1082                          ;NOW BEGINS COMMAND LOOK UP LOOP... 'H,L' POINTS TO FIRST COMMAND CHAR.
 1083                          ;WHILE 'D,E' POINTS TO FIRST CHAR IN THE LIST OF ALLOWABLE COMMANDS
 1084  001240  032              DCODL:  LDAX    D       ;CMD LIST CHAR TO ACCUM
 1085  001241  267                      ORA     A       ;IS THIS A ZERO BYTE??
 1086  001242  312  261  002           JZ      NXMDC   ;IF YES,END OF LIST
 1087  001245  023                      INX     D       ;WHILE YOU R AT IT UPDAT CMD LIST PNTR
 1088
 1089  001246  276                      CMP     M       ;COMPARE CMD LIST CHAR WITH TYPED CHAR.
 1090  001247  312  203  003           JZ      MTCH1   ;IF THEY MATCH, BR TO 'MTCH1'
```

```
1092                        ;HERE IF NO MATCH..UPDATE CMD LIST PNTR TO START OF NEXT COMMAND,
1093                        ;THEN CHECK IF TRIED ENTIRE LIST YET..
1094                        ;IF TRIED ENTIRE LIST, REPORT COMMAND ERROR AND BACK TO NULL JOB LOOP.
1095                        ;IF NOT TRIED ENTIRE LIST YET, LOOP BACK AND TRY SOME MORE
1096  001252  023           MISS2: INX     D       ;BUMP PNTR PAST SECOND CHAR OF CMD
1097  001253  023                  INX     D       ;BUMP PAST THE DISPATCH ADDR
1098  001254  023                  INX     D
1099  001255  004                  INR     B       ;UPDATE NUMBER OF 'TRIES'
1100  001256  303  240  002        JMP     DCODL   ;IF NOT, JUMP BACK AND TRY AGAIN
1101
1102                        ;IF FALL TO HERE, WAS ILLEGAL COMMAND.. CLEAR REST
1103                        ;UP TO EOL... TYPE ERROR MESSSAGE AND BACK TO NULL JOB LOOP
1104                        ;FIRST STEP IS CLEARING BAD COMMAND FROM COMMAND BUFFER
1105  001261               NXMDC: PLINE   CMDNG   ;COMMAND NO GOOD
 (1)  001261  337                  RST     3       ;PRINT LINE OF CHARS
 (1)  001262  026  037             .ADDR   CMDNG   ;BUFF TO PRINT
1106
1107  001264  303  153  032        JMP     MMERR   ;AND BACK TO PROMPT..
```

```
                                .SBTTL ****CTY SERVICE****
                                ;CODE ENTERED FROM THE NULL JOB LOOP WHENEVER WE ARE IN USER MODE(MODE 3)
                                ;CODE CHECKS TO SEE IF CHAR IS AVAILABLE FROM THE 10;& ALSO IF KEEP ALIVE
                                ;COUNT IS ACTIVE. WE ALSO CHECK FOR REBOOT REQUESTS (AS IN 20 SYSTEM MONITOR'S
                                ;"BUGHALTS")
001267  333  301        LIVE10: IN      SMSTS   ;SEE IF INTERRUPT FROM THE 10 IS PENDING
001271  247                    ANA     A       ;SET CONDITION CODES
001272  365                    PUSH    PSW     ;SAVE FLAGS FOR A LITTLE BIT
001273  364  237  007          CP      CHRRDY  ;IF PLUS, A CHARACTER IS READY, GO PROCESS

                                ;FIRST, WE ONLY WANT TO CHECK THE KEEP-ALIVE COUNTER EVERY SECOND.
                                ;FIRST WE WILL CHECK OUR COUNTERS AND SEE IF TIME TO CHECK KEEP-ALIVE DATA
001276  361                    POP     PSW     ;GET FLAGS BACK
001277  364  011  003          CP      FAKLIT  ;IF TYPED A CHAR, THEN DOWN COUNT FOR TIME WASTED
001302  312  313  002          JZ      CHKKA   ;IF ZERO, DO WHAT MUST BE DONE

001305  315  023  003          CALL    DTIME   ;ALWAYS DO IT AT LEAST ONCE
001310  302  335  001          JNZ     NULLJ   ;IF NOT ZERO, IT HASN'T BEEN A SECOND YET. BACK NULL

                                ;WHEN YOU FALL TO HERE, IT IS TIME TO CHECK FOR "KEEP-ALIVE" OR "RELOAD"
001313  041  244  006  CHKKA:  LXI     H,KPAINI ;FIRST WE MUST RESET OUR MAJOR LOOP COUNTER
001316  042  342  040          SHLD    KATIM1  ;EACH NUM IN THIS LOC .EQ. .444 SECONDS BETWEEN CHECKS

001321                         INTON           ;DON'T PRINT THIS CRUD
 (1)  001321  327              RST     2       ;GO SET INTERNAL MODE
001322  363                    DI              ;NO INTERRUPTS
001323                         EXAM    31      ;MOS MEMORY LOC 31 HAS THE INFORMATION
 (1)  001323  315  216  030    CALL    EXAMSH  ;AND DO EXAM ASSUMING SHORT ADDRESS
 (1)  001326  031  000          .ADDR   31      ;ADDR TO BE ZAPPED PASSED AS TRAILING ARG
001330  373                    EI              ;OK,,INTERRUPTS

001331                         INTOFF          ;INTERNAL MODE OFF NOW
 (1)  001331  367              RST     6       ;GO EXIT FROM INTERNAL MODE
001332  072  015  040          LDA     EMBUF+3 ;GRAB THE RELOAD BITS,, WE CERTAINLY NEED THOSE
001335  027                    RAL             ;"FORCED RELOAD" SHIFT INTO CARRY
001336  332  035  003          JC      FRELOAD ;IF "RELOAD" SET, GO EXECUTE A RELOAD
```

```
 1142                          ;NOW SEE IF KEEP-ALIVE ACTIVE
 1143  001341  027                    RAL             ;'KEEP-ALIVE' BIT INTO THE C-BIT
 1144  001342  322  335  001          JNC     NULLJ   ;IF ITS NOT SET, GO BACK TO NULL JOB
 1145
 1146                          ;KEEPING A WATCH ON THE KEEP ALIVE COUNT...BLINKY LIGHTS
 1147  001345  072  360  040          LDA     STATE   ;GET CURRENT STATE OF THE LIGHTS
 1148  001350  127                    MOV     D,A     ;SAVE IT JUST FOR A LITTLE BIT
 1149  001351  346  373              ANI     STMSK   ;CLR THE 'STATE' LIGHT FROM THE CURRENT SELECTIONS
 1150  001353  137                    MOV     E,A     ;AND NOW SAVE THIS LITTLE BIT IN A REGISTER
 1151
 1152  001354  172                    MOV     A,D     ;GET BACK THE ORIGINAL 'STATE'
 1153  001355  057                    CMA             ;'BLINK'
 1154  001356  346  004              ANI     STBIT   ;ONLY WANT TO BLINK THE SINGLE LIGHT
 1155  001360  263                    ORA     E       ;THROW IN THE PART THAT IS AS WAS..
 1156  001361  062  360  040          STA     STATE   ;PUT THE WHOLE MESS BACK, THE NULL LOOP WILL BLINKY
 1157  001364  323  101              OUT     LIGHTS  ;BLINK THE LIGHTS HERE
 1158  001366  372  335  001          JM      NULLJ   ;AND IF MINUS, WE ARE "SHUTING DOWN", LET LIGHTS GO
 1159
 1160                          ;WE DID AN EXAM 31 A VERY SHORT TIME AGO, SO THE EXAMINE BUFFER SHOULD STILL
 1161                          ;HAVE A VALID 'KEEP-ALIVE' UPDATE COUNT IN IT
 1162  001371  041  250  040          LXI     H,KACNTR ;GET POINTER TO THE PREVIOUS 'KA' COUNT
 1163  001374  072  013  040          LDA     EMBUF+1 ;GET WHAT SHOULD BE AN UPDATED 'KA' COUNT
 1164  001377  276                    CMP     M       ;COMPARE..BETTER BE DIFFERENT
 1165  001400  312  130  003          JZ      DIEING  ;IF SAME, GO  SEE IF CPU IS REALLY DIEING
 1166                          ;FALL THRU IF THE COUNTS WERE DIFF..SAVE THE NEW COUNT
 1167  001403  167                    MOV     M,A     ;NEW COUNT GOES TO RAM, OVERWRITES THE OLD
 1168  001404                         CLRB    DIECNT  ;CLEAR THAT DIE COUNT
 1169  001406  303  335  001          JMP     NULLJ   ;BACK TO THE NULL LOOP
 1170
 1171                          ;ROUTINE FOR WHEN DOING CTY OUT PUT THE LIGHTS STILL BLINK AT A REASONABLE
 1172                          ;RATE
 1173  001411  052  342  040  FAKLIT: LHLD    KATIM1  ;GET CURRENT COUNT FOR BETWEEN THE LIGHTS
 1174  001414  175                    MOV     A,L     ;GET LO ORDER PIECE
 1175  001415  346  374              ANI     ^O374   ;TWEEK LITE COUNT, CAUSE OUR TYPING MESSED UP TIMER LOOP
 1176  001417  157                    MOV     L,A     ;PUT LO ORDER PIECE BACK(MASKING .EQV. TO SUBTRACT 100)
 1177  001420  303  027  003          JMP     DTM1    ;AND PROCEED
 1178                          ;ROUTINE TO DOWN COUNT KEEP ALIVE COUNTER
 1179  001423  052  342  040  DTIME: LHLD    KATIM1  ;GET A 16-BIT MINOR LOOP COUNTER
 1180  001426  053                    DCX     H       ;DECREMENT THE COUNTER
 1181  001427  042  342  040  DTM1: SHLD     KATIM1  ;AND PUT IT BACK
 1182  001432  175                    MOV     A,L     ;NOW CHECK THE COUNT FOR .EQ. 0
 1183  001433  264                    ORA     H       ;'OR' HI PIECE WITH THE 'LO' PIECE
 1184  001434  311                    RET
```

```
1166                        ;ROUTINE TO ENTERED WHEN THE 'FORCE RELOAD' BIT HAS BEEN SET BY A RUNNING
1187                        ;PROGRAM IN THE KS10
1188  001435                FRELOAD: CLRB   NOPNT   ;CLEAR THE NO PRINT FLAG
1189  001437                         PLINE  FRCMSG  ;PRINT MESSAGE TO INDICATE THE FORCED RELOAD
 (1)  001437  337                   RST    3       ;PRINT LINE OF CHARS
 (1)  001440  336  037              ,ADDR  FRCMSG  ;BUFF TO PRINT
1190  001442  076  002              MVI    A,2     ;BIT 34 IS THE APPROPRIATE GUY ON A FORCED RELOAD
1191  001444  062  255  040         STA    GOCODE  ;SET BIT IN THE 8080 RAM BUFFER
1192  001447  333  300              IN     RUNFP   ;NOW SEE IF WE MUST HALT THE PROCESSOR
1193  001451  346  010              ANI    ^O10    ;WE DID NOT INVERT SIGNAL.0=HALTED, 1=RUNNING
1194
1195  001453                         INTON          ;DONT PRINT WHAT HAPPENS TO THE HALT
 (1)  001453  327                   RST    2       ;GO SET INTERNAL MODE
1196  001454  304  207  017         CNZ    HACMD   ;MAKE SURE THE PROCESSOR IS STOPPED
1197  001457                         INTOFF         ;OK TO PRINT NOW
 (1)  001457  367                   RST    6       ;GO EXIT FROM INTERNAL MODE
1198  001460  072  256  040         LDA    SECRET  ;SECRET LOCATION
1199  001463  247                   ANA    A       ;SET 8080 FLAGS
1200  001464  302  255  001         JNZ    REINI   ;IF FLAG SET, WE WILL NOT DO AUTO RELOADS
1201
1202  001467  021  004  002         LXI    D,^O1004 ;POINTER TO THE MONITOR PRE-BOOT
1203  001472  315  266  022         CALL   FILEINIT ;GO READ IN THE MONITOR PRE-BOOT
1204  001475  332  231  032          JC    L.BTERR ;IF ERROR, FATAL OUT
1205
1206  001500  041  001  000         LXI    H,1     ;NO ERR, NOW GO START THE MICRO-CODE AT LOC 1
1207  001503  315  215  016         CALL   SM1.5   ;START MICROCODE, ADDRESS PASSED IN H,L
1208  001506                         LONG.DELAY 2   ;GUARANTEE THAT THE 'SM 1' HAS TIME TO FINISH
 (1)  001506  041  220  001         LXI    H,200. * 2 ;SET UP THE TIMING COUNT IN H,L REG
 (1)  001511  315  134  014         CALL   LTLOOP  ;AND GO DELAY ABOUT 1 SECOND FOR EACH (300 COUNT)
1209  001514  315  355  021         CALL   BT.GO1  ;NOW FIX PARITY AND OTHER THINGS THAT WE BROKE
1210  001517  315  320  021         CALL   INFOBT  ;AND PASS THE SOURCE OF THE PRE-BOOT IN MOS MEM
1211
1212  001522  315  263  021         CALL   LB.GO1  ;AND GO EXECUTE THE BOOT CODE
1213  001525  303  335  001         JMP    NULLJ   ;AND BACK TO NULL JOB
1214
1215  001530  041  257  040  DIEING: LXI    H,DIECNT ;GET THE NUMBER OF CONSECUTIVE 'NO CHANGES' IN KA CNT
1216  001533  064                   INR    M       ;INCREMENT
1217  001534  176                   MOV    A,M     ;GET COUNT INTO ACCUM
1218  001535  376  043              CPI    KATIMX  ;NOW SEE IF OFFICIAL DEATH
1219  001537  372  335  001         JM     NULLJ   ;IF NOT YET, GO BACK AND WAIT FOR MORE
1220
1221                        ;ELSE, FALL ON THROUGH TO THE RELOAD CODE
```

```
1223                        ;ROUTINE ENTERED WHEN THE "KEEP-ALIVE" COUNT DOES NOT CHANGE, AND WE
1224                        ;ISSUE A RELOAD BECAUSE WE BELIEVE THAT THE PROGRAM RUNNING IN THE KS10
1225                        ;HAS DIED
1226  001542                        CLRB    NOPNT   ;CLEAR THE NO PRINT FLAG
1227  001544                        CLRB    DIECNT  ;AND RESET THE DIE COUNT
1228  001546                        PLINE   KAMSG   ;PRINT MESSAGE TO INDICATE KEEP-ALIVE FAILURE
 (1)  001546  337                   RST     3       ;PRINT LINE OF CHARS
 (1)  001547  331  037              .ADDR   KAMSG   ;BUFF TO PRINT
1229
1230  001551                        INTON           ;DONT PRINT ANY OF THIS STUFF
 (1)  001551  327                   RST     2       ;GO SET INTERNAL MODE
1231  001552  315  207  017         CALL    HACMD   ;FIRST, STOP THE MACHINE
1232  001555  363                   DI              ;NO INTERRUPTS, TILL THE EXAMINE IS COMPLETE
1233  001556                        EXAM    71      ;EXAMINE THE INSTRUCTION IN 71
 (1)  001556  315  216  030         CALL    EXAMSH  ;AND DO EXAM ASSUMING SHORT ADDRESS
 (1)  001561  071  000              .ADDR   71      ;ADDR TO BE ZAPPED PASSED AS TRAILING ARG
1234  001563  373                   EI              ;OK TO INTERRUPT NOW
1235
1236  001564  076  001             MVI     A,1     ;KEEP ALIVE CODE IS A 1
1237  001566  062  255  040         STA     GOCODE  ;SAVE FOR PASSING TO KS10
1238  001571  315  043  017         CALL    EXINTM  ;NOW GO EXECUTE LOC 71 IN PAGE 0 OF MONITOR SPACE
1239  001574  315  164  017         CALL    COCMD   ;LET THE PROCESSOR RESUME
1240  001577                        INTOFF          ;MAY RESUME PRINTING THIS STUFF
 (1)  001577  367                   RST     6       ;GO EXIT FROM INTERNAL MODE
1241  001600  303  335  001         JMP     NULLJ   ;AND BACK TO NULL JOB
```

```
1243                        .SBTTL  **COMMAND DECODER CONTINUED**
1244                        ;CONTINUE COMMAND DECODER.....
1245                        ;HERE IF FIRST CHARACTER MATCHED.SEE IF SECOND CHAR MATCHED.
1246  001603  043           MTCH1: INX     H       ;INPUT BUFFER NOW PNTS TO SECOND CHAR TYPED
1247  001604  032                  LDAX    D       ;ACCUM GETS SECOND 'EXPECTED' CHAR FROM COMMAND LIST
1248  001605  276                  CMP     M       ;NOW. DOES SECOND CHARACTER MATCH??
1249  001606  312  215  003        JZ      MTCH2   ;JUMP IF YES, SECOND CHAR MATCHES.
1250
1251                        ;HERE IF SECOND CHARACTER DID NOT MATCH.RESET 'H,L' & GO BACK TO TRY AGAIN
1252  001611  053                  DCX     H       ;RESET 'H,L'
1253  001612  303  252  002        JMP     MISS2   ;AND CONTINUE PROCESSING
1254
1255                        ;GET TO HERE IF SECOND CHARACTER MATCHED
1256  001615  023           MTCH2: INX     D       ;UPDATE PAST 2ND CHAR
1257  001616  043                  INX     H       ;UPDATE BUFFER POINTER
1258
1259  001617  315  245  033        CALL    SEPCHR  ;GET RID OF SEPARATORS
```

```
1261                        ;'H,L' SHOULD NOW POINTS TO THE CMD BUFF
1262                        ;'D,E' NOW POINTS TO THE DISP ADDR
1263                        ;OR ELSE TO 'EOL' CHAR IF CMD HAS NO ARGS.. ANYTHING ELSE IS COMMAND ERROR
1264  001622  042 223 040          SHLD    .ARG1   ;SAVE 'H,L' POINTER TO FIRST ARG.
1265
1266  001625  170                 MOV     A,B     ;NOW LOAD ACCUM WITH NUMBER 'TRIES' TO FIND MATCH
1267  001626  353                 XCHG            ;DISP POINTER NOW TO 'H,L'
1268
1269  001627  136                 MOV     E,M     ;SET LOW ODER HALF OF 'TABLE ENTRY' INTO 'E'
1270  001630  043                 INX     H       ;BUMP POINTER
1271  001631  126                 MOV     D,M     ;SET HIGH ORDER HALF OF 'TABLE ENTRY' INTO 'D'
1272  001632  041 344 003          LXI     H,NORML ;SET A RETURN VALUE
1273  001635  345                 PUSH    H       ;AND PLACE ON STACK FOR RETURNS
1274
1275                        ;PIECE OF CODE TO SAVE EACH COMMAND DISPATCH ADDRESS IN THE RAM BUFFER
1276                        ;SO THAT THE REPEAT FUNCTION CAN DISPATCH THRU THE LIST WITHOUT DECODING
1277                        ;THE TYPED IN COMMAND STRING AGAIN..
1278  001636  247                 ANA     A       ;CHECK IF THIS IS RP COMMAND..IF YES MUST
1279  001637  062 000 040          STA     T80DT   ;SAVE WHICH COMMAND IS BEING EXECUTED
1280  001642  312 271 003          JZ      CMDGO   ;JUMP SO THAT CMD LST BUFFER NO CHANGED
1281
1282  001645  072 131 040          LDA     CMDS.,  ;SEE IF THIS IS THE FIRST COMMAND IN A LINE
1283  001650  247                 ANA     A       ;SET FLAGS
1284  001651  314 303 020          CZ      RPNEW   ;IF IS THE FIRST COMMAND, RESET REPEAT BUFFER POINTERS
1285
1286  001654  052 211 040          LHLD    RPLST   ;GET POINTER TO CURRENT FREE BUFFER LOCATION
1287  001657  162                 MOV     M,D     ;SAVE HI ORDER PIECE OF DISPATCH ADDRESS
1288  001660  043                 INX     H       ;UPDATE MEM POINTER
1289  001661  163                 MOV     M,E     ;SAVE LO ORDER PIECE OF DISPATCH ADDRESS
1290  001662  043                 INX     H       ;UPDATE POINTER TO FIRST FREE..
1291  001663  257                 XRA     A       ;CLR ACCUM
1292  001664  057                 CMA             ;IN ORDER TO MAKE IT -1
1293  001665  167                 MOV     M,A     ;SET -1 AS END-OF-LIST INDICATOR
1294  001666  042 211 040          SHLD    RPLST   ;AND RESTORE THE POINTER
```

```
 1296                           ;CONTINUE THE DISPATCH
 1297  001671  353              CMDGO:  XCHG            ;SET 'TABLE ENTRY' INTO 'H,L'
 1298                           ;AND DISPATCH TO ACTUAL COMMAND CODE
 1299  001672  315  344  032            CALL    EOCML   ;SET 'C-BIT' TO SAY END-OF-LINE IF TRUE
 1300  001675  365                      PUSH    PSW     ;SAVE STATE OF PROCESSOR FLAGS
 1301  001676  324  317  003            CNC     REMARG  ;IF NO C-BIT, CMD HAD ARG.,MUST REMEMBER IT
 1302
 1303                           ;FINALLY SEE IF COMMAND REQUIRES AN ARG
 1304  001701  322  311  003            JNC     CMDGO9  ;IF REQUIRES NO ARG, GO GO GO
 1305
 1306                           ;FALL HERE IF COMMAND HAD NO ARG.,SEE IF IT SHOULD HAVE HAD ONE
 1307  001704  174                      MOV     A,H     ;GET HI ORDER OF DISP ADDRESS
 1308  001705  027                      RAL             ;SHIFT HI ORDER BIT (BIT15) INTO THE C-BIT
 1309  001706  332  142  032            JC      RRARG   ;WELL IF SET, IT NEEDED ARG., REPORT THAT HAD NONE
 1310
 1311                           ;ELSE ALL OK., CLEAR BIT15 IF SET AND PROCEED
 1312  001711  174              CMDGO9: MOV     A,H     ;HI ORDER TO ACCUM
 1313  001712  346  177                ANI     ^O177   ;CLEAR BIT 15 NO MATTER WHAT
 1314  001714  147                      MOV     H,A     ;PUT HI ORDER BACK
 1315  001715  361                      POP     PSW     ;GET THE PROCESSOR FLAGS BACK
 1316  001716  351                      PCHL            ;ADDR TO PC TAKES THE DISPATCH
```

```
1318                            ;ROUTINE TO REMEMBER IF COMMAND HAD AN ARG.,USED BY REPEAT FUNCTION
1319  001717  365                REMARG: PUSH    PSW     ;SAVE FLAGS
1320  001720  072  000  040             LDA     T80DT   ;SEE IF THIS WAS A REPEAT, IN WHICH CASE, DO NOTHING
1321  001723  247                       ANA     A       ;SET CPU FLAGS
1322  001724  312  342  003             JZ      REMAR1  ;IF WAS RP, GET OUT
1323
1324  001727  345                       PUSH    H       ;AND SAVE DISPATCH ADDRESS
1325  001730  052  211  040             LHLD    RPLST   ;GET THIS DISPATCH FROM REPEAT LIST
1326  001733  053                       DCX     H       ;BACK UP ADDR POINTER TO POINT
1327  001734  053                       DCX     H       ;  HI ORDER PIECE OF ADDRESS
1328  001735  176                       MOV     A,M     ;NOW GET HI ORDER PIECE INTO ACCUM
1329  001736  366  200                 ORI     ^O200   ;ADD SIGN BIT TO REMEMBER ARG
1330  001740  167                       MOV     M,A     ;NOW PUT IT BACK
1331  001741  341                       POP     H       ;RESTORE DISPATCH ADDRESS
1332  001742  361                REMAR1: POP     PSW     ;AND RESTORE C-BIT FROM PREVIOUS "EOCML"
1333  001743  311                       RET             ;AND RETURN
```

```
 1335                           ;THIS IS 'NORML', FOR NORMAL RETURNS., IT SETS UP PREVIOUS POINTERS
 1336                           ;AND THEN GOES BACK TO PROMPT
 1337  001744  041  204  040    NORML:  LXI     H,EOL   ;GET PNTR TO COMMAND COUNT
 1338  001747  176                    MOV     A,M     ;COPY TO ACCUM
 1339  001750  075                    DCR     A       ;DECREMENT IT
 1340  001751  167                    MOV     M,A     ;PUT BACK WHERE U GOT IT
 1341                           ;IF COUNT WAS DOWN TO ZERO, THEN RESET IT
 1342  001752  075                    DCR     A       ;IF COUNT WAS ZERO, THIS MAKES IT NEGATIVE
 1343  001753  372  004  004         JM      NORDIS  ;IF EOL HAS GONE MINUS, TAKE NORMAL DISPATCH
 1344  001756  315  371  003         CALL    FXNXT   ;OTHER WISE BE CLEVER & CRYPTIC
 1345  001761                       CLRB    ERRCD   ;AND CLEAR ERROR CODE
 1346  001763  041  222  002         LXI     H,DCODE ;FIX H,L FOR NORMAL NULL JOB
 1347
 1348  001766  303  051  002         JMP     N1      ;AND FINALLY, ALL ELSE GOES TO NULL LOOP
 1349  001771  052  223  040    FXNXT:  LHLD    .ARG1   ;GET CMD PNTR
 1350  001774  043                    INX     H       ;UPDATE IT PAST THE CURRENT EOL CHAR(, OR CR-LF)
 1351  001775  042  227  040         SHLD    FIRST   ;FIX CURRENT CMD LINE PNTR
 1352  002000  311                    RET             ;AND THATS ENOUGH FOR NOW
 1353
 1354                           ;CODE FOR BUFFER OVER FLOW
 1355  002001                 BFOVR:  PLINE   BV      ;"BUFFER OVERFLOW
  (1)  002001  337                    RST     3       ;PRINT.LINE OF CHARS
  (1)  002002  021  037                .ADDR   BV      ;BUFF TO PRINT
 1356  002004  052  350  040    NORDIS: LHLD    NOREND  ;GET THE CURRENT DISPATCH ADDRESS FOR NORMAL ENDS
 1357  002007  351                    PCHL            ;AND GO
```

```
                                ;SUBROUTINE TO PRINT A SINGLE CHARACTER..
                                ;CHARACTER TO BE PRINTED IS PASSED IN THE ACCUM..
                                ;IF THE UART SHOULD FAIL AND NEVER REACH THE TRANSMITTER READY STAT
                                ;THE 8080 WILL HANG IN THIS LOOP FOREVER TRYING TO PRINT
002010  365           PCHR:   PUSH    PSW     ;SAVE CHARACTER ON THE STACK
002011  072 120 040           LDA     NOPNT   ;GET NO PRINT FLAG
002014  267                   ORA     A       ;IS IT SET??
002015  312 022 004           JZ      PCHR0   ;IF NOT CONTINUE AS NORMAL

002020  361                   POP     PSW     ;IF IT IS..NO PRINTING
002021  311                   RET             ;SO RETURN

002022  072 247 040   PCHR0: LDA     CSLMODE ;GET CURRENT KLINIK LINE MODE
002025  376 020               CPI     .MODE4  ;IS THIS APT MODE??
002027  302 057 004           JNZ     PCHR1   ;IF NO, PRINT

                                ;HERE IF DOING APT AND MUST MERELY STACK CHARS TO BE ENVELOPED AND SENT OUT
                                ;A LITTLE LATER, THE CHARACTER TO BE STACKED IS SITTING ON THE TOP OF THE
                                ;STACK
002032  361                   POP     PSW     ;NOW GET THE CHARACTER THAT WAS STACKED
002033  345                   PUSH    H       ;MUST SAVE H,L IN HERE
002034  052 352 040           LHLD    ENVPNT  ;GET THE POINTER TO THE ENVELOPE
002037  167                   MOV     M,A     ;PUT CHARACTER INTO THE BUFFER
002040  043                   INX     H       ;UPDATE THE POINTER
002041  066 000               MVI     M,0     ;GUARANTEE LAST BYTE IS A ZERO
002043  042 352 040           SHLD    ENVPNT  ;PUT IT BACK WHERE U GOT IT
002046  341                   POP     H       ;AND RESTORE REG
002047  376 015               CPI     CRCHR   ;IF WE ARE BUFFERING A CR, MUST SET THE FLAG
002051  300                   RNZ             ;IF NOT A CR, JUST LEAVE
002052  062 252 040           STA     MAILFG  ;ELSE SET THE FLAG
002055  311                   RET             ;AND OUT

002056  365           PCHR1Z: PUSH    PSW     ;CHAR ON STACK FOR A BIT
002057  333 201       PCHR1:  IN      CTYCTL  ;GET UART STATUS
002061  346 001               ANI     01      ;CHECK BITS TO SEE IS XMITTER READY??
002063  312 057 004           JZ      PCHR1   ;JUMP BACK IF NOT READY YET

                                ;NOW,BEFORE PRINTING, SEE IF NEED TO TYPE TO KLINIK LINE TOO.
002066  072 247 040           LDA     CSLMODE ;GET CURRENT KLINIK LINE MODE
002071  376 010               CPI     .MODE3  ;KLINIK IN PARALLEL MODE??
002073  302 113 004           JNZ     PCHROV  ;IF NOT, JUST GO PRINT

                                ;FALL THROUGH IF NEED KLINIK TOO
002076  333 203       PCHR2:  IN      REMCTL  ;GET KLINIK UART STATUS
002100  346 001               ANI     01      ;CHECK THE READY BIT
002102  312 076 004           JZ      PCHR2   ;IF NOT READY, GO BACK AND TRY AGAIN
002105  361                   POP     PSW     ;GET CHAR OFF STACK WHEN THINGS ARE READY
002106  323 202               OUT     REMDAT  ;PRINT CHAR ON THE KLINIK LINE
002110  323 200               OUT     CTYDAT  ;PRINT ON CTY
002112  311                   RET             ;AND BACK TO CALLER

002113  361           PCHROV: POP     PSW     ;FINALLY READY..GET CHAR FROM STACK
002114  323 200               OUT     CTYDAT  ;SEND CHARACTER
002116  311                   RET             ;AND RETURN
```

```
1414                        ;SUBROUTINE KCHR, FOR PRINTING A SINGLE CHARACTER ON THE KLINIK LINE ONLY
1415                        ;PARTICULARLY USEFUL FOR THE '?NA' AND 'PW:' MESSAGES
1416                        ;CHARACTER TO BE PRINTED CAN BE A TRAILING ARG, OR YOU CAN CALL THIS ROUTINE
1417                        ;IN THE MIDDLE AND PASS THE CHAR TO BE PRINTED IN THE ACCUM
1418  002117  343           KCHR:   XTHL            ;SWAP STACK TOP WITH H,L
1419  002120  176                   MOV     A,M     ;GET THE TRAILING ARG FROM PROM
1420  002121  043                   INX     H       ;UPDATE RETURN ADDRESS PAST THE TRAILING ARG
1421  002122  343                   XTHL            ;PUT THE RETURN BACK ON THE STACK
1422  002123  365           KCHR0:  PUSH    PSW     ;SAVE THE CHARACTER JUST FOR A LITTLE BIT
1423  002124  333  203      KCHR1:  IN      REMCTL  ;GET KLINIK UART STATUS
1424  002126  346  001              ANI     01      ;SEE IF UART IS READY
1425  002130  312  124  004        JZ      KCHR1   ;LOOP TILL IT IS
1426
1427  002133  361                   POP     PSW     ;READY NOW, GET THE CHAR OFF THE STACK
1428  002134  323  202              OUT     REMDAT  ;SEND TO UART
1429  002136  311                   RET             ;AND BACK TO CALLER
```

```
1431                        ;SUBROUTINE KLINE, FOR PRINTING A LINE OF CHARACTERS, A TRAILING ARG
1432                        ;POINTING TO THE STRING TO BE PRINTED IS USED. '\' MEANS <CR-LF>.
1433  002137  343          KLINE:  XTHL            ;SWAP STACK, GET POINTER TO TRAILING ARG TO H,L
1434  002140  315  050  033        CALL    TARG1   ;GET POINTER TO ARG INTO D,E
1435  002143  343                  XTHL            ;FIX RETURN ADDRESS
1436
1437  002144  032          KLINE1: LDAX    D       ;GET FIRST CHAR INTO ACCUM
1438  002145  023                  INX     D       ;UPDATE THE CHARACTER POINTER
1439
1440  002146  247                  ANA     A       ;WELL THEN, SEE IF CHAR IS 0, MEANING END OF STRING
1441  002147  310                  RZ              ;OUT IF YES
1442
1443                        ;WELL THEN , MIGHT AS WELL GO PRINT THE THING
1444  002150  315  123  004        CALL    KCHRO   ;GO PRINT THE CHARACTER
1445  002153  303  144  004        JMP     KLINE1  ;AND WHEN RETURN, GO FETCH UP THE NEXT CHAR
```

```
 1447                           ;SUBROUTINE TO PRINT A LINE OF CHARACTERS..
 1448                           ;POINTER TO THE LINE OF CHARACTERS TO BE PRINTED IS PASSED IN
 1449                           ;'H,L' REGISTER..NO REGISTERS ARE DESTROYED BY THIS ROUTINE
 1450                           ;THE END-OF-MESSAGE FOR THE LINE OF CHARACTERS TO BE
 1451                           ;PRINTED IS INDICATED BY A '00' BYTE AT THE END OF THE MESSAGE TEXT
 1452  002156  343              PLNE:   XTHL            ;REPLACE RETURN..PUT ON STACK
 1453  002157  353                      XCHG            ;TRAILING ARG FROM 'D,E' TO 'H,L'
 1454
 1455  002160  176              PLN1:   MOV     A,M     ;GET CHARACTER FROM MEM.
 1456  002161  043                      INX     H       ;INCREMENT TO NEXT CHARACTER TO BE PRINTED
 1457
 1458  002162  376  134                  CPI     BSLASH  ;IS THIS A BACK SLASH(I.E. IN-LINE CRLF)
 1459  002164  312  174  004            JZ      PLN2    ;JUMP IF YES...
 1460
 1461  002167  267                      ORA     A       ;IS IT A '00' BYTE???
 1462  002170  310                      RZ              ;RETURN IF DONE
 1463
 1464                           ;FALL HERE IF GOTTA REAL CHAR..
 1465  002171  315  010  004            CALL    PCHR    ;GO PRINT CHAR IN ACCUM
 1466  002174  314  203  004   PLN2:   CZ      CRLFIN  ;GET HERE ON ZERO FLAG,ONLY IF NEED CRLF
 1467  002177  303  160  004            JMP     PLN1    ;CONTINUE LOOP
 1468                           ;SUBROUTINE TO PRINT A CARRIAGE RETURN-LINE FEED
 1469                           ;NO REGISTERS DESTROYED..JUST CALL TO GET YOUR
 1470                           ;<CR><LF> PRINTED
 1471  002202  341              .CRLF:  POP     H       ;MUST FIX THE STACK
 1472  002203                   CRLFIN: PCHAR   CRCHR   ;PRINT CARRIAGE RETURN
  (1)  002203  317                      RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (1)  002204  015                      .BYTE   CRCHR   ;CHAR TO PRINT
 1473  002205                         PCHAR   LFCHR   ;PRINT LINE FEED
  (1)  002205  317                      RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (1)  002206  012                      .BYTE   LFCHR   ;CHAR TO PRINT
 1474  002207  311                      RET             ;RETURN
 1475
```

```
                        .SBTTL  **COMMAND DISPATCH LIST**
                        ;THIS LIST CONTAINS ALL CHARACTER PAIRS WHICH ARE CONSIDERED
                        ;LEGITIMATE COMMANDS TO THE KS10 CONSOLE
002210                  CMDLST:
002210  122  120               .BYTE   'R,'P   ;*REPEAT IN FAST LOOP*
002212  140  020               .ADDR   RPCMD
002214  104  116               .BYTE   'D,'N   ;*DEPOSIT NEXT*
002216  323  212               .ADDR   BIT15!DNCMD
002220  104  103               .BYTE   'D,'C   ;*DEPOSIT CRAM*
002222  100  216               .ADDR   BIT15!DCCMD
002224  104  115               .BYTE   'D,'M   ;*DEPOSIT MEMORY*
002226  224  212               .ADDR   BIT15!DMCMD
002230  114  103               .BYTE   'L,'C   ;*LOAD CRAM ADDRESS*
002232  331  213               .ADDR   BIT15!LCCMD

002234  114  101               .BYTE   'L,'A   ;*LOAD MEMORY ADDRESS*
002236  120  213               .ADDR   BIT15!LACMD
002240  104  111               .BYTE   'D,'I   ;*DEPOSIT I/O*
002242  012  213               .ADDR   BIT15!DICMD
002244  114  111               .BYTE   'L,'I   ;*LOAD I/O ADDRESS*
002246  125  213               .ADDR   BIT15!LICMD
002250  104  102               .BYTE   'D,'B   ;*DEPOSIT BUS*
002252  240  011               .ADDR   DBCMD
002254  104  113               .BYTE   'D,'K   ;*DEPOSIT KONSOLE*
002256  146  213               .ADDR   BIT15!DKCMD

002260  114  113               .BYTE   'L,'K   ;*LOAD ADR FOR  KONSOLE*
002262  132  213               .ADDR   BIT15!LKCMD
002264  105  113               .BYTE   'E,'K   ;*EXAMINE KONSOLE*
002266  053  013               .ADDR   EKCMD
002270  114  106               .BYTE   'L,'F   ;LOAD DIAG FUNCTION
002272  242  214               .ADDR   BIT15!LFCMD
002274  104  106               .BYTE   'D,'F   ;DEPOSIT INTO DIAG FUNCTION
002276  251  214               .ADDR   BIT15!DFCMD
002300  115  113               .BYTE   'M,'K   ;*MARK MICRO-CODE*
002302  012  220               .ADDR   BIT15!MKCMD
```

```
1514  002304  125  115              .BYTE   'U,'M   ;*UNMARK MICRO-CODE*
1515  002306  005  220              .ADDR   BIT15!UMCMD
1516  002310  120  105              .BYTE   'P,'E   ;*PARITY ENABLE*
1517  002312  371  016              .ADDR   PECMD
1518  002314  103  105              .BYTE   'C,'E   ;*CACHE ENABLE*
1519  002316  336  013              .ADDR   CECMD
1520  002320  124  105              .BYTE   'T,'E   ;*1 MSEC CLOCK ENABLE*
1521  002322  012  014              .ADDR   TECMD
1522  002324  124  120              .BYTE   'T,'P   ;*TRAP ENABLE*
1523  002326  104  014              .ADDR   TPCMD
1524
1525  002330  123  124              .BYTE   'S,'T   ;*START*
1526  002332  102  217              .ADDR   BIT15!STCMD
1527  002334  110  101              .BYTE   'H,'A   ;*HALT*
1528  002336  207  017              .ADDR   HACMD
1529  002340  103  117              .BYTE   'C,'O   ;*CONTINUE*
1530  002342  164  017              .ADDR   COCMD
1531  002344  123  111              .BYTE   'S,'I   ;*SINGLE INSTRUCT*
1532  002346  173  014              .ADDR   SICMD
1533  002350  123  115              .BYTE   'S,'M   ;*START MICRO-CODE*
1534  002352  201  016              .ADDR   SMCMD
1535
1536  002354  115  122              .BYTE   'M,'R   ;*MASTER RESET*
1537  002356  225  005              .ADDR   MRCMD
1538  002360  103  123              .BYTE   'C,'S   ;*START CPU CLOCK*
1539  002362  214  014              .ADDR   CSCMD
1540  002364  103  110              .BYTE   'C,'H   ;*HALT CPU CLOCK*
1541  002366  227  014              .ADDR   CHCMD
1542  002370  103  120              .BYTE   'C,'P   ;*PULSE CPU CLOCK*
1543  002372  164  013              .ADDR   CPCMD
1544  002374  105  116              .BYTE   'E,'N   ;*EXAMINE NEXT*
1545  002376  173  012              .ADDR   ENCMD
```

```
1547  002400  105  115              .BYTE   'E,'M   ;*EXAMINE MEMORY*
1548  002402  024  012              .ADDR   EMCMD
1549  002404  105  111              .BYTE   'E,'I   ;*EXAMINE I/O*
1550  002406  347  012              .ADDR   EICMD
1551  002410  105  103              .BYTE   'E,'C   ;*EXAMINE CRAM*
1552  002412  260  015              .ADDR   ECCMD
1553  002414  105  102              .BYTE   'E,'B   ;*EXAMINE BUS*
1554  002416  113  011              .ADDR   EBCMD
1555  002420  105  112              .BYTE   'E,'J   ;*EXAMINE CURRENT CRAM INFO*
1556  002422  103  015              .ADDR   EJCMD
1557
1558  002424  124  122              .BYTE   'T,'R   ;*TRACE*
1559  002426  172  015              .ADDR   TRCMD
1560  002430  122  103              .BYTE   'R,'C   ;*FUNCTION READ CRAM CONTROL REG*
1561  002432  017  015              .ADDR   RCCMD
1562  002434  132  115              .BYTE   'Z,'M   ;*ZERO KS10 MOS MEMORY*
1563  002436  064  020              .ADDR   ZMCMD
1564  002440  120  115              .BYTE   'P,'M   ;*PULSE MICRO-CODE..*
1565  002442  247  015              .ADDR   PMCMD
1566  002444  102  124              .BYTE   'B,'T   ;*BOOT SYS*
1567  002446  244  021              .ADDR   BTCMD
1568
1569  002450  102  103              .BYTE   'B,'C   ;*BOOT CHECK*
1570  002452  360  025              .ADDR   BCCMD
1571  002454  114  102              .BYTE   'L,'B   ;*LOAD BOOT*
1572  002456  274  021              .ADDR   LBCMD
1573  002460  105  130              .BYTE   'E,'X   ;*EXECUTE*
1574  002462  037  217              .ADDR   BIT15!EXCMD
1575                         ;%%%    .BYTE   'L,'T   ;*LAMP TEST*
1576                         ;%%%    .ADDR   LTCMD
1577  002464  113  114              .BYTE   'K,'L   ;*KLINIK*
1578  002466  245  017              .ADDR   KLCMD
```

```
1580  002470  105  122              .BYTE   'E,'R   ;*EXAMINE REGISTER*
1581  002472  215  013              .ADDR   ERCMD
1582  002474  114  122              .BYTE   'L,'R   ;*LOAD REGISTER*
1583  002476  300  013              .ADDR   LRCMD
1584  002500  104  122              .BYTE   'D,'R   ;*DEPOSIT REGISTER*
1585  002502  310  013              .ADDR   DRCMD
1586  002504  115  124              .BYTE   'M,'T   ;*MAGTAPE BOOT*
1587  002506  373  021              .ADDR   MTCMD
1588  002510  104  123              .BYTE   'D,'S   ;*DISK SELECT*
1589  002512  324  020              .ADDR   DSCMD
1590  002514  115  123              .BYTE   'M,'S   ;*MAGTAPE SELECT*
1591  002516  002  021              .ADDR   MSCMD
1592  002520  123  110              .BYTE   'S,'H   ;*SHUTDOWN*
1593  002522  215  017              .ADDR   SHCMD
1594  002524  115  102              .BYTE   'M,'B   ;*MAGTAPE BOOTSTRAP*
1595  002526  207  022              .ADDR   MBCMD
1596  002530  120  127              .BYTE   'P,'W   ;*PASSWORD*
1597  002532  333  017              .ADDR   PWCMD
1598  002534  124  124              .BYTE   'T,'T   ;*KLINIK LINE TO TTY*
1599  002536  307  017              .ADDR   TTCMD
1600  002540  126  124              .BYTE   'V,'T   ;*VERIFY AGAINST TAPE*
1601  002542  014  024              .ADDR   VTCMD
```

```
 1603  002544  126  104              .BYTE   'V,'D   ;*VERIFY AGAINST DISK*
 1604  002546  001  024              .ADDR   VDCMD
 1605  002550  130  061              .BYTE   'X,'1   ;DUMMY
 1606  002552  054  040              .ADDR   RAMX1
 1607  002554  106  111              .BYTE   'F,'I   ;*FILE*
 1608  002556  303  223              .ADDR   BIT15!FICMD
 1609  002560  102  062              .BYTE   'B,'2   ;**TEMP BOOTCHECK 2**
 1610  002562  371  023              .ADDR   B2CMD
 1611  002564  115  115              .BYTE   'M,'M   ;MANUFACTURING MODE
 1612  002566  147  014              .ADDR   MMCMD
 1613  002570  123  103              .BYTE   'S,'C   ;SOFT CRAM ERROR RECOVERY 'ON/OFF' SWITCH
 1614  002572  046  014              .ADDR   SCCMD
 1615  002574  000                  .BYTE   0       ;END LIST MARKER
```

```
                        .SBTTL  CHECKSUMS AS COMPUTED BY SPECIAL 10-BASED PROGRAM

        001                     .IF DF,PASS1

                        CHECKS: .ADDR   0       ;PSUEDO BYTES FOR RAM NUMBER 1
                                .ADDR   0       ;PSUEDO BYTES FOR RAM NUMBER 2
                                .ADDR   0       ;PSUEDO BYTES FOR RAM NUMBER 3
                                .ADDR   0       ;PSUEDO BYTES FOR RAM NUMBER 4
        000                     .ENDC

        001                     .IF DF,PASS2
 002575  052  240       CHECKS: .ADDR   CHKSM0  ;CHECKSUM FOR RAM NUMBER 1
 002577  217  301               .ADDR   CHKSM1  ;CHECKSUM FOR RAM NUMBER 2
 002601  157  340               .ADDR   CHKSM2  ;CHECKSUM FOR RAM NUMBER 3
 002603  216  006               .ADDR   CHKSM3  ;CHECKSUM FOR RAM NUMBER 4
        000                     .ENDC
                        .TITLE VER 6.2 KS10 CONSOLE PROGRAM
 002605  134  113  123  INIMS: .ASCIZ /\KS10 CSL.V6.2\/ ;POWER UP MESSAGE AND IDENTIFIER
```

```
 1637                           .SBTTL  *** "MR" CMD ***
 1638                           ;THIS CODE PERFORMS THE "MASTER RESET" CONSOLE FUNCTION
 1639  002625                   MRCMD:
 1640  002625  257                     XRA     A       ;SET ACCUM=0
 1641  002626  323  212                OUT     CPUCTL  ;SET 0'S TO "RUN,EXECUTE,CONT"
 1642  002630  315  227  014           CALL    CHCMD   ;AND INSURE CPU HAS STOPPED...
 1643
 1644                           ;ISSUE SM10 BUS RESET
 1645  002633  076  005        MRINT:  MVI     A,5     ;BITS FOR "DP RESET", & "CRAM RESET"
 1646  002635  323  204                OUT     CRMCTL  ;***** I/O WRT 204/5 *****
 1647
 1648  002637  076  200                MVI     A,^B10000000 ;BIT7 FOR RESET
 1649  002641  323  100                OUT     RESET   ;ISSUE RESET,SET CONSOLE MODE
 1650
 1651  002643  315  356  016           CALL    SMFINI  ;GET CURRENT PARITY SETTINGS & SET IN KS
 1652
 1653  002646  072  355  040           LDA     TRAPEN  ;BIT FOR "CLR TEN INT" SHOULD BE LOW
 1654  002651  323  205                OUT     DIAG    ;***** I/O WRT 205/XX *****
 1655
 1656  002653  006  000                MVI     B,0     ;WILL SET NO BITS IN THE STATE WORD
 1657  002655  315  326  032           CALL    STATEM  ;SET THE STATE
 1658  002660  012                     .BYTE   ^D12    ;OFF THE STUFF WE DONT WANT
 1659  002661                         ENDCMD          ;AND OUT
  (1)  002661  311                     RET             ;RETURN TO CALLER
```

```
 1661                           .SBTTL ****INTERRUPT HANDLER****
 1662                           ;HERE ON INTERRRUPTS..REGS ALREADY SAVED AT 'RST' BLOCK
 1663  002662  041  206  006   INTRP:  LXI     H,ENDIN ;PUSH OUR FAVORITE EXIT ADDRESS ON THE STACK
 1664  002665  345                   PUSH    H       ; AND WE CAN DO 'RET''S TO LEAVE ROUTINE
 1665  002666  333  201               IN      CTYCTL  ;GET CTY TTY STATUS
 1666  002670  107                   MOV     B,A     ;SAVE IT IN THE 'B' REG FOR A COUPLE INSTRS
 1667  002671  333  203               IN      REMCTL  ;NOW FETCH UP THE REMOTE STATUS
 1668  002673  260                   ORA     B       ;AND THROW BOTH STATUS'S TOGETHER
 1669  002674  346  070               ANI     ^O70    ;ANY ERR BITS SET??
 1670  002676  302  067  010         JNZ     TTERR   ;GO TELL ERR IF YES
 1671
 1672                           ;FALL THROUGH IF NO ONE IN ERROR, NOW SEE WHO THE CHARACTER IS FROM
 1673  002701  170                   MOV     A,B     ;COPY CTY STATUS INTO ACCUM
 1674  002702  346  002               ANI     2       ;IS A CHARACTER IN THE CTY UART??
 1675  002704  302  335  005         JNZ     INTCH   ;JUMP IF YES, FIND CHARACTER IN CTY UART.
 1676
 1677                           ;FALL HERE IF IT WAS A KLINIK CHAR
 1678  002707  333  202               IN      REMDAT  ;FETCH OUT THE CHARACTER
 1679  002711  346  177               ANI     ^O177   ;OFF THE PARITY BIT
 1680  002713  107                   MOV     B,A     ;MAKE SECOND COPY OF CHARACTER IN 'B'
 1681
 1682  002714  376  031               CPI     CNTLY   ;BEFORE DISPATCHING, SEE IF THIS IS 'CONTROL-Y'
 1683  002716  302  331  005         JNZ     KL.DSP  ;IF NOT, DO EVERYTHING AS PER NORMAL
 1684
 1685                           ;AHA, IT WAS A CONTROL-Y..NOW SEE IF IN MM MODE
 1686  002721  072  251  040         LDA     MMFLG   ;GET FLAG
 1687  002724  247                   ANA     A       ;SET 8080 CONDITION CODES
 1688  002725  302  165  032         JNZ     MMERR1  ;IF YES, MM MODE, THEN GO ABORT WHAT EVER YOU ARE DOING
 1689  002730  170                   MOV     A,B     ;NOW REPLACE THE CHAR WE JUST BOMBED
 1690
 1691                           ;ELSE FALL THROUGH
 1692  002731  052  346  040   KL.DSP: LHLD    MODDIS  ;GET CURRENT KLINIK MODE DISPATCH
 1693  002734  351                   PCHL            ;DISPATCH TO DO THE RIGHT THING
```

```
 1695                           ;CHECK IF WE ARE IN USER MODE
 1696  002735  333  200         INTCH:  IN      CTYDAT  ;INPUT CHARACTER
 1697  002737  346  177                 ANI     ^O177   ;STRIP BIT 8
 1698  002741  107                       MOV     B,A     ;SAVE CHAR FOR 2 INSTRUCTIONS
 1699
 1700                           ;SEE IF THIS IS MANUFACTURING MODE BEFORE WE CONTINUE
 1701  002742  072  247  040             LDA     CSLMODE ;GRAB CURRENT CSL MODE
 1702  002745  346  020                 ANI     .MODE4  ;AND SEE IF ITS MODE 4
 1703  002747  312  373  005             JZ      CMNBUF  ;IF NO, CTY INPUT NORMAL
 1704
 1705                           ;PLACE YOU GO IF MANUFACTURING MODE,, CTY CHARS ARE JUST SENT TO KLINIK
 1706  002752  170                       MOV     A,B     ;GRAB CHARACTER TO BE SENT TO KLINIK
 1707  002753  315  123  004             CALL    KCHR0   ;ONLY ECHO CTY STUFF AGAINST THE KLINIK LINE
 1708
 1709                           ;NOW SEE IF THAT WAS A MODE CHANGE CHAR WE JUST SENT DOWN THE LINE
 1710  002756  376  031                 CPI     CNTLY   ;IS IT "CONTROL-Y"
 1711  002760  300                       RNZ             ;IF WAS NOT, SIMPLY GET OUT
 1712
 1713  002761                           CLRB    KLNKSW  ;FORCE RE-EXAMINE OF THINGS
 1714  002763                           CLRB    MMFLG   ;TURN OFF MANUFACTURING MODE
 1715  002765  315  101  034             CALL    SETM2   ;FORCE KLINIK LINE IMMEDIATELY INTO MODE 2
 1716  002770  303  255  001             JMP     REINI
```

```
 1718                           ;ALSO STANDARD COMMON ENTRY POINT WHEN KLINIK LINE PARALLELS THE CTY
 1719  002773                   MODE3:
 1720  002773  072 156 040     CMNBUF: LDA     USRMD   ;GET USER MODE FLAG
 1721  002776  247                     ANA     A       ;IS IT SET??
 1722  002777  170                     MOV     A,B     ;COPY CHAR INTO ACCUM(GET HERE FOR KLINIK OR CTY)
 1723  003000  302 277 006             JNZ     USER    ;JUMP IF IN USER MODE...
 1724
 1725                           ;FALL THRU TO HERE IF NOT USER MODE AND WE NEED DO SOMETHING WITH CHAR
 1726  003003  376 017                 CPI     CNTLO   ;CONTROL O???
 1727  003005  302 025 006             JNZ     SKP2    ;JMP IF NO
 1728
 1729                           ;ELSE FALL INTO CONTROL-O CODE..STOP THE PRINTER
 1730  003010                         PCHAR   UPARR
  (1)  003010  317                     RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (1)  003011  136                     ,BYTE   UPARR   ;CHAR TO PRINT
 1731  003012                         PCHAR   OCHR
  (1)  003012  317                     RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (1)  003013  117                     ,BYTE   OCHR    ;CHAR TO PRINT
 1732  003014  072 120 040             LDA     NOPNT   ;GET CURRENT STATE OF "NO PRINT"
 1733  003017  306 200                 ADI     ^O200   ;ZAP PRINT FLAG
 1734  003021  062 120 040             STA     NOPNT   ;PUT IT BACK
 1735  003024  257                     XRA     A       ;ZAP CHAR SO WE CAN EARLY EXIT
 1736
 1737  003025  376 023           SKP2: CPI     CNTLS   ;IS IT CONTROL-S
 1738  003027  314 141 007             CZ      CNTS    ;CALL IF YES
 1739
 1740  003032  376 021                 CPI     CNTLQ   ;IS IT CONTROL-Q
 1741  003034  302 042 006             JNZ     SKP6    ;JMP IF NO
 1742
 1743                           ;FALL TO HERE IF YES, ZAPP CNTL-Q FLAG
 1744  003037                         CLRB    STPPD   ;ZAP!!
 1745  003041  257                     XRA     A       ;CLEAR ACCUM
```

```
 1747  003042  062  157  040  SKP6:  STA    RPEND  ;ANY OTHER CHARS MEAN END REPEAT LOOP
 1748  003045  376  032               CPI    CNTLZ  ;CONTROL Z??
 1749  003047  312  365  006          JZ     CNTZ   ;JMP IF YES
 1750
 1751  003052  376  025               CPI    CNTLU  ;CONTROL-U??
 1752  003054  302  071  006          JNZ    SKP8   ;JMP IF NO
 1753
 1754                      ;FALL TO HERE TO DO THE CONTROL-U CODE
 1755  003057                         PCHAR  UPARR
  (1)  003057  317                    RST    1      ;GO PRINT CHAR IN TRAILING BYTE
  (1)  003060  136                    .BYTE  UPARR  ;CHAR TO PRINT
 1756  003061                         PCHAR  UCHR
  (1)  003061  317                    RST    1      ;GO PRINT CHAR IN TRAILING BYTE
  (1)  003062  125                    .BYTE  UCHR   ;CHAR TO PRINT
 1757  003063                         PCRLF         ;AND A CR-LF TO GIVE CLEAN LINE
  (2)  003063  347                    RST    4
  (2)  003064  002                    .BYTE  2
 1758  003065  315  160  010          CALL   BFRST  ;CLEAR INPUT BUFFER
 1759  003070  257                    XRA    A      ;AND SET ACCUM FOR EARLY OUT
 1760
 1761  003071  376  003       SKP8:  CPI    CNTLC  ;CONTROL-C??
 1762  003073  312  146  010          JZ     CNTC   ;JMP IF YES
 1763
 1764  003076  376  000               CPI    Q.OUT  ;SEE IF MUST TAKE A QUICK OUT
 1765  003100  310                    RZ            ;LEAVE IF YES
 1766
 1767  003101  376  054               CPI    COMMA  ;IS IT A COMMA?
 1768  003103  302  112  006          JNZ    M11    ;IF NOT COMMA, AVOID THIS NEXT COUPLE INSTRUCTIONS
 1769
 1770                      ;FALL TO HERE IF WAS A COMMA
 1771  003106  041  225  040          LXI    H,CMCNT ;POINT TO THE COMMA COUNTER
 1772  003111  064                    INR    M      ;UPDATE,, AND CONTINUE
```

```
 1774                          ;NOT SPECIAL CHAR.,PROCESS NORMAL
 1775  003112  376  034        M11:  CPI     CNBCK   ;CONTROL BACKSLASH SHOULD LOOK LIKE CRLF
 1776  003114  314  225  006         CZ      EOMRK   ;CALL IF YES
 1777
 1778  003117  376  015              CPI     CRCHR   ;CARRIAGE RET??
 1779  003121  314  225  006         CZ      EOMRK   ;CALL IF YES
 1780
 1781  003124  376  012              CPI     LFCHR   ;LINE FEED??
 1782  003126  314  225  006         CZ      EOMRK   ;CALL IF YES
 1783
 1784  003131  052  215  040         LHLD    BUF.    ;POINTER TO FIRST FREE BUFFER PLACE
 1785  003134  376  177              CPI     RBOUT   ;RUB-OUT CHAR??
 1786  003136  312  244  006         JZ      RUB     ;JMP IF YES
 1787
```

```
                        ;OTHERWISE ITS A REGULAR CHAR..
                        ;THIS IS DUMB CODE FOR FIRST GO AROUND
                        ;TYPE-AHEAD WONT WORK

003141  315  214  006          CALL    UP.LO   ;CONVERT SO PROGRAM INTERNAL ONLY SEES UPPER CASE

003144  167                    MOV     M,A     ;CHAR INTO BUFFER SPACE
003145  043                    INX     H       ;UPDATE PNTR
003146  042  215  040          SHLD    BUF.    ;AND REPLACE PNTR

003151  107                    MOV     B,A     ;SAVE THE CHAR JUST TYPED
003152  326  040               SUI     ^040    ;CHECK IF ITS A PRINTING CHAR
003154  372  171  006          JM      NOECH   ;IF IT IS NONE-PNT..GO NO ECHO

003157  170                    MOV     A,B     ;GET CHAR BACK
003160  326  176               SUI     ^0176   ;IS IT TOO HI TO BE PRINTING CHAR??
003162  362  171  006          JP      NOECH   ;IF YES, GO NO ECHO

003165  170                    MOV     A,B     ;GET CHAR BACK AGAIN
003166  315  010  004          CALL    PCHR    ;NOW GO ECHO IT.....

003171  072  161  040  NOECH:  LDA     BFCNT   ;GET CHAR COUNT
003174  074                    INR     A       ;BUMP UP
003175  376  120               CPI     80.     ;TOO MANY??
003177  312  001  004          JZ      BFOVR   ;JMP BUFFER OVERFLOW IF YES
003202  062  161  040          STA     BFCNT   ;REPLACE COUNT
003205  341                    POP     H       ;CLEAR STACK OF THE PSEUDO RETURN FIRST
                        ;FALL INTO END INTERRRUPT CODE IF CHAR COUNT OK
003206  341            ENDIN:  POP     H       ;RESTORE REGS
003207  321                    POP     D
003210  301                    POP     B
003211  361                    POP     PSW
003212  373                    EI              ;INTERRUPTS BACK ON
003213  311                    RET             ;AND OUT

                        ;AND ACCEPT LOWER CASE AS REQUESTED
003214  376  141        UP.LO:  CPI     ^0141   ;LOW CASE "A" OR BETTER??
003216  370                     RM             ;IF MINUS, NOT LOW CASE, CONTINUE
003217  376  173               CPI     ^0173   ;LOW CASE "Z" OR WORSE??
003221  360                     RP             ;IF POS, OR ZERO, ITS NOT LOW CASE RANGE

                        ;FALL INTO HERE IF IT WAS LOWER CASE....
003222  326  040               SUI     ^040    ;MAKE IT UPPER FOR ALL
003224  311                    RET             ;AND OUT
```

```
 1835                           ;CODE FOR END-OF-LINE CHAR TYPED IN
 1836  003225                   EOMRK:  PCRLF           ;GIVE CR-LF
  (2)  003225  347                      RST     4
  (2)  003226  002                      .BYTE   2
 1837
 1838  003227  072  225  040            LDA     CMCNT   ;GET COUNT OF COMMAS
 1839  003232  074                      INR     A       ;UP BY ONE, FOR THE CR-LF
 1840  003233  062  204  040            STA     EOL     ;AND SET EOL MARKER
 1841  003236  257                      XRA     A       ;CLEAR ACCUM
 1842  003237  062  225  040            STA     CMCNT   ;SET LOCATION
 1843  003242  057                      CMA             ;SET ACCUM = -1
 1844  003243  311                      RET             ;RETURN
 1845
 1846
 1847                           ;CODE FOR A RUB-OUT
 1848  003244  072  161  040   RUB:     LDA     BFCNT   ;GET CURRENT CHAR COUNT
 1849  003247  247                      ANA     A       ;IS IT 0??
 1850  003250  310                      RZ              ;GOOD, NOTHING TO DELETE
 1851  003251  075                      DCR     A       ;DECREMENT OTHERWISE
 1852  003252  062  161  040            STA     BFCNT   ;AND PUT IT BACK
 1853
 1854  003255  053                      DCX     H       ;AND BACK UP THE BUFFER PNTR
 1855  003256  042  215  040            SHLD    BUF.    ;PUT IT BACK
 1856  003261                           PSLASH          ;TYPE SLASH AS RUBOUT INDICATOR
  (2)  003261  317                      RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (2)  003262  057                      .BYTE   SLASH   ;CHAR TO PRINT
 1857  003263  176                      MOV     A,M     ;GET CURRENT CHAR IN BUFFER
 1858  003264  315  010  004            CALL    PCHR    ;ECHO WHAT WAS RUBBED OUT
 1859  003267  376  054                CPI     COMMA   ;OH WAIT, WAS THAT A COMMA??
 1860  003271  300                      RNZ             ;JMP IF NO, TAKE A NORMAL OUT
 1861                           ;FALL THRU IF WAS A COMMA
 1862  003272  041  225  040            LXI     H,CMCNT ;GET COMMA COUNT
 1863  003275  065                      DCR     M       ;DECREMENT
 1864  003276  311                      RET             ;AN EXIT THIS PLACE
```

```
 1866                          ;HERE IF USER MODE FLAG IS SET..
 1867  003277  376  034       USER:   CPI     CNBCK   ;IS IT '^\'??
 1868  003301  302  160  007          JNZ     TENCHR  ;IF NOT,THEN ITS A CHAR FOR THE KS-10
 1869
 1870                          ;BEFORE WE LEAVE USER MODE, WE MUST CHECK THE CONSOLE ENABLE SWITCH
 1871  003304  333  301       END.USR: IN BOOTSW      ;***** I/O RD 301 ***** IS CONSOLE LOCKED UP??
 1872
 1873                          ;BIT 2 IS 'LO' IF CONSOLE ENABLE IS TRUE. IF BIT 2 IS 'HI', WE ARE DISABLED
 1874                          ;DUE TO THE HARDWARE INVERSION OF SIGNAL LEVELS
 1875  003306  346  004               ANI     ^D4     ;CHECK BIT 2
 1876  003310  300                    RNZ             ;IF HI, WE ARE DISABLED AND WILL IGNORE
 1877
 1878                          ;IF YES, USER MODE MUST BE CLEARED
 1879  003311                         CLRB    NOPNT   ;CLR NO PRINT FLAG IN CASE WE WERE IN 'INTERNAL MODE'
 1880  003313  315  346  006          CALL    CLRUSE  ;EXIT FROM USER MODE
 1881  003316                         PLINE   RDYMS   ;'ENABLED'
  (1)  003316  337                    RST     3       ;PRINT LINE OF CHARS
  (1)  003317  335  006               .ADDR   RDYMS   ;BUFF TO PRINT
 1882  003321  041  255  001          LXI     H,REINI ;SET UP AN EXIT ADDRESS
 1883  003324  321              IOUT:  POP     D       ;CLEAR THE PSEUDO RETURN TO 'ENDIN' FROM STACK
 1884  003325  321                    POP     D       ;CLEAR ORIGINAL SAVED 'H,L' OFF STACK
 1885  003326  321                    POP     D
 1886  003327  301                    POP     B
 1887  003330  361                    POP     PSW
 1888  003331  063                    INX     SP      ;NOW GET OLD RETURN ADDR OFF STACK
 1889  003332  063                    INX     SP
 1890
 1891  003333  373                    EI              ;ENABLE INTS..
 1892  003334  351                    PCHL            ;AND GO TO PROMPT
 1893  003335  105  116  101  RDYMS:  .ASCIZ /ENABLED\/
 1894
 1895  003346                 CLRUSE: CLRB    USRMD   ;AND CLEAR THE USER MODE FLAG
 1896  003350  072  251  040          LDA     MMFLG   ;BEFORE DROPPING USER, SEE IF IN MM MODE
 1897  003353  247                    ANA     A       ;SET 8080 FLAGS
 1898  003354  310                    RZ              ;IF NOT MM MODE, OK TO GET OUT
 1899
 1900                          ;IF WAS SET, MUST DROP BACK TO MODE 4
 1901  003355  315  050  034          CALL    SETM4
 1902  003360  315  117  004          CALL    KCHR    ;NON-PRINTING CHAR, ALSO TELLS HOST TO SWITCH MODES
 1903  003363  034                    .BYTE   CNBCK   ;'CONTROL-BACKSLASH IS THE MAGIC CHAR'
 1904  003364  311                    RET             ;JUST LEAVE
```

```
 1906                           ;CONTROL-Z CODE...ENTER USER MODE
 1907  003365  315  017  007   CNTZ:  CALL    SETUSE  ;SET THE USER MODE
 1908  003370  315  160  010          CALL    BFRST   ;BUFFER RESET..RE-INIT TTY INPUT BUFFER
 1909  003373                       CLRB    KLNKSW  ;FORCE LIGHTS TO GET FIXED AFTER ENTER USER
 1910
 1911  003375                       PLINE   U       ;PRINT 'USER MODE'
  (1)  003375  337                 RST     3       ;PRINT LINE OF CHARS
  (1)  003376  006  007            .ADDR   U       ;BUFF TO PRINT
 1912  003400  041  335  001          LXI     H,NULLJ ;LOAD 'H,L' WITH A PLACE TO GO
 1913  003403  303  324  006          JMP     IOUT    ;AND GET OUT
 1914  003406  125  123  122   U:     .ASCIZ /USR MOD\/
```

```
 1916  003417                  SETUSE:
 1917                          ;HERE IS THE DEPOSIT WORD 31 CODE..
 1918  003417                  WRD31:  INTON           ;DONT PRINT THIS STUFF
  (1)  003417  327                       RST     2       ;GO SET INTERNAL MODE
 1919  003420                          EXAM    31      ;MUST SAVE CURRENT STATE OF KEEP ALIVE & RELOAD BITS
  (1)  003420  315  216  030             CALL    EXAMSH  ;AND DO EXAM ASSUMING SHORT ADDRESS
  (1)  003423  031  000                  ,ADDR   31      ;ADDR TO BE ZAPPED PASSED AS TRAILING ARG
 1920  003425                          INTOFF          ;ITS OK NOW
  (1)  003425  367                       RST     6       ;GO EXIT FROM INTERNAL MODE
 1921  003426  072  255  040             LDA     GOCODE  ;BYTE 28-35 GETS THE REASON FOR RELOAD
 1922  003431  041  047  040             LXI     H,DMDAT ;MAKE H,L POINT TO THE DESIRED BUFFER
 1923  003434  167                       MOV     M,A     ;SET THE GOCODE BITS INTO THE BYTE 'DMDAT'
 1924
 1925  003435  072  355  040             LDA     TRAPEN  ;NOW GRAB THE TRAP BIT
 1926  003440  007                       RLC             ;AND SHIFT IT TO THE APPROPRIATE POSITION(20 TO 40)
 1927  003441  007                       RLC             ;(40 TO 100)
 1928  003442  007                       RLC             ;(100 TO 200)
 1929  003443  107                       MOV     B,A     ;SAVE IT IN B FOR A WHILE
 1930
 1931  003444  072  251  040             LDA     MMFLG   ;GET 'MAINTENANCE MODE' FLAG
 1932  003447  247                       ANA     A       ;SET 8080 FLAGS
 1933  003450  365                       PUSH    PSW     ;SAVE THE STATE OF THE FLAGS FOR LATER USE
 1934  003451  312  056  007             JZ      WRD.PR  ;IF NO MM MODE, DONT SET A BIT
 1935
 1936                          ;WAS MM MODE, MUST SET THE BIT
 1937  003454  076  100                   MVI     A,^O100 ;A BIT FOR MM MODE
 1938  003456  260             WRD.PR: ORA     B       ;THROW TOGETHER WITH THE TRAP BIT
 1939  003457  043                       INX     H       ;PUT INTO THE DEPOSIT BUFFER
 1940  003460  043                       INX     H
 1941  003461  167                       MOV     M,A     ;THIS IS LOC 'DMDAT+2'
 1942
 1943  003462  072  354  040             LDA     PARBT   ;NOW FOR THE SELECTION OF PARITY BITS
 1944  003465  017                       RRC             ;RIGHT ONCE TO FREE UP 200 WEIGHT
 1945  003466  107                       MOV     B,A     ;SAVE IN B REG
 1946  003467  072  247  040             LDA     CSLMODE ;GET CURRENT KLINIK MODE
 1947  003472  346  014                   ANI     .MODE2!.MODE3 ;IF EITHER OF THESE MODES, MUST SET THE BIT
 1948  003474  312  101  007             JZ      WRD.DP  ;JUMP IF NOT THOSE BITS
 1949
 1950
 1951                          ;HERE IF ONE OF THOSE MODES WAS SET
 1952  003477  076  100                   MVI     A,^O100 ;A BIT TO SET
 1953
 1954  003501  260             WRD.DP: ORA     B       ;THROW THIS BIT WITH THE OTHERS
 1955  003502  017                       RRC             ;FINAL JUSTIFICATION
 1956  003503  107                       MOV     B,A     ;NOW SAVE THIS GOOD STUFF IN 'B'
 1957  003504  072  015  040             LDA     EMBUF+3 ;GET THE BYTE THAT HAS CURRENT 'KA' BIT
 1958  003507  346  300                   ANI     ^O300   ;OFF EVERYTHING ELSE
 1959  003511  260                       ORA     B       ;NOW THROW WHOLE MESS TOGETHER AGAIN
 1960
 1961  003512  043                       INX     H       ;BUMP POINTER TO 'DMDAT+3'
 1962  003513  167                       MOV     M,A     ;AND PUT DATA INTO RAM
 1963  003514                          DEPOS   31      ;PUT INTO MOS MEMORY AT LOC 31
  (1)  003514  247                       ANA     A       ;CLR 'C-BIT' FOR USE BY COMMON CODE
  (1)  003515  315  217  030             CALL    DEPSHT  ;AND DO THE DEPOSIT ASSUMING SHORT ADDR
```

```
 (1) 003520 031 000            .ADDR   31      ;ADDR TO BE ZAPPED PASSED AS TRAILING ARG
1964 003522                    CLRB    GOCODE  ;CLEAR THE RELOAD CODE
1965
1966 003524 361                POP     PSW     ;THIS WORD HAS FLAGS SET FROM BEFORE WHEN WE TESTED 'MM'
1967 003525 076 377            MVI     A,-1    ;FLAGS WONT CHANGE WHILE WE SET THE USER MODE FLAG
1968 003527 062 156 040        STA     USRMD   ;SET USER MODE... NOW DO SOMETHING BASED ON THE FLAGS
1969 003532 310                RZ              ;IF NOT SET, A SIMPLE OUT
1970
1971                       ;HERE IF SET, WE MUST SEND AN "ACK" DOWN KLINIK LINE BEFORE ANYTHING ELSE
1972 003533 315 205 035        CALL    ACK     ;'ACK' DOWN THE KLINIK
1973 003536 303 101 034        JMP     SETM2   ;ALSO SET MODE 2 AND USE HIS 'RET' TO RETURN
1974
1975                       ;TYPED 'CONTROL-S' TO STOP CONSOLE OUTPUT
1976 003541 041 162 040    CNTS:   LXI     H,STPPD ;POINTER TO STOPPED FLAG
1977 003544 176                    MOV     A,M     ;GET THE FLAG
1978 003545 057                    CMA             ;SET .EQ. 0 IF WAS ALREADY SET
1979 003546 247                    ANA     A       ;NOW SET FLAGS, 'CAUSE CMA DOESN'T
1980 003547 310                    RZ              ;IF .EQ. 0 NOW, MERELY LEAVE..ALREADY SET
1981
1982 003550 167                    MOV     M,A     ;AND SET THE FLAG .EQ. -1 IF HERE
1983 003551 373                    EI              ;LET THE CNTL-Q THRU
1984
1985 003552 176            CNTSL:  MOV     A,M     ;GET FLAG STATUS
1986 003553 247                    ANA     A       ;IS IT SET???
1987 003554 310                    RZ              ;IF NOT,THEN TIME TO QUIT
1988 003555 303 152 007            JMP     CNTSL   ;STAY IN LOOP IF FLAG STILL SET
1989
```

```
 1991                          .SBTTL  **8080 TO KS10 CHARACTER SERVICE**
 1992  003560  062  035  040   TENCHR: STA     CHRBUF ;PUT CHARACTER IN A RAM BUFFER
 1993  003563  076  032                 MVI     A,^O32 ;DESIRED ADDRESS FOR DEPOSTING CHARACTER
 1994
 1995  003565  323  103                 OUT     A2835   ;WRITE ONLY RELEVANT PIECE OF THE ADDRESS
 1996  003567  257                     XRA     A       ;THEN CLR ACCUM
 1997  003570  323  105                 OUT     A2027   ;AND CLR THE REST OF THE HARDWARE ADDRESS REGISTER
 1998  003572  323  107                 OUT     A1219
 1999  003574  323  106                 OUT     W1219   ;CLEAR PIECES OF DEPOSIT DATA WHICH MUST BE ZERO
 2000  003576  323  110                 OUT     W0411
 2001  003600  323  112                 OUT     W0003
 2002
 2003  003602  076  002                 MVI     A,02    ;BIT TO SAY 'WRITE FUNCTION'
 2004  003604  323  113                 OUT     A0003   ;***** I/O WRT 113 *****
 2005
 2006                          ;THE FOLLOWING 'ADD A' WORKS BY LUCK..I.E. 2+2=4
 2007  003606  207                     ADD     A       ;BIT INTO ACCUM FOR 'COM/ADR CYCLE'
 2008  003607  323  115                 OUT     BUSARB  ;***** I/O WRT 115/4 *****
 2009
 2010  003611  072  035  040           LDA     CHRBUF  ;NOW GET THE CHARACTER WE WANT
 2011  003614  323  102                 OUT     W2835   ;PUT IT IN THE HARDWARE REGISTER
 2012  003616  076  001                 MVI     A,1     ;AND GET THE VALID BIT TO GO WITH THE CHARACTER
 2013  003620  323  104                 OUT     W2027   ;PUT IT IN THE HARDWARE REGISTER
 2014
 2015                          ;AND BY LUCK, THE ACCUM HAS JUST WHAT WE NEED FOR THE NEXT STEP
 2016  003622  323  114                 OUT     DTARB   ;***** I/O WRT 114/1 *****
 2017
 2018  003624  076  360                 MVI     A,^O360 ;BITS FOR 'CHECK NXM','CONSOLE REQ','T ENB FOR COM/ADR'
 2019                                                  ;'T ENB FOR DATA CYCLE'
 2020  003626  323  210                 OUT     BUSCTL  ;*****I/O WRT 210/360 *****
 2021
 2022                          ;DO THIS TWICE TO GUARANTEE THAT THE INTERRUPT HAPPENS
 2023  003630  076  001                 MVI     A,1     ;BIT FOR SETTING INTERRUPT TO THE KS10
 2024  003632  323  116                 OUT     INT2KS  ;SET THE INTERRUPT
 2025  003634  323  116                 OUT     INT2KS  ;SET THE INTERRUPT
 2026
 2027  003636  311                     RET             ;AND EXIT NORMALLY....
```

```
                                ;ENTER HERE WITH THE CHAR IN THE ACCUM
                                .SBTTL  **KS10 TO 8080 CHARACTER SERVICE**
                                ;ROUTINE FOR HANDLING INTERRUPT CHARACTERS FROM A RUNNING KS10.
                                ;ONLY CHARS FROM KS TO CTY IMPLEMENTED
003637                          CHRRDY: INTON           ;SET UP INTERNAL MODE
(1) 003637 327                          RST     2       ;GO SET INTERNAL MODE
003640 363                              DI              ;COMMON CODE,NOT TO BE DISTURBED

                                ;DISABLE INTERRUPTS FOR THIS OPERATION
003641 072 355 040                      LDA     TRAPEN  ;GET DEFAULT FOR THE TRAP ENABLE BITS
003644 323 205                          OUT     DIAG    ;*****I/O WRT/ TO CLR THE INTERRUPT*****

                                ;FALL TO HERE IF YES WE ARE IN KLINIK MODE 2., ITS POSSIBLE THAT THIS INTERRUPT
                                ;IS FROM THE KLINIK COMM WORD, FOR THE KLINIK LINE
003646                                  EXAM    35      ;EXAM THE KLINIK COMM WORD
(1) 003646 315 216 030                  CALL    EXAMSH  ;AND DO EXAM ASSUMING SHORT ADDRESS
(1) 003651 035 000                      .ADDR   35      ;ADDR TO BE ZAPPED PASSED AS TRAILING ARG
003653                                  INTOFF          ;KEEP THIS FLAG IN STEP
(1) 003653 367                          RST     6       ;GO EXIT FROM INTERNAL MODE
003654 072 013 040                      LDA     EMBUF+1 ;GRAB THE CONTENTS OF THE BYTE WITH THE CONTROL KEY
003657 247                              ANA     A       ;SET 8080 FLAGS
003660 312 336 007                      JZ      CTYONLY ;IF CONTROL KEY CLEAR, NOTHING FROM KLINIK, TRY CTY

003663 107                              MOV     B,A     ;SAVE THE DATA IN THE ACCUM FOR A LITTLE WHILE
003664 072 247 040                      LDA     CSLMODE ;GET CURRENT MODE..DECIDE IF MUST THROW AWAY CHARS, OR
003667 346 013                          ANI     .MODE0!.MODE1!.MODE3 ;  JUST ACT AS A NULL BIT BUCKET
003671 302 330 007                      JNZ     NULKL   ;JUMP TO A NULL ACTION IF ANY OF THESE 3 MODES

                                ;NOW HERE IF CONTROL KEY IS .NE. 0
003674 170                              MOV     A,B     ;RETRIEVE DATA
003675 376 001                          CPI     1       ;IS IT THE KEY FOR A SIMPLE CHARACTER TO BE OUTPUT??
003677 312 314 007                      JZ      KLPCHR  ;IF YES, GO PROCESS THE CHARACTER

003702 376 002                          CPI     2       ;IS IT FOR A HANGUP
003704 302 065 010                      JNZ     NOACTN  ;NOPE, IGNORE ENTIRELY

003707 315 132 034                      CALL    KILL.KLINIK  ;YUP, HANG 'EM UP
003712 373                              EI              ;IDENTICAL CODE TO LOCATION 'NOACTN', BUT THE 2 BYTES
003713 311                              RET             ; HERE ARE CHEAPER THAN A 'JMP'..I'M OUT OF PROM SPACE
```

```
 2065                           ;CODE FOR PRINTING THE DESIRED CHARACTER ON THE KLINIK LINE
 2066  003714                   KLPCHR: TSTRDY  REMCTL  ;SEE IF THE LINE IS READY FOR THE NEXT CHARACTER
  (1)  003714  333  203                 IN      REMCTL  ;READ DESIRED UART STATUS
  (1)  003716  346  001                 ANI     01      ;CHECK IF SET; ZBIT=0 IF READY; ZBIT=1 IF NOT YET READY
 2067  003720  312  314  007           JZ      KLPCHR  ;IF NOT READY YET, BETTER WAIT LONGER
 2068
 2069                           ;HERE WHEN READY
 2070  003723  072  012  040           LDA     EMBUF   ;GET THE CHARACTER
 2071  003726  323  202                 OUT     REMDAT  ;PRINT IT
 2072  003730  076  035         NULKL:  MVI     A,^035  ;NOW MUST CLEAR THE WORD, AND INTERRUPT TO SAY DONE
 2073  003732  315  017  010           CALL    TTOCOM  ;GO COMMON CODE
 2074  003735  363                     DI              ;KEEP INTERRUPTS OFF BECAUSE TTOCOM TURNED IT ON
```

```
 2076  003736                 CTYONLY: INTON          ;DON'T PRINT THIS CRUD
  (1)  003736  327                     RST     2       ;GO SET INTERNAL MODE
 2077  003737                         EXAM    33      ;GET THE COMMUNICATION WORD
  (1)  003737  315  216  030          CALL    EXAMSH  ;AND DO EXAM ASSUMING SHORT ADDRESS
  (1)  003742  033  000              .ADDR   33      ;ADDR TO BE ZAPPED PASSED AS TRAILING ARG
 2078  003744                         INTOFF          ;INTERNAL MODE OFF
  (1)  003744  367                     RST     6       ;GO EXIT FROM INTERNAL MODE
 2079  003745  072  013  040          LDA     EMBUF+1 ;GET THE INTERRUPT CODE
 2080  003750  376  001              CPI     1       ;IS INTERRUPT CODE .EQ. 1??
 2081  003752  302  065  010          JNZ     NOACTN  ;JUMP TO "NO ACTION" IF NOT, CODE OUT OF BOUNDS
 2082
 2083  003755  072  012  040          LDA     EMBUF   ;ACTUAL CHAR TO ACCUM
 2084  003760  107                    MOV     B,A     ;AND SAVE IT IN THE B REG
 2085
 2086                         ;CODE TO PRINT A CHAR PASSED FROM THE KS-10 CPU, CODE INTERRUPTS
 2087                         ;THE 10 WHEN THE CHARACTER HAS FINISHED PRINTING
 2088  003761                 CTYPCHR: TSTRDY CTYCTL  ;CHECK IS THE XMITTER READY??
  (1)  003761  333  201              IN      CTYCTL  ;READ DESIRED UART STATUS
  (1)  003763  346  001              ANI     01      ;CHECK IF SET; ZBIT=0 IF READY; ZBIT=1 IF NOT YET READY
 2089  003765  312  361  007          JZ      CTYPCHR ;LOOP UNTIL IT IS
 2090
 2091                         ;FALL THRU WHEN READY
 2092                         ;BUT BEFORE PRINTING, CHECK THE KLINIK LINE TO SEE IF IT GETS THE
 2093                         ;CHARACTER TOO
 2094  003770  072  247  040          LDA     CSLMODE ;CHECK THE KLINIK MODE
 2095  003773  376  010              CPI     .MODE3  ;IS THE KLINIK PARALLEL TO THE CTY LINE??
 2096  003775  302  012  010          JNZ     CTYOUT  ;JUMP IF NO. KLINIK DOES NOT GET THIS CHARACTER
 2097
 2098                         ;FALL HERE IF YES, KLINIK LINE GETS A PIECE OF THIS CHARACTER TOO.
 2099  004000                 KLTOO:  TSTRDY  REMCTL  ;SEE IF KLINIK LINE IS READY
  (1)  004000  333  203              IN      REMCTL  ;READ DESIRED UART STATUS
  (1)  004002  346  001              ANI     01      ;CHECK IF SET; ZBIT=0 IF READY; ZBIT=1 IF NOT YET READY
 2100  004004  312  000  010          JZ      KLTOO   ;IF NOT YET, GO BACK AND TRY AGAIN
 2101
 2102  004007  170                    MOV     A,B     ;CHAR TO ACCUM
 2103  004010  323  202              OUT     REMDAT  ;PRINT IT ON THE KLINIK LINE
 2104
 2105  004012  170                 CTYOUT: MOV     A,B     ;GET THE CHAR WE SAVED IN THE B REG.
 2106  004013  323  200              OUT     CTYDAT  ;SEND TO THE UART
```

```
 2108                        ;NOW CLEAR A DATA BUFFER FOR DEPOSITING 0'S INTO THE MOS MEMORY
 2109                        ;WE ARE USING IN LINE CODE HERE IN ORDER TO SPEED UP THE TYPE-OUT
 2110                        ;ON KS10 TO 8080 XFER'S..
 2111
 2112  004015  076  033             MVI     A,^033  ;THIS IS THE ADDRESS WE WISH TO DEPOSIT
 2113  004017  323  103     TTOCOM: OUT     A2835   ;PUT IT INTO THE HARDWARE REGISTER
 2114  004021  257                   XRA     A       ;CLEAR ACCUM, BECAUSE THE REST OF THE ADDR MUST BE ZERO
 2115  004022  323  105             OUT     A2027   ;CLR THE OTHER HARDWARE REGISTERS
 2116  004024  323  107             OUT     A1219
 2117  004026  323  102             OUT     W2835   ;AND WE WILL MAKE ALL OF THE HARDWARE DATA REGS 0
 2118  004030  323  104             OUT     W2027
 2119  004032  323  106             OUT     W1219
 2120  004034  323  110             OUT     W0411
 2121  004036  323  112             OUT     W0003
 2122
 2123  004040  076  002             MVI     A,02    ;BIT TO SAY "WRITE FUNCTION"
 2124  004042  323  113             OUT     A0003   ;***** I/O WRT 113 *****
 2125
 2126                        ;THIS 'ADD A' WORKS BY LUCK..I.E. 2+2=4
 2127  004044  207                   ADD     A       ;BIT INTO ACCUM FOR 'COM/ADR CYCLE'
 2128  004045  323  115             OUT     BUSARB  ;***** I/O WRT 115/4 *****
 2129
 2130  004047  076  001             MVI     A,1     ;BIT INTO ACCUM FOR 'DATA CYCLE'
 2131  004051  323  114             OUT     DTARB   ;***** I/O WRT 114/1 *****
 2132
 2133  004053  076  360             MVI     A,^0360 ;BITS FOR 'CHECK NXM','CONSOLE REQ','T ENB FOR COM/ADR'
 2134                                             ;'T ENB FOR DATA CYCLE'
 2135  004055  323  210             OUT     BUSCTL  ;*****I/O WRT 210/360 *****
 2136
 2137
 2138                        ;DO THIS TWICE TO GUARANTEE THE INTERRUTP GETS THRU
 2139  004057  076  001     POKE10: MVI     A,1     ;BIT FOR SETTING INTERRUPT TO THE KS10
 2140  004061  323  116             OUT     INT2KS  ;SET THE INTERRUPT
 2141  004063  323  116             OUT     INT2KS  ;SET THE INTERRUPT
 2142  004065  373           NOACTN: EI              ;OK FOR INTERRUPTS NOW
 2143  004066  311                   RET             ;AND OUT
```

```
 2145                           .SBTTL  **TTY HANDLER FOR SPECIAL CHARACTERS**
 2146                           ;WHEN HERE, 'B' REG CONTAINS THE STATUS OF THE CTY LINE
 2147  004067  170              TTERR:  MOV     A,B     ;COPY CTY STATUS TO ACCUM
 2148  004070  346  070                 ANI     ^O70    ;ANY ERRS IN THE CTY UART??
 2149  004072  302  111  010           JNZ     TTERR1  ;IF YES, GO CHECK THINGS ON THE CTY LINE
 2150
 2151                           ;HERE IF GOT KLINIK ERRORS
 2152  004075  076  025                 MVI     A,^O25  ;BEFORE JUMPING, RESET THE UART SO IT WILL WORK
 2153  004077  323  203                 OUT     REMCTL  ;I/O WRITE TO RESET THE UART
 2154
 2155  004101  072  156  040           LDA     USRMD   ;CHECK USER MODE., WILL NOT REPORT ERR IF IT IS
 2156  004104  247                     ANA     A       ;SET FLAGS
 2157  004105  302  335  005           JNZ     INTCH   ;IF WAS USER MODE, IGNORE OVERRUN AND HANDLE CHARACTER
 2158  004110  311                     RET             ;DONE INT
```

```
 2160  004111  346  050        TTERR1: ANI     ^O50    ;SEE IF OVERRRUN OR A FATAL ERROR
 2161
 2162                          ;NOW MUST CLR ERROR FROM THE UART FIRST
 2163  004113  076  025                MVI     A,^O25  ;BITS TO CLR ERROR CONDITIONS IN UART
 2164  004115  323  201                OUT     CTYCTL  ;*****I/O WRT 200/25 *****
 2165  004117  302  131  010          JNZ     TTERMS  ;NOW JUMP IF FATAL
 2166
 2167  004122  072  156  040          LDA     USRMD   ;BEFORE ISSUEING MESSAGE,IS USER MODE SET??
 2168  004125  267                    ORA     A       ;TEST USER MODE FLAG
 2169  004126  302  335  005          JNZ     INTCH   ;IF YES, USER MODE, THEN IGNORE THE ERROR
 2170
 2171                          ;NOW MUST CLR OVERRUN ERROR FROM THE UART
 2172  004131  041  247  040   TTERMS: LXI     H,CSLMODE ;GET CURRENT MODE OF KLINIK
 2173  004134  116                    MOV     C,M     ;SAVE IT IN C
 2174  004135  066  000                MVI     M,0     ;NOW CLEAR CSL MODE
 2175  004137  345                    PUSH    H       ;AND SAVE 'H,L'
 2176  004140                         PLINE   TTM     ;OUTPUT THE ERROR MESSAGE
  (1)  004140  337                    RST     3       ;PRINT LINE OF CHARS
  (1)  004141  034  037               .ADDR   TTM     ;BUFF TO PRINT
 2177  004143  341                    POP     H       ;AND RESTORE MEM POINTER
 2178  004144  161                    MOV     M,C     ;REPLACE CSL MODE AND GET OUT
 2179  004145  311                    RET             ;AND RESTART NULL LOOP
 2180
 2181  004146  061  000  044   CNTC:   LXI     SP,RAMST+^O2000 ;GUARANTEE THAT CNTRL-C WINS
 2182  004151                         PCHAR   UPARR
  (1)  004151  317                    RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (1)  004152  136                    .BYTE   UPARR   ;CHAR TO PRINT
 2183  004153                         PCHAR   CCHR
  (1)  004153  317                    RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (1)  004154  103                    .BYTE   CCHR    ;CHAR TO PRINT
 2184  004155  303  255  001          JMP     REINI   ;JUMP AWAY
```

```
 2186                          ;LOCAL SUBROUTINE TO RESET TTY INPUT BUFFER
 2187  004160  041  114  041   BFRST: LXI     H,BUFBG ;BUFFER BEGINNING
 2188  004163  042  215  040          SHLD    BUF.    ;RESET CURRENT BUFFER POINTER
 2189  004166  042  227  040          SHLD    FIRST   ;RESET CMD POINTER
 2190  004171                       CLRB    RPEND   ;CLEAR REPEAT KILLER
 2191  004173                       CLRB    CMDS..  ;SAY LINE IS DONE.. AT BEGINNING OF THINGS
 2192  004175                       CLRB    BFCNT   ;CLEAR CHAR COUNT
 2193  004177  311                  RET             ;AND RETURN
```

```
 2195                          ;THIS IS THE INITIAL MODE OF THE KLINIK LINE AFTER A POWER UP, ALSO, WHEN
 2196                          ;THE KLINIK LINE IS DISABLED, OR IN "PROTECT" MODE, BUT NO PASSWORD HAS
 2197                          ;BEEN SET BY THE OPERATOR
 2198  004200  376  007       MODE0:  CPI     BELL    ;NO ECHO IF RECEIVE BELL
 2199  004202  310                   RZ              ;BELL, SO OUT
 2200  004203                        KLINE   NOACCS  ;PRINT A MESSAGE FOR THE KLINIK LINE ONLY
  (1)  004203  315  137  004         CALL    KLINE   ;PRINT LINE OF CHARS
  (1)  004206  351  037              ,ADDR   NOACCS  ;BUFF TO PRINT
 2201  004210  303  132  034         JMP     KILL,KLINIK  ;HANG UP LINE SO REPEAT ?NA'S DONT HANG SYSTEM
 2202
 2203
 2204
 2205                          ;THIS IS KLINIK MODE 1, THIS IS THE MODE OF THE KLINIK LINE WHENEVER
 2206                          ;THE FRONT PANEL SWITCH IS IN THE PROTECT POSITION, AND WE ARE WAITING
 2207                          ;FOR THE PASSWORD TO BE ENTERED.
 2208  004213                  MODE1:  KLINE   QPW     ;BEGIN BY PRINTING "PW:"
  (1)  004213  315  137  004         CALL    KLINE   ;PRINT LINE OF CHARS
  (1)  004216  357  037              ,ADDR   QPW     ;BUFF TO PRINT
 2209  004220  041  271  040         LXI     H,KPWBUF ;INITIALIZE THE BUFFER FOR SAVING TYPED PASSWORD
 2210  004223  042  125  040         SHLD    KPWPNT  ;SAVE IN THE BUFFER POINTER
 2211
 2212  004226                        CLRB    KPWCNT  ;AND CLEAR THE PASSWORD CHARACTER COUNTER
 2213
 2214  004230  041  237  010         LXI     H,PW.WAIT ;NOW ADDITIONAL KLINIK CHARS MUST DISPATCH TO
 2215  004233  042  346  040         SHLD    MODDIS  ; THE PLACE THAT WAITS FOR A COMPLETE PASSWORD
 2216  004236  311                   RET             ;END OF INTERRUPT
```

```
                                ;THIS IS THE ENTRY POINT WHEN THE PERSON IS IN THE PROCESS OF TYPING THE
                                ;PASSWORD, WE STORE THE PASSWORD AS IT IS TYPED, THEN WHEN DONE, WE WILL
                                ;VERIFY THAT IT IS CORRECT
004237  376  015        PW.WAIT: CPI     CRCHR    ;IS IT AN END OF LINE CHARACTER
004241  312  275  010            JZ      PW.TST   ;JUMP IF YES.. TIME TO VERIFY THE PASSWORD

004244  315  214  006            CALL    UP.LO    ;GENERATE ONLY UPPER CASE FOR PROGRAM INTERNALS
004247  107                      MOV     B,A      ;SAVE THE CHAR IN B REG FOR A LITTLE BIT

                                ;IF NOT END OF LINE, JUST ADD IT TO THE BUFFER OF CHARS THAT IS THE PASSWORD
004250  072  127  040            LDA     KPWCNT   ;FIRST THINGS FIRST, SEE HOW MANY CHARS IN THE BUFFER
004253  074                      INR     A        ;UPDATE TO ACCOUNT FOR THIS ONE
004254  376  007                CPI     7        ;IS IT TOO MANY
004256  312  333  010            JZ      PW.ERR   ;JUMP IF YES, ITS A PASSWORD ERROR
004261  062  127  040            STA     KPWCNT   ;ELSE SAVE THE UPDATED COUNT AND CONTINUE

004264  052  125  040            LHLD    KPWPNT   ;GET THE BUFFER POINTER
004267  160                      MOV     M,B      ;PUT THE CHARACTER IN THE BUFFER
004270  043                      INX     H        ;UPDATE THE BUFFER POINTER
004271  042  125  040            SHLD    KPWPNT   ;PUT THE POINTER BACK
004274  311                      RET              ;AND END OF INTERRUPT

                                ;CODE FOR VERIFYING THAT THE PASSWORD ENTERED IS THE CORRECT AND VALID
                                ;PASSWORD
004275  021  262  040  PW.TST: LXI     D,PASSWORD ;D,E POINTS TO THE EXPECTED PASSWORD
004300  041  271  040            LXI     H,KPWBUF ;H,L POINTS TO THE TYPED IN BUFFER
004303  006  000                MVI     B,00     ;'B' WILL BE THE COUNTER

004305  032              PW..:  LDAX    D        ;FETCH UP AN EXPECTED CHARACTER
004306  247                      ANA     A        ;SET THE FLAGS
004307  312  324  010            JZ      PW.END   ;IF 'END', GO MAKE SURE TYPEIN IS TERMINATED

004312  004                      INR     B        ;ELSE UPDATE OUR COUNTER
004313  276                      CMP     M        ;AND COMPARE A CHAR
004314  302  333  010            JNZ     PW.ERR   ;IF MIS-COMPARE REPORT IT AS ERROR

004317  023                      INX     D        ;UPDATE EXPECTED POINTER
004320  043                      INX     H        ;UPDATE TYPED IN POINTER
004321  303  305  010            JMP     PW..     ;CONTINUE
```

```
 2258                          ;END,, THIS IS ONLY TO VERIFY THAT TYPED TERMINATED AT THE SAME NUMBER OF
 2259                          ;CHARACTERS AS EXPECTED
 2260  004324  072  127  040   PW.END: LDA     KPWCNT  ;GET EXPECTED COUNT
 2261  004327  270                 CMP     B       ;CHECK AGAINST THE CURRENT COUNT
 2262  004330  312  355  010         JZ      PW.OK   ;AND JUMP IF COUNTS MATCH
 2263
 2264                          ;FALL THRU TO ERROR IF CHARACTER COUNTS DON'T MATCH
 2265  004333                 PW.ERR: KLINE   CMDNG   ;GIVE USER AN ERROR MESSAGE
  (1)  004333  315  137  004         CALL    KLINE   ;PRINT LINE OF CHARS
  (1)  004336  026  037               .ADDR   CMDNG   ;BUFF TO PRINT
 2266  004340  041  277  040         LXI     H,PWRTRY ;HAD ERROR, ONLY GET 3 CHANCES FOR ERRORS
 2267  004343  064                 INR     M       ;UPDATE ERROR COUNT
 2268  004344  176                 MOV     A,M     ;PLACE COUNT IN ACCUM FOR A TEST
 2269  004345  376  003             CPI     3       ;SEE IF STRUCK OUT
 2270  004347  312  373  010         JZ      KLIRST  ;GO RESET KLINIK LINE IF USER STRUCK OUT
 2271
 2272  004352  303  213  010         JMP     MODE1   ;ELSE GIVE HIM 'PW:' MESSAGE AGAIN
 2273
 2274                          ;HERE IF EVERYTHING MATCHED
 2275  004355  315  101  034   PW.OK:  CALL    SETM2   ;CHANGE LINE TO MODE 2
 2276  004360                 KCHAR   'O      ;WHEN GOOD PW, SENT OUT AN 'OK'
  (1)  004360  315  117  004         CALL    KCHR    ;GO PRINT THE CHARACTER
  (1)  004363  117               .BYTE   'O
 2277  004364                 KCHAR   'K
  (1)  004364  315  117  004         CALL    KCHR    ;GO PRINT THE CHARACTER
  (1)  004367  113               .BYTE   'K
 2278  004370                 PW.OUT: CLRB    PWRTRY  ;CLEAR ERROR COUNTER
 2279  004372  311                 RET             ;EXIT
 2280
 2281                          ;KLINIK LINE RESET CODE, FOR RESETING KLINIK LINE AND HANGING UP THE USER
 2282  004373  315  134  034   KLIRST: CALL    HANGUP  ;GO HANG UP THE KLINIK LINE
 2283  004376  315  006  034         CALL    SETM1   ;DROP BACK TO MODE 1
 2284  004401  303  370  010         JMP     PW.OUT  ;ZAP ERROR FLAG THEN OUT
```

```
                                ;KLINIK LINE MODE 2, THIS IS STREAM INPUT/OUTPUT, ALL CHARACTERS FROM
                                ;THE KLINIK UART ARE SENT TO THE SPECIAL KLINIK COMMUNICATION WORDS
                                ;AND ALL WORDS FROM THE KLINIK COMM WORDS ARE OUTPUT TO THE KLINIK LINE.
004404  376  034        MODE2:  CPI     CNBCK   ;FIRST SEE IF THE KLINIK USER WANTS A MODE CHANGE
004406  302  034  011          JNZ     KL3435  ;IF NOT, GO SEND INFO TO THE KLINIK COMM WORD

004411  072  251  040          LDA     MMFLG   ;BEFORE GOING TO MODE 3, SEE IF MM MODE
004414  247                    ANA     A       ;SET 8080 FLAGS
004415  302  304  006          JNZ     END.USR ;IF YES, MM MODE, ACT LIKE FROM A CTY

                                ;FALL THRU IF WANTS TO CHANGE MODES, BUT BEFORE CHANGING, CHECK IF HE'S
                                ;ALLOWED TO CHANGE MODES.
004420  072  300  040          LDA     KLLINE.ON ;CHECK IF KLINIK IS ON, & USER IS ALLOWED TO CHANGE
004423  247                    ANA     A       ;SET FLAGS
004424  310                    RZ              ;IF NOT ENABLED TO CHANGE, JUST IGNORE THIS INTERRUPT

                                ;WELL, HE IS ALLOWED TO CHANGE, SEE IF THE FRONT PANEL SWITCH IS UNLOCKED
004425  333  301              IN      BOOTSW  ;***** I/O RD 301 *****
                                ;NOTE THAT BIT LO IS TRUTH, IF BIT HI IS FALSE(DISABLED)
004427  346  004              ANI     4       ;CHECK THE CONSOLE ENABLE BIT
004431  312  040  034          JZ      SETM3   ;GO MODE 3 ONLY IF PANEL NOT LOCKED.LET SETM3 DO 'RET'
```

```
2308                        ;THIS IS WHERE YOU ACTUALLY WRITE THE DESIRED CHARACTER INTO THE KLINIK
2309                        ;LINE COMMUNICATION WORD
2310  004434                KL3435:
2311  004434  062  035  040          STA     CHRBUF ;PUT CHARACTER IN A RAM BUFFER
2312  004437  076  034              MVI     A,^O34 ;DESIRED ADDRESS FOR DEPOSTING CHARACTER
2313
2314  004441  323  103              OUT     A2835   ;WRITE ONLY RELEVANT PIECE OF THE ADDRESS
2315  004443  257                    XRA     A       ;THEN CLR ACCUM
2316  004444  323  105              OUT     A2027   ;AND CLR THE REST OF THE HARDWARE ADDRESS REGISTER
2317  004446  323  107              OUT     A1219
2318  004450  323  106              OUT     W1219   ;CLEAR PIECES OF DEPOSIT DATA WHICH MUST BE ZERO
2319  004452  323  110              OUT     W0411
2320  004454  323  112              OUT     W0003
2321
2322  004456  076  002              MVI     A,02    ;BIT TO SAY 'WRITE FUNCTION'
2323  004460  323  113              OUT     A0003   ;***** I/O WRT 113 *****
2324
2325                        ;THE FOLLOWING 'ADD A' WORKS BY LUCK..I.E. 2+2=4
2326  004462  207                    ADD     A       ;BIT INTO ACCUM FOR 'COM/ADR CYCLE'
2327  004463  323  115              OUT     BUSARB  ;***** I/O WRT 115/4 *****
2328
2329  004465  072  035  040          LDA     CHRBUF  ;NOW GET THE CHARACTER WE WANT
2330  004470  323  102              OUT     W2835   ;PUT IT IN THE HARDWARE REGISTER
2331  004472  076  001              MVI     A,1     ;AND GET THE VALID BIT TO GO WITH THE CHARACTER
2332  004474  323  104              OUT     W2027   ;PUT IT IN THE HARDWARE REGISTER
2333
2334                        ;AND BY LUCK, THE ACCUM HAS JUST WHAT WE NEED FOR THE NEXT STEP
2335  004476  323  114              OUT     DTARB   ;***** I/O WRT 114/1 *****
2336
2337  004500  076  360              MVI     A,^O360 ;BITS FOR 'CHECK NXM','CONSOLE REQ','T ENB FOR COM/ADR'
2338                                          ;'T ENB FOR DATA CYCLE'
2339  004502  323  210              OUT     BUSCTL  ;*****I/O WRT 210/360 *****
2340
2341                        ;DO THIS TWICE TO GUARANTEE THAT THE INTERRUPT HAPPENS
2342  004504  076  001              MVI     A,1     ;BIT FOR SETTING INTERRUPT TO THE KS10
2343  004506  323  116              OUT     INT2KS  ;SET THE INTERRUPT
2344  004510  323  116              OUT     INT2KS  ;SET THE INTERRUPT
2345
2346  004512  311                    RET             ;AND EXIT NORMALLY....
```

```
 2348                           .SBTTL **** CONSOLE COMMANDS AS IMPLIMENTED ****
 2349                           .SBTTL  *** 'EB' CMD ***
 2350                           ;THIS CODE PERFORMS THE 'EXAMINE BUS' CONSOLE FUNCTION.
 2351  004513  076  001        EBCMD:  MVI     A,01    ;FIRST CLR 'R CLK ENB'
 2352  004515  323  210                OUT     BUSCTL  ;***** I/O WRT 210/001 *****
 2353  004517  315  074  033          CALL    RDATT   ;***** I/O RD '0,1,2,3,103' (READ BITS 0-35) *****
 2354  004522  012  040                .ADDR   EMBUF   ;PLACE BITS 0-35 INTO RAM BUFFER AREA 'EMBUF'
 2355
 2356                           ;READ THE REST OF THE I/O REGISTERS AND SAVE IN THE RAM
 2357  004524  041  172  040          LXI     H,RM100 ;GET BEGINNING ADDRESS OF RAM BUFFER AREA
 2358  004527  021  230  011          LXI     D,IORGS ;D,E WILL POINT TO SOURCE OF REGS TO BE READ
 2359  004532  006  010                MVI     B,8     ;THERE ARE 8 REGISTERS TO BE READ
 2360
 2361  004534  032              EB.RDIN: LDAX   D       ;FETCH UP FIRST REGISTER TO BE READ
 2362  004535  315  267  013          CALL    ER.UTL  ;CALL ER COMMAND
 2363  004540  167                    MOV     M,A     ;COPY RESULTS OF READ INTO THE RAM SPACE
 2364  004541  023                    INX     D       ;UPDATE SOURCE POINTER
 2365  004542  043                    INX     H       ;UPDATE DESTINATION POINTER
 2366  004543  005                    DCR     B       ;DOWN THE COUNTER
 2367  004544  362  134  011          JP      EB.RDIN ;CONTINUE LOOP
 2368
 2369  004547  257                    XRA     A       ;CLR ACCUM  MUST SET 'R CLK ENB'
 2370  004550  323  210                OUT     BUSCTL  ;***** I/O WRT 210/0 *****
 2371
 2372  004552                         PLINE   EBHED   ;EB CMD HEADER MSG
  (1)  004552  337                    RST     3       ;PRINT LINE OF CHARS
  (1)  004553  041  037                .ADDR   EBHED   ;BUFF TO PRINT
 2373  004555  315  205  034          CALL    DECNET  ;PRINT THE HEADING
 2374  004560  315  347  030          CALL    P36.    ;GO PRINT IT
 2375  004563                         PCRLF           ;AND A <CR><LF>
  (2)  004563  347                    RST     4
  (2)  004564  002                    .BYTE   2
 2376  004565  315  205  034          CALL    DECNET  ;AND MAKE SURE THIS GETS SENT
```

```
 2378  004570  041  230  011          LXI     H,IORGS ;'H,L' NOW PNTS TO LIST OF I/O REGISTER NAMES
 2379  004573  021  172  040          LXI     D,RM100 ;'D,E' NOW PNTS TO CORRESPONDING LIST OF DATA FOR I/O REG
 2380  004576  006  010               MVI     B,8     ;ACCUM NOW CONTAINS A COUNT OF 8 (FOR 8 I/O REGS)
 2381
 2382  004600  315  254  030  EB1:    CALL    P8BIT   ;PRINT FIRST REG NAME
 2383  004603  043                    INX     H       ;BUMP TO NEXT
 2384  004604                         PSLASH          ;PRINT '1'
  (2)  004604  317                    RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (2)  004605  057                    .BYTE   SLASH   ;CHAR TO PRINT
 2385  004606  353                    XCHG            ;SWAP SO 'H,L' POINTS TO DATA
 2386
 2387  004607  315  254  030          CALL    P8BIT   ;PRINT DATA FOR THAT REG
 2388  004612  043                    INX     H       ;BUMP TO NEXT
 2389  004613  353                    XCHG            ;SWAP BACK-'H,L' POINTS TO NAME AGAIN
 2390  004614                         PSPACE          ;SPACE OVER
  (2)  004614  317                    RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (2)  004615  040                    .BYTE   SPACE   ;CHAR TO PRINT
 2391  004616  005                    DCR     B       ;DOWN COUNT
 2392  004617  302  200  011          JNZ     EB1     ;CONTINUE TILL DONE ALL EIGHT REGS
 2393  004622                         PCRLF
  (2)  004622  347                    RST     4
  (2)  004623  002                    .BYTE   2
 2394  004624  315  205  034          CALL    DECNET  ;AND FINALLY MAKE SURE LAST THING GETS SENT
 2395  004627                         ENDCMD          ;END-OF-COMMAND
  (1)  004627  311                    RET             ;RETURN TO CALLER
 2396
 2397                         ;END THIS CODE WITH A 6 BYTE BUFFER OF THE I/O REGS NAMES, IN BINARY
 2398  004630  100            IORGS:  .BYTE   ^O100   ;FIRST REG NAME IN BINARY
 2399  004631  101                    .BYTE   ^O101   ;2ND
 2400  004632  102                    .BYTE   ^O102   ;3RD
 2401  004633  103                    .BYTE   ^O103   ;4TH
 2402  004634  300                    .BYTE   ^O300   ;5TH
 2403  004635  301                    .BYTE   ^O301   ;6TH
 2404  004636  302                    .BYTE   ^O302   ;7TH
 2405  004637  303                    .BYTE   ^O303   ;8TH
 2406
```

```
2408                        .SBTTL  *** 'DB' CMD ***
2409                        ;THIS CODE PERFORMS THE 'DEPOSIT BUS' CONSOLE FUNCTION
2410  004640               DBCMD:  RUN.,           ;IS CPU RUNNING??
 (1)  004640  347                    RST     4
 (1)  004641  006                    .BYTE   6
2411  004642  332  251  011         JC      DB1     ;SKIP CODE IF AT END OF COMMAND
2412  004645                       ARG36           ;IF NOT, GO ASSEMBLE ARG.
 (1)  004645  347                    RST     4
 (1)  004646  010                    .BYTE   8
2413  004647  042  040              .ADDR   BUSAD   ;AND PUT INTO BUFFER 'BUS AD'
2414
2415  004651  315  171  033  DB1:   CALL    ADATT   ;***** I/O WRT TO R DATA 0-35 DATA REG(ODDS) *****
2416  004654  042  040              .ADDR   BUSAD   ;BUFFER ADDRESS OF SOURCE OF DATA
2417
2418  004656  257                    XRA     A       ;CLR ACCUM SO CAN CLR I/O REG 115
2419  004657  323  115              OUT     BUSARB  ;***** I/O WRT 115/0 *****
2420
2421  004661  076  141              MVI     A,^O141 ;BITS TO SET 'CONSOLE REQ' & 'T ENB FOR COM/ADR'
2422  004663  323  210              OUT     BUSCTL  ;***** I/O WRT 210/141 *****
2423
2424  004665                       BUSRESP ARBRESP ;***** I/O READ 301 *****
 (1)  004665  315  245  032         CALL    BUSRESP ;DO A CALL TO EXECUTE THIS CODE
 (1)  004670  020                    .BYTE   ARBRESP ;BITS TO BE CHECKED
2425  004671  302  022  036         JNZ     NOARB   ;IF NO ARB RESPONSE WITH 'BUS REQ', ABORT
2426
2427  004674  315  353  011         CALL    DBRDIN  ;GO READ RESULTS, AND DO A 36-BIT COMPARE
2428  004677  302  370  011         JNZ     DBERR   ;IF 'Z-BIT' NOT SET REPORT MISCOMPARE & ABORT
2429
```

```
 2431                          ;SECOND HALF OF COMMAND
 2432  004702  315  171  033           CALL    ADATT   ;CLR OUT OLD CRUD
 2433  004705  364  016                 .ADDR   MAD000  ;WITH ALL ZEROES
 2434
 2435  004707  315  136  033           CALL    WDATT   ;***** I/O WRT DATA 0-35 ADDR REG(EVENS) *****
 2436  004712  042  040                 .ADDR   BUSAD   ;BUFFER ADDRESS OF SOURCE OF DATA
 2437
 2438  004714  076  001                 MVI     A,01    ;BITS TO SET 'DATA CYCLE'
 2439  004716  323  114                 OUT     ^O114   ;***** I/O WRT 114/1 *****
 2440
 2441  004720  076  363                 MVI     A,^O363 ;BITS FOR 'CONSOLE REQ', 'TENB FOR COM/AD R'
 2442                                                    ;'T ENB FOR DATA CYCLE','LATCH DATA SENT'
 2443  004722  323  210                 OUT     BUSCTL  ;***** I/O WRT 210/363 *****
 2444
 2445  004724                          BUSRESP ARBRESP ;***** I/O RD 301 *****
  (1)  004724  315  245  032           CALL    BUSRESP ;DO A CALL TO EXECUTE THIS CODE
  (1)  004727  020                     .BYTE   ARBRESP ;BITS TO BE CHECKED
 2446
 2447  004730  302  022  036           JNZ     NOARB   ;IF NO ARB RESP, ABORT WITH 2ND HALF MESSAGE
 2448
 2449  004733                          BUSRESP DATACK  ;HOW ABOUT DATA ACKNOWLEDGE??
  (1)  004733  315  245  032           CALL    BUSRESP ;DO A CALL TO EXECUTE THIS CODE
  (1)  004736  001                     .BYTE   DATACK  ;BITS TO BE CHECKED
 2450  004737  312  010  036           JZ      NOACK   ;JUMP IF NONE
 2451
 2452  004742  315  353  011           CALL    DBRDIN  ;GO READ IN RESULTS AND COMPARE RESULT
 2453  004745  310                     RZ              ;IF Z-BIT, THEN OK TO EXIT
 2454
 2455                          ;ELSE FALL INTO THIS CODE IF A MISCOMPARE
 2456  004746                          PCHAR   DCHR    ;PRINT 'D'
  (1)  004746  317                     RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (1)  004747  104                     .BYTE   DCHR    ;CHAR TO PRINT
 2457  004750  303  372  011           JMP     DBCOM   ;AND NOW THE ERR FINISHES UP LIKE THE 1ST HALF
 2458
 2459                          ;COMMON SUBROUTINE TO READ IN THE CONTENTS OF THE KS10 BUS & COMPARE AGAINST
 2460                          ;THE DATA WHICH WAS PUT ONTO THE BUS
 2461  004753  315  074  033   DBRDIN: CALL    RDATT   ;***** I/O RD 0,1,2,3,103 *****
 2462  004756  100  040                 .ADDR   TMPB2   ;PLACE TO PUT RDATA 0-35
 2463
 2464  004760  315  022  033           CALL    CMP36   ;CHECK DATA JUST READ VS. DATA SENT
 2465  004763  042  040                 .ADDR   BUSAD   ;SENT DATA
 2466  004765  100  040                 .ADDR   TMPB2   ;RECEIVED DATA
 2467  004767  311                     RET             ;DONE
```

```
 2469                           ;'DB' COMMAND CODE FOR THE CASES WHERE DATA DEPOSITED ON THE
 2470                           ;BUS IS NOT THE SAME AS THE DATA READ BACK FROM THE BUS..
 2471                           ;...BY THE WAY..RIGHT NOW THIS IS PRETTY SLOPPY CODE....
 2472  004770                   DBERR:  PCHAR   CCHR    ;PRINT A 'C'
  (1)  004770  317                       RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (1)  004771  103                       .BYTE   CCHR    ;CHAR TO PRINT
 2473  004772                   DBCOM:  PLINE   MSG10   ;'?C CYC'
  (1)  004772  337                       RST     3       ;PRINT LINE OF CHARS
  (1)  004773  062  037                  .ADDR   MSG10   ;BUFF TO PRINT
 2474
 2475  004775  041  042  040            LXI     H,BUSAD ;ADDR OF 36-BIT DATA
 2476  005000  315  352  030            CALL P36        ;NOW PRINT THAT DATA
 2477
 2478  005003                           PLINE   DRCVD   ;'RECEIVED DATA'
  (1)  005003  337                       RST     3       ;PRINT LINE OF CHARS
  (1)  005004  075  037                  .ADDR   DRCVD   ;BUFF TO PRINT
 2479
 2480  005006  041  100  040            LXI     H,TMPB2 ;THIS IS ADDR OF RECEIVED DATA
 2481  005011  315  352  030            CALL    P36     ;PRINT THAT 36-BIT DATA
 2482  005014                           PCRLF           ;AND CR-LF
  (2)  005014  347                       RST     4
  (2)  005015  002                       .BYTE   2
 2483  005016  041  004  000            LXI     H,4     ;PASS ERROR CODE BEFORE EXIT
 2484  005021  303  016  036            JMP     ERRRTN  ;GO FINISH WITH THE ERROR CODE
```

```
 2486                           .SBTTL  *** 'EM' CMD ***
 2487                           ;THIS IS THE ACTUAL 'EM' COMMAND CODE
 2488
 2489  005024  332  032  012   EMCMD:  JC      EM1     ;SKIP CODE IF AT END OF COMMAND
 2490  005027  315  120  013           CALL    LACMD   ;GO FETCH UP AN ADDRESS TO EXAMINE
 2491
 2492  005032  257           EM1:    XRA     A       ;CLEAR ACCUM
 2493  005033  062  031  040           STA     ENEXT   ;AND SET SO 'EN ' CMD WILL KNOW WHAT TO DO
 2494
 2495  005036  021  017  040   EM2:    LXI     D,MEMAD ;ADDRESS FOR MEMORY LOC.
 2496
 2497  005041  076  004        EMINT:  MVI     A,04    ;BIT TO SAY 'READ FUNCTION'
 2498
 2499  005043  107           EN2ND:  MOV     B,A     ;SAVE FUNCTION DATA
 2500  005044  353                   XCHG            ;DATA POINTER TO 'H,L'
 2501  005045  042  205  040           SHLD    AM.AI   ;STORE FOR LATER USE BY COMMON CODE
 2502  005050  353                   XCHG            ;RESTORE 'D,E'
 2503  005051  315  176  033           CALL    ADATP   ;***** I/O WRT 103,105,107,111,113 *****
 2504
 2505  005054  170                   MOV     A,B     ;GET FUNCTION
 2506  005055  323  113        EM.CRM: OUT     A0003   ;***** I/O WRT 113/4 *****
 2507
 2508                           ;NOW SET 'COM/ADR' CYCLE
 2509  005057  076  004               MVI     A,^004  ;BIT TO SET COM/ADR CYC
 2510  005061  323  115               OUT     BUSARB  ;***** I/O WRT 115/4 *****
 2511
 2512                           ;CHECK IF DOING EI OR EM
 2513  005063  072  163  040           LDA     EIFLAG  ;GET THE EI FLAG
 2514  005066  247                   ANA     A       ;SET CODES, IF .NE. 0, THEN IT IS AN EI CODE
 2515  005067  302  074  012           JNZ     EMCONT  ;AND IF WAS EI, GO DO IT
```

```
 2517                          ;OTHERWISE JUST FALL THRU AND USE THE DM CODES
 2518  005072  076  343              MVI     A,^O343 ;BITS FOR 'CHECK NXM','CONSOLE REQ','T ENB FOR COM/ADR'
 2519                                                 ;'LATCH DATA SENT',' R CLK DISABLE'
 2520  005074  323  210        EMCONT: OUT     BUSCTL  ;***** I/O WRT 210/343 *****
 2521
 2522  005076  257                    XRA     A       ;CLEAR THE ACCUM
 2523  005077  062  163  040          STA     EIFLAG  ;CLEAR FLAG ON THE WAY OUT
 2524
 2525  005102                         BUSRESP ARBRESP ;***** I/O RD 301 *****
  (1)  005102  315  245  032          CALL    BUSRESP ;DO A CALL TO EXECUTE THIS CODE
  (1)  005105  020                    .BYTE   ARBRESP ;BITS TO BE CHECKED
 2526  005106  302  022  036          JNZ     NOARB   ;IF GET NO 'BUS REQ', ARB FAILED SO ABORT
 2527
 2528  005111                         BUSRESP NONXMEM ;***** I/O RD 301 *****
  (1)  005111  315  245  032          CALL    BUSRESP ;DO A CALL TO EXECUTE THIS CODE
  (1)  005114  100                    .BYTE   NONXMEM ;BITS TO BE CHECKED
 2529  005115  302  036  036          JNZ     NIXOM   ;JUMP IF NON-EXISTANT MEM FLAG IS SET
 2530
 2531                          ;NOW MUST WAIT FOR 'DATA ACKNOWLEDGE' FROM MEMORY
 2532  005120                         BUSRESP DATACK  ;***** I/O RD 301 *****
  (1)  005120  315  245  032          CALL    BUSRESP ;DO A CALL TO EXECUTE THIS CODE
  (1)  005123  001                    .BYTE   DATACK  ;BITS TO BE CHECKED
 2533  005124  312  010  036          JZ      NOACK   ;JMP IF NO 'DATA ACK'(BUS HAS 15 MIC. SEC TO RESP)
```

```
 2535                          ;HERE IF 'DATA ACKNOWLEDGE' RECEIVED,,GET RESULTS & PRINT
 2536  005127  021  012  040          LXI     D,EMBUF ;PLACE TO PUT RECEIVED DATA
 2537  005132  315  101  033          CALL    RDATP   ;***** I/O RD 0,1,2,3,103 *****
 2538
 2539  005135  257                    XRA     A       ;SET ACCUM .EQ. 0 FOR 'R CLK ENABLE'
 2540  005136  323  210               OUT     BUSCTL  ;***** I/O WRT 210/0 *****
 2541
 2542  005140  072  120  040          LDA     NOPNT   ;GET THE PRINT FLAG
 2543  005143  247                    ANA     A       ;SET CONDITION CODES
 2544  005144  300                    RNZ             ;AND DONT WASTE TIME IF NOT PRINTING
 2545
 2546  005145  052  205  040          LHLD    AM,AI   ;GET POINTER TO MEM ADDR JUST EXAMINED
 2547  005150  315  352  030          CALL    P36     ;PRINT IT
 2548
 2549  005153                         PSLASH          ;PRINT '/'
  (2)  005153  317                    RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (2)  005154  057                    .BYTE   SLASH   ;CHAR TO PRINT
 2550  005155  315  347  030          CALL    P36.    ;AND PRINT IT
 2551  005160                         PCRLF           ;CR-LF
  (2)  005160  347                    RST     4
  (2)  005161  002                    .BYTE   2
 2552  005162                         ENDCMD          ;ALL DONE
  (1)  005162  311                    RET             ;RETURN TO CALLER
 2553
```

```
 2555                           .SBTTL  *** 'EN' CMD ***
 2556                           ;ACTUAL CODE FOR 'EN' CMD
 2557
 2558  005163  315  055  033    ENEM:   CALL    INC36   ;ADD 1 TO 36-BIT BUFFER
 2559  005166  017  040                  .ADDR   MEMAD   ;THIS IS THE BUFFER TO INCREMENT
 2560  005170  303  032  012            JMP     EM1     ;AND NO GO PROCESS JUST LIKE 'EM' CMD
 2561
 2562
 2563                           .SBTTL  *** 'EN' CND ***
 2564                           ;EXAMINE NEXT WILL DO THE NEXT, SAME AS THE LAST
 2565  005173  052  031  040    ENCMD:  LHLD    ENEXT   ;GET INDEX FOR WHICH EXAMINE IS NEXT
 2566  005176  021  207  012            LXI     D,ENLST ;GET PNTR TO DISPATCH LIST
 2567  005201  031                    DAD     D       ;AND NOW ADD 'WHICH' EXAMINE
 2568  005202  136                    MOV     E,M     ;GET LO ORDER PIECE
 2569  005203  043                    INX     H       ;UPDATE MEM PNTR
 2570  005204  126                    MOV     D,M     ;GET HI ORDER PIECE
 2571  005205  353                    XCHG            ;PUT THIS NEW ADDR INTO 'H,L'
 2572  005206  351                    PCHL            ;AND TAKE THE DISPATCH
 2573
 2574  005207  163  012          ENLST:  .ADDR   ENEM    ;DISPATCH FOR EXAM MEM CMD
 2575  005211  001  013                  .ADDR   ENEI    ;DISPATCH FOR EXAM I/O
 2576  005213  106  013                  .ADDR   ENEK    ;DISPATCH FOR EXAM KONSOL
 2577  005215  057  016                  .ADDR   ENEC    ;DISPATCH FOR EXAMINCRAM
```

```
                                ,SBTTL  *** 'DM' CMD ***
                                ;DEPOSIT MEMORY ACTUAL COMMAND CODE
                                ,SBTTL  *** 'DN' CMD ***
005217  315  055  033   DNDM:   CALL    INC36   ;INCREMENT MEMORY ADDRESS
005222  017  040                ,ADDR   MEMAD   ;HERE IS CURRENT MEMORY ADDRESS
005224                  DMCMD:  ARG36           ;OTHERWISE, ASSEMBLE THE ARG
(1) 005224  347                 RST     4
(1) 005225  010                 ,BYTE   8
005226  047  040                ,ADDR   DMDAT   ;PLACE TO PUT ASSEMBLED DATA

005230  257            DM1:     XRA     A       ;0 IS THE INDEX FOR MEM NEXT CMDS
005231  062  033  040            STA     DNEXT   ;SAVE SO 'NEXT' COMMAND WILL KNOW WHAT TO DO

005234  021  017  040   DM2:     LXI     D,MEMAD ;PNTR TO SM10 MEMORY ADDRESS

005237  076  002        DMINT:   MVI     A,02    ;BIT TO SAY 'WRITE FUNCTION'

005241  107            DN2ND:   MOV     B,A     ;SAVE FUNCTION STATUS
005242  315  176  033            CALL    ADATP   ;***** I/O WRT 103,105,107,111,113 *****

005245  170                     MOV     A,B     ;GET FUNCTION DATA
005246  323  113                OUT     A0003   ;***** I/O WRT 113 *****

005250  076  004                MVI     A,04    ;BIT INTO ACCUM FOR 'COM/ADR CYCLE'
005252  323  115                OUT     BUSARB  ;***** I/O WRT 115/4 *****

005254  315  136  033            CALL    WDATT   ;***** I/O WRT 102,104,106,110,112 *****
005257  047  040                ,ADDR   DMDAT   ;PLACE TO GET DATA FOR DEPOSIT

005261  076  001                MVI     A,01    ;BIT INTO ACCUM FOR 'DATA CYCLE'
005263  323  114                OUT     DTARB   ;***** I/O WRT 114/1 *****

                                ;CHECK TO SEE IF DOING DI OR DM
005265  072  164  040            LDA     DIFLAG  ;GET THE FLAG
005270  247                     ANA     A       ;SET THE CONDITION CODES
005271  302  276  012            JNZ     DMCONT  ;IF .NE. 0, THEN YOU GOT THE CODE FOR A DI
```

```
 2614                           ;OTHERWISE, FALL THRU TO DO A DM
 2615  005274  076  362         DMGO:   MVI     A,^O362 ;BITS FOR 'CHECK NXM','CONSOLE REQ','T ENB FOR COM/ADR'
 2616                                                   ;'T ENB FOR DATA CYCLE'(LATCH DATA SENT PREVENTS FALSE PAR ERR)
 2617  005276  323  210         DMCONT: OUT     BUSCTL  ;*****I/O WRT 210/362 *****
 2618
 2619  005300  257                         XRA     A       ;CLEAR THE ACCUM
 2620  005301  062  164  040              STA     DIFLAG  ;AND CLEAR THE FLAG
 2621
 2622  005304                             BUSRESP ARBRESP ;***** I/O RD 301 *****
  (1)  005304  315  245  032              CALL    BUSRESP ;DO A CALL TO EXECUTE THIS CODE
  (1)  005307  020                        .BYTE   ARBRESP ;BITS TO BE CHECKED
 2623  005310  302  022  036              JNZ     NOARB   ;IF NO 'BUS REQ', ARB FAILED, SO ABORT
 2624
 2625                           ;IF THAT WAS OK, CHECK FOR NON-EXISTANT MEMORY
 2626  005313                             BUSRESP NONXMEM ;***** I/O RD 301 *****
  (1)  005313  315  245  032              CALL    BUSRESP ;DO A CALL TO EXECUTE THIS CODE
  (1)  005316  100                        .BYTE   NONXMEM ;BITS TO BE CHECKED
 2627  005317  302  036  036              JNZ     NIXOM   ;IF FLAG SAYS NXM, THEN WE JUMP
 2628
 2629                           ;ELSE ALL OK....
 2630  005322                             ENDCMD
  (1)  005322  311                        RET             ;RETURN TO CALLER
```

```
 2632                           .SBTTL  *** 'DN' CMD ***
 2633                           ;ROUTINE WILL DEPOSIT NEXT, JUST AS THE LAST
 2634  005323  052  033  040    DNCMD:  LHLD    DNEXT   ;GET CODE FOR WHICH DEPOSIT IS NEXT
 2635  005326  021  337  012            LXI     D,DNLST ;PNTR TO DISPATCH LIST
 2636  005331  031                    DAD     D       ;ADD GIVES PNTR TO WHICH IS NEXT
 2637
 2638  005332  136                    MOV     E,M     ;LO ORDER PIECE TO REG
 2639  005333  043                    INX     H       ;UPDATE MEM PNTR
 2640  005334  126                    MOV     D,M     ;HI ORDER PIECE TO REG
 2641  005335  353                    XCHG            ;NOW THE DISPATCH GOES TO 'H,L'
 2642  005336  351                    PCHL            ;AND DISPATCH
 2643
 2644  005337  217  012          DNLST:  .ADDR   DNDM    ;DISPATCH FOR DEP NEXT TO MEM
 2645  005341  007  013                  .ADDR   DNDI    ;FOR DEPOSIT NEXT TO I/O
 2646  005343  137  013                  .ADDR   DNDK    ;FOR DEPOSIT NEXT TO KONSOLE
 2647  005345  071  016                  .ADDR   DNDC    ;FOR DEPOSIT NEXT CRAM
```

```
 2649                           .SBTTL  *** 'EI' CMD ***
 2650  005347                   EICMD:  RUN..           ;ILLEGAL COMMAND IF CPU RUNNING
  (1)  005347  347                       RST     4
  (1)  005350  006                       .BYTE   6
 2651  005351  332  357  012            JC      EI1     ;SKIP CODE IF AT END OF COMMAND
 2652  005354  315  125  013            CALL    LICMD   ;FETCH UP THE DESIRED I/O ADDRESS
 2653
 2654  005357  076  002          EI1:    MVI     A,2     ;DISP CODE FOR EXAMINE NEXT..
 2655  005361  062  031  040            STA     ENEXT   ;TELL EXAMINE NEXT TO COME HERE
 2656
 2657  005364  021  024  040            LXI     D,IOAD  ;'H,L' GETS PNTR TO ADDR BUFFER
 2658
 2659  005367  076  143                  MVI     A,^O143 ;SPECIAL CODE FOR WHEN DOING DI
 2660  005371  062  163  040            STA     EIFLAG  ;PASS IT TO ROUTINE
 2661
 2662  005374  076  014                  MVI     A,^O14  ;BITS FOR 'I/O FUNC' & 'READ FUNC'
 2663  005376  303  043  012            JMP     EN2ND   ;JUMP TO COMMON CODE
 2664
 2665
 2666                           ;EXAMINE I/O ENTRY PNT FOR EXAMINE NEXT SITUATION
 2667  005401  315  040  013   ENEI:   CALL    IO.INC  ;GO INCREMENT I/O ADDRESS TWICE
 2668  005404  303  357  012            JMP     EI1     ;THEN ON TO COMMON CODE
```

```
 2670                           .SBTTL  *** 'DI' CMD ***
 2671                           .SBTTL  *** 'DN' CMD ***
 2672  005407  315  040  013   DNDI:   CALL    IO.INC  ;GO INCREMENT THE I/O ADDRESS TWICE
 2673  005412                 DICMD:  ARG36           ;OTHERWISE GO ASSEMBLE THE ARG
  (1)  005412  347                       RST     4
  (1)  005413  010                       .BYTE   8
 2674  005414  047  040                 .ADDR   DMDAT   ;AND STORE IT HERE
 2675
 2676  005416  076  002         DI1:    MVI     A,02    ;SET WORD THAT SAYS DEP NEXT WILL BE DI
 2677  005420  062  033  040             STA     DNEXT   ;AND SAVE FOR 'DN' CMD
 2678
 2679  005423  021  024  040             LXI     D,IOAD  ;PNTR TO ADDRESS DATA TO USE
 2680
 2681  005426  076  160                 MVI     A,^O160 ;SET CODE FOR USE BY DI COMMAND
 2682  005430  062  164  040             STA     DIFLAG  ;AND PASS IT TO ROUTINE
 2683
 2684  005433  076  012                 MVI     A,^O012 ;BITS TO SAY 'I/O FUNC' & 'WRITE FUNC'
 2685  005435  303  241  012             JMP     DN2ND   ;AND JUMP TO COMMON CODE
 2686
 2687  005440  315  055  033   IO.INC: CALL    INC36   ;NOW INCREMENT I/O ADDRESS
 2688  005443  024  040                 .ADDR   IOAD    ;ITS RIGHT HERE
 2689  005445  315  055  033             CALL    INC36   ;NOW INCREMENT I/O ADDRESS
 2690  005450  024  040                 .ADDR   IOAD    ;ITS RIGHT HERE
 2691  005452  311                       RET             ;AND BACK
```

```
                        .SBTTL  *** 'EK' CMD ***
                        ;ROUTINE EXAMINES 8080 LOCATIONS
005453  332  062  013   EKCMD:  JC      EK1     ;IF NOT, NO ARG TO BE ASSEMBLED

                        ;OTHERWISE, MUST ASSEMBLE ARG
005456                          ARG16           ;GO GET 16 BIT ADDR TO EXAMINE
(1) 005456  347                 RST     4
(1) 005457  004                 .BYTE   4
005460  002  040                .ADDR   C80AD   ;AND PUT INTO CURRENT ADDR BUFFER

005462  076  004        EK1:    MVI     A,04    ;INDEX SAYS EK IS NEXT
005464  062  031  040           STA     ENEXT   ;SAVE IN THE RAM

005467  041  002  040           LXI     H,C80AD ;GET CURRENT ADDR
005472  315  317  030           CALL    P16     ;AND PRINT IT AS IS

005475                          PSLASH          ;NOW A '/'
(2) 005475  317                 RST     1       ;GO PRINT CHAR IN TRAILING BYTE
(2) 005476  057                 .BYTE   SLASH   ;CHAR TO PRINT

005477  052  002  040           LHLD    C80AD   ;GET ADDR JUST PRINTED
005502  176                     MOV     A,M     ;PASS ARG TO PRINT IN THE ACCUM
005503  303  031  017           JMP     P8CRLF  ; AND PRINT DATA PLUS CRLF, FOR FREE


                        .SBTTL  *** 'EN' CMD ***
005506  052  002  040   ENEK:   LHLD    C80AD   ;GET CURRENT 8080 ADDRESS
005511  043                    INX     H       ;UPDATE
005512  042  002  040           SHLD    C80AD   ;PUT IT BACK
005515  303  062  013           JMP     EK1     ;COMMON CODE
```

```
 2721                           .SBTTL  *** 'LA' CMD ***
 2722  005520                   LACMD:
 2723  005520                          ARG36           ;OTHERWISE, GET ARG & PUT IN 36-BIT BUFFER
  (1)  005520  347                     RST     4
  (1)  005521  010                     .BYTE   8
 2724  005522  017  040                .ADDR   MEMAD   ;PLACE TO PUT DATA
 2725
 2726  005524                          ENDCMD          ;AND DONE
  (1)  005524  311                     RET             ;RETURN TO CALLER
 2727
 2728
 2729                           .SBTTL  *** 'LI' CMD ***
 2730  005525                   LICMD:  ARG36           ;GET ARG AND PUT INTO A TEMP BUFFER
  (1)  005525  347                     RST     4
  (1)  005526  010                     .BYTE   8
 2731  005527  024  040                .ADDR   IOAD    ;THIS TEMP BUFFER
 2732
 2733  005531                          ENDCMD          ;AND DONE
  (1)  005531  311                     RET             ;RETURN TO CALLER
 2734
 2735                           .SBTTL  *** 'LK' CMD ***
 2736                           ;ROUTINE SETS CURRENT 8080 ADDRESS INTO RAM..
 2737                           ;IF USER TRIES TO DEPOSIT PROM, TOO BAD. HE SHOULD KNOW BETTER
 2738  005532                   LKCMD:  ARG16           ;IF OK, GO ASSEMBLE 16 BIT ARG
  (1)  005532  347                     RST     4
  (1)  005533  004                     .BYTE   4
 2739  005534  002  040                .ADDR   C8OAD   ;THIS IS A GOOD PLACE TO KEEP IT
 2740
 2741  005536                          ENDCMD          ;AND END
  (1)  005536  311                     RET             ;RETURN TO CALLER
```

```
 2743                          .SBTTL  *** 'DN' CMD ***
 2744  005537  052  002  040   DNDK:   LHLD    C80AD   ;GET 8080 ADDRESS
 2745  005542  043                 INX     H       ;INCREMENT BY 1
 2746  005543  042  002  040          SHLD    C80AD   ;PUT IT BACK
 2747                          ;FALL INTO THE 'DK' COMMAND
 2748
 2749                          .SBTTL  *** 'DK' CMD ***
 2750                          ;CODE TO DEPOSIT INTO 8080 RAM.. IF U TRY TO DEPOSIT PROM
 2751                          ;ITS YOUR OWN FAULT
 2752  005546  315  244  030   DKCMD:  CALL    ARG16.  ;OK, NOW GO ASSEMBLE 16 BITS OF DATA
 2753
 2754  005551  175                 MOV     A,L     ;GET DATA FROM LOC
 2755  005552  052  002  040          LHLD    C80AD   ;AND CURRENT ADDRESS TO 'H,L'
 2756  005555  167                 MOV     M,A     ;WRITE THE 8-BIT DATA
 2757  005556  076  004                 MVI     A,04    ;GET CODE THAT SAYS EXAMINE NEXT SHOULD BE 'KONSOLE'
 2758  005560  062  033  040          STA     DNEXT   ;AND SAVE IN RAM
 2759
 2760  005563                      ENDCMD          ;ALL DONE
  (1)  005563  311                 RET             ;RETURN TO CALLER
 2761
 2762                          .SBTTL  *** 'CP' CMD ***
 2763                          ;COMMAND TO SINGLE PULSE THE SM10 CPU CLK
 2764  005564  332  204  013   CPCMD:  JC      CP1     ;IF NO ARG, ONLY GIVE SINGLE CPU CLK
 2765
 2766  005567  315  244  030          CALL    ARG16.  ;ELSE GET ARG
 2767
 2768                          ;NOW GIVE NUMBER OF CLKS REQUESTED
 2769  005572  175            CPMLT:  MOV     A,L     ;LO ORDER PIECE INTO ACCUM
 2770  005573  264                 ORA     H       ;ADD THE HI ORDER PIECE
 2771  005574  310                 RZ              ;ALL DONE IF DOWN TO ZERO
 2772
 2773  005575  315  204  013          CALL    CP1     ;OTHERWISE, GIVE CLOCK
 2774  005600  053                 DCX     H       ;DECREMENT
 2775  005601  303  172  013          JMP     CPMLT   ;AND CONTINUE TILL DONE ALL
 2776
 2777  005604  076  010        CP1:   MVI     A,^O010 ;SET BIT FOR 'SS MODE'
 2778  005606  323  204                OUT     CRMCTL  ;*****I/O WRT 204/010 *****
 2779  005610  076  002                MVI     A,2     ;SET BIT FOR 'SINGLE CLK'
 2780  005612  323  206                OUT     CLKCTL  ;***** I/O WRT 206/2 *****
 2781  005614                      ENDCMD          ;DONE..
  (1)  005614  311                 RET             ;RETURN TO CALLER
```

```
 2783                           .SBTTL  *** 'ER' CMD ***
 2784                           ;COMMAND TO EXAMINE ONE OF THE 8080 INTERNAL REGISTER, AND DISPLAY
 2785                           ;THE CONTENTS OF THAT REGISTER.
 2786  005615  332  227  013   ERCMD:  JC      ER1     ;IF NO ARG, GO USE THE ONE ALREADY IN THE RAM
 2787
 2788  005620  315  244  030           CALL    ARG16.  ;ELSE, PICK UP THE ARG THAT WAS TYPED
 2789
 2790                           ;FALL TO HERE IF ARG OK..
 2791  005623  175                   MOV     A,L     ;GET ACTUAL ARG INTO THE ACCUM
 2792  005624  062  123  040           STA     ERADDR  ;WELL, BEST SAVE THIS THING IN THE RAM
 2793  005627  072  123  040   ER1:    LDA     ERADDR  ;COMMON TYPE CODE.. A NO-OP IF ARG WAS TYPED
 2794
 2795  005632  365                   PUSH    PSW     ;NOW SAVE ACCUM PLEASE
 2796  005633  315  304  030           CALL    P8BITA  ;PRINT NAME OF THE 8080 REG THAT IS BEING EXAMINED
 2797  005636                         PSLASH          ;AND SEPARATE FROM ITS CONTENTS WITH A SLASH
  (2)  005636  317                   RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (2)  005637  057                   .BYTE   SLASH   ;CHAR TO PRINT
 2798
 2799  005640  361                   POP     PSW     ;RESTORE ACCUM PLEASE
 2800  005641  315  267  013           CALL    ER.UTL  ;EXECUTE THE INSTR PAIR FROM THE RAM SPACE
 2801
 2802                           ;BACK HERE AND THE DATA IS IN THE ACCUM
 2803  005644  315  031  017           CALL    P8CRLF  ;PRINT THE RESULTS
 2804  005647  311                   RET             ;AND DONE
```

```
 2806                          ;ROUTINE TO EXECUTE AN 'IN' OR 'OUT' FROM THE 8080 RAM SPACE
 2807  005650  042  070  040   RAMXCT: SHLD    ER.LOC  ;THE 'IN/OUT' AND THE REG NUMBER INTO RAM SPACE
 2808  005653  365                 PUSH    PSW     ;SAVE ACCUM, IN CASE ROUTINE IS AN 'OUT'
 2809  005654  076  311             MVI     A,.RET  ;A 'RETURN' INTO ACCUM
 2810  005656  062  072  040         STA     ER.LOC+2 ; AND THEN THE RETURN GETS PUT INTO RAM SPACE
 2811  005661  361                 POP     PSW     ;RESTORE ACCUM, ANYWAY
 2812  005662  315  070  040         CALL    ER.LOC  ;GO EXECUTE THE RAM LOC
 2813  005665  057                 CMA             ;FIX HARDWARE INVERSION
 2814  005666  311                 RET             ;BACK TO CALLER
 2815
 2816                          ;ROUTINE ER.UTL., DOES AN EXAMINE REGISTER, INTERNAL TYPE FORMAT.
 2817                          ;NO PRINTING, JUST THE EXAMINE
 2818                          ;PASS DESIRED I/O REG ADDRESS IN ACCUM.
 2819                          ;ACCUM GETS THE RESULTS OF THE READ.
 2820  005667  345            ER.UTL: PUSH    H       ;SAVE H,L PAIR
 2821  005670  147                 MOV     H,A     ;NOW, THE NUMBER THAT WAS TYPED IS PUT INTO HI HALF
 2822  005671  056  333             MVI     L,.IN   ;AND AN 'IN' INSTR GOES LO HALF
 2823
 2824  005673  315  250  013         CALL    RAMXCT  ;NOW ACTUALLY EXECUTE THE CODE TO DO THE READ
 2825  005676  341                 POP     H       ;FIX H,L
 2826  005677  311                 RET             ;OUT
```

```
 2828                           .SBTTL  *** 'LR' CMD ***
 2829                           ;COMMAND TO SET INTO THE 8080 RAM, THE I/O REGISTER TO BE EITHER
 2830                           ;DEPOSITED OR EXAMINED
 2831  005700  315  244  030    LRCMD:  CALL    ARG16.  ;FETCH IN THE NUMBER TYPED
 2832
 2833  005703  175                    MOV     A,L     ;DESIRED REG TO ACCUM
 2834  005704  062  123  040            STA     ERADDR  ;PUT IN 8080 RAM
 2835  005707  311                    RET             ;AND OUT
 2836
 2837
 2838                           .SBTTL  *** 'DR' CMD ***
 2839                           ;COMMAND TO DEPOSIT A NUMBER INTO THE LAST SPECIFIED 8080 I/O REG.
 2840  005710                   DRCMD:  ARG16
  (1)  005710  347                    RST     4
  (1)  005711  004                    .BYTE   4
 2841  005712  000  040                .ADDR T80DT     ;TAKE ARG AND PUT INTO RAM SPACE
 2842
 2843  005714  056  323                MVI     L,.OUT  ;'L' GETS THE OPERATION TYPE WE WILL PERFORM
 2844  005716  072  123  040            LDA     ERADDR  ;FETCH UP THE CURRENTLY SELECTED I/O REG
 2845  005721  147                    MOV     H,A     ;AND PUT IT INTO THE 'H'
 2846  005722  072  000  040            LDA     T80DT   ;NOW THE DATA TO BE WRITTEN GOES TO THE ACCUM
 2847
 2848  005725  315  250  013            CALL    RAMXCT  ;PERFORM THE OPERATION
 2849  005730  311                    RET             ;THATS ALL
```

```
                                .SBTTL  *** 'LC' CMD ***
                                ;COMMAND TO LOAD THE 8080 RAM CURRENT CRAM ADDRESS
005731                          LCCMD:  ARG16           ;OK, ASSEMBLE THE 16 BITS
(1) 005731  347                         RST     4
(1) 005732  004                         .BYTE   4
005733  004  040                        .ADDR   CRMAD   ;TEMP PLACE TO KEEP BITS
005735                                  ENDCMD          ;DONE..
(1) 005735  311                         RET             ;RETURN TO CALLER

                                .SBTTL  *** 'CE' CMD ***
                                ;COMMAND TO SET CACHE ENABLE ON THE CSL BOARD
                                ; OR PERHAPS CLR CACHE ENABLE IF DESIRED

005736  332  367  013           CECMD:  JC      CEDIS   ;IF NO ARG, DISPLAY 'CACHE ENABLE'

005741  315  244  030                   CALL    ARG16.  ;MUST ASSEMBLE ARG IF FALL THRU

005744  175                             MOV     A,L     ;ARG TO ACCUM
005745  027                             RAL             ;BIT 0 TO 1
005746  027                             RAL             ;BIT TO 2
005747  027                             RAL             ;BIT TO 3
005750  346  010                       ANI     ^O10    ;OFF ALL BITS BUT THE CACHE BIT
005752  107                             MOV     B,A     ;SAVE RESULT IN 'B' FOR A LITTLE WHILE
005753  072  354  040                  LDA     PARBT   ;GET CURRENT PARITY BIT STATUS
005756  346  367                       ANI     ^O367   ;OFF THE CACHE BIT

                                ;HERE IS SOME COMMON CODE, USEFUL BY ROUTINES WHICH MUST ADJUST
                                ;THE DATA IN THE PARBT LOCATION
005760  260                   ENACOM: ORA     B       ;ADD NEW DATA TO DEFAULTED 'PARBT'
005761  062  354  040          KS.PAR: STA     PARBT   ;NOW SAVE THE NEW DEFAULT
005764  323  100                       OUT     RESET   ;***** I/O WRT 100/STUFF *****
005766                                  ENDCMD          ;AND ALL DONE
(1) 005766  311                         RET             ;RETURN TO CALLER

                                ;CODE ENTERED WHEN WE WANT TO DISPLAY THE CACHE ENABLE STATUS
005767  072  354  040          CEDIS:  LDA     PARBT   ;GET CURRENT STATUS
005772  346  010                       ANI     ^O10    ;IS THE CACHE BIT SET??
005774  302  003  014          CHOOSE: JNZ     PNT.ON  ;HERE IF YES
005777                                  PLINE   OFFMSG  ;OFF MESSAGE DEPENDING THINGS
(1) 005777  337                         RST     3       ;PRINT LINE OF CHARS
(1) 006000  154  037                    .ADDR   OFFMSG  ;BUFF TO PRINT
006002  311                             RET
006003                          PNT.ON: PCHAR   'O      ;PRINTING "ON" A CHAR AT A TIME SAVES 1 BYTE
(1) 006003  317                         RST     1       ;GO PRINT CHAR IN TRAILING BYTE
(1) 006004  117                         .BYTE   'O      ;CHAR TO PRINT
006005                                  PCHAR   'N      ;OVER PRINTING IT AS A STRING..SPACE IS A LITTLE TIGHT
(1) 006005  317                         RST     1       ;GO PRINT CHAR IN TRAILING BYTE
(1) 006006  116                         .BYTE   'N      ;CHAR TO PRINT
006007                                  PCRLF
(2) 006007  347                         RST     4
(2) 006010  002                         .BYTE   2
006011  311                             RET
```

```
 2893                           .SBTTL  *** 'TE' CMD ***
 2894                           ;CONSOLE COMMAND TO ENABLE OR DISABLE THE 1 MSEC CLOCK
 2895
 2896  006012  332  036  014   TECMD:  JC      TEDIS   ;IF NO ARG, DISPLAY CURRENT STATE
 2897
 2898  006015  315  244  030           CALL    ARG16.  ;OTHERWISE, GO FETCH THE ARG
 2899
 2900  006020  175                   MOV     A,L     ;GET INFO JUST TYPED
 2901  006021  027                   RAL             ;BIT 0 TO 1
 2902  006022  027                   RAL             ;BIT TO 2
 2903  006023  346  004               ANI     ^O4     ;OFF ALL BUT THE TIME BIT
 2904  006025  107                   MOV     B,A     ;SAVE STUFF IN B
 2905  006026  072  354  040           LDA     PARBT   ;GET CURRENT DEFAULT
 2906  006031  346  373               ANI     ^O373   ;OFF THE 1 MSEC CLOCK SIGNAL
 2907
 2908  006033  303  360  013           JMP     ENACOM  ;GO DO COMMON CODE
 2909
 2910                           ;THIS CODE ENTERED WHEN WE ONLY WANT TO DISPLAY CURRENT STATE OF 1 MSEC CLOCK
 2911  006036  072  354  040   TEDIS:  LDA     PARBT   ;WE NEED TO REPORT STATE.,GET DEFAULT
 2912  006041  346  004               ANI     ^O4     ;IS THE BIT SET??
 2913  006043  303  374  013           JMP     CHOOSE  ;GO TO A COMMON PLACE THAT CHOOSES 'YES' OR 'NO'
 2914
 2915                           .SBTTL  *** 'SC' CMD ***
 2916                           ;CODE TO TURN OFF OR ON, THE ABILITY TO RECOVER FROM SOFT CRAM ERRORS
 2917                           ;FLAG AT 0, MEANS TRY AND RECOVER, THEREFORE ITS THE DEFAULT ON
 2918                           ;MACHINE POWER ON...
 2919  006046  332  074  014   SCCMD:  JC      SCDIS   ;IF NO ARG TYPED, GO DISPLY STATE OF THE THING
 2920
 2921  006051  315  244  030           CALL    ARG16.  ;ELSE GO GATHER UUP AN ARGUMENT
 2922  006054  175                   MOV     A,L     ;ARG GOES INTO ACCUM
 2923  006055  247                   ANA     A       ;SET 8080 FLAGS
 2924  006056  312  066  014           JZ      SC.TOFF ;IF ZERO , TURN OFF SC SOFT CRAM RECOVERY
 2925
 2926                           ;FALL THRU IF TURNING ON SCE
 2927  006061  257                   XRA     A       ;ZERO ACCUM
 2928  006062  062  326  042           STA     SC.OFF  ;SO THAT WE CAN SET THE APPROPRIATE FLAG
 2929  006065                         ENDCMD          ;THAT'S IT
  (1)  006065  311                   RET             ;RETURN TO CALLER
 2930  006066  076  377       SC.TOFF: MVI    A,-1    ;WANT TO TURN OFF SCE, NEED -1 TO DO IT
 2931  006070  062  326  042           STA     SC.OFF  ;ZAP
 2932  006073                         ENDCMD          ;AND OUT
  (1)  006073  311                   RET             ;RETURN TO CALLER
 2933
 2934  006074  072  326  042   SCDIS:  LDA     SC.OFF  ;GRAB THE FLAG
 2935  006077  057                   CMA             ;SINCE 0 .EQ. ON, WE MUST INVERT FLAVOR OF FLAG
 2936  006100  247                   ANA     A       ;SET 8080 PROCESSOR FLAGS
 2937  006101  303  374  013           JMP     CHOOSE  ;AND GO PRINT THE RIGHT THING
```

```
                                .SBTTL  *** 'TP CMD' ***
                                ;CONSOLE COMMAND TO ENABLE OR DISABLE THE TEN STYLE TRAPS
006104  332  124  014  TPCMD:  JC      TPDIS   ;GO DISPLAY CURRENT STATE IF NOTHING TYPED

006107  315  244  030          CALL    ARG16.  ;OTHERWISE, GO ASSEMBLE A NUMBER TYPED IN

006112  175                  MOV     A,L     ;GET INFO THAT WAS TYPED
006113  027                  RAL             ;BIT 0 TO 1
006114  027                  RAL             ;1 TO 2
006115  027                  RAL             ;2 TO 3
006116  027                  RAL             ;3 TO 4
006117  346  020              ANI     ^O20    ;OFF ALL BUT TRAP BIT
006121  303  365  021          JMP     TP.SET  ;JUMP TO PLACE THAT SETS TRAPS, AND SAVES DATA

                                ;CODE TO DISPLAY CURRENT STATE OF SIGNAL
006124  072  355  040  TPDIS:  LDA     TRAPEN  ;GET CURRENT STATE OF TRAPS BIT
006127  346  020              ANI     ^O20    ;SET CONDITION CODES
006131  303  374  013          JMP     CHOOSE  ;AND GO DO IT
```

```
 2958                           ;Z%%.SBTTL      *** 'LT' CMD ***
 2959                           ;CONSOLE COMMAND TO TURN ON THE LIGHTS ON THE CONSOLE FRONT PANEL
 2960                           ;Z%%LTCMD:      CLRB    KLNKSW  ;FORCE A FIXING OF THE LIGHTS
 2961                           ;Z%%    MVI     A,7     ;LOAD ACCUM WITH A BIT FOR EACH OF THE 3 LIGHTS
 2962                           ;Z%%    OUT     LIGHTS  ;***** I/O WRT 101/7 *****
 2963                           ;Z%%    CALL    LTDLY   ;LEAVE LIGHTS ON FOR ABOUT A SECOND
 2964                           ;Z%%    XRA     A       ;CLEAR ACCUM
 2965                           ;Z%%    OUT     LIGHTS  ;***** I/O WRT 101/0 *****
 2966                           ;FALL INTO CODE THAT WAITS A WHILE WITH THE LIGHTS OFF
 2967
 2968                           ;Z%%LTDLY:      LXI     H,300   ;DELAY ABOUT A SECOND AND A HALF
 2969  006134  315  265  033   LTLOOP: CALL    DELAY.  ;GO DO A LITTLE DELAY
 2970  006137  377                   .BYTE   -1      ;MAX COUNT
 2971
 2972  006140  053                   DCX     H       ;DOWN THE COUNT
 2973  006141  175                   MOV     A,L     ;GET PIECE OF THE COUNT
 2974  006142  264                   ORA     H       ;THROW IN THE REST OF THE COUNT
 2975  006143  302  134  014         JNZ     LTLOOP  ;CONTINUE WAITING
 2976  006146  311                   RET             ; UNTIL ALL DONE
 2977
 2978                           .SBTTL  *** 'MM' CMD ***
 2979                           ;COMMAND TO PUT THE 8080 INTO MANUFACTURING MODE.
 2980                           ;SETS THE STATE FOR THE KLINIK LINE THEN SENDS A COMMUNICATIONS CLEAR
 2981                           ;TO WHATEVER IS AT THE OTHER END OF THE KLINIK LINE
 2982  006147  315  050  034   MMCMD: CALL    SETM4   ;SET KLINIK LINE TO MODE 4
 2983  006152  076  041               MVI     A,^041  ;WE MUST ALWAYS RESET THE MESSAGE NUMBERS
 2984  006154  062  361  040         STA     LSTMSG  ;THIS IS THE 'RECEIVE' MESSAGE NUMBER
 2985  006157  062  011  042         STA     ENVMNO  ;AND THIS IS THE 'SEND' MESSAGE NUMBER
 2986  006162  062  251  040         STA     MMFLG   ;SAY MANUFACTURING MODE HAS BEEN ENTERED
 2987  006165  315  376  035         CALL    Z.TBUF  ;CLEAR SOME COMMUNICATION DEC10 BUFFERS
 2988  006170  303  336  034         JMP     DECEX2  ;CLEAR THE MAILING ENVELOPES
 2989                                                 ;**USING JMP USES OTHER GUY'S RETURN TO RETURN
 2990
```

```
 2992                           .SBTTL  *** 'SI' CMD ***
 2993                           ;COMMAND TO CAUSE SM10 TO EXECUTE A SINGLE INSTR.
 2994  006173  333  300        SICMD:  IN      RUNFP   ;BEFORE CONTINUING,MUST READ MACHINE STATE
 2995  006175  346  004                ANI     4       ;IS THE RUN FLOP SET(IS IT ALREADY RUNNING??)(TRUE LO)
 2996  006177  312  027  027          JZ      YSRUN   ;IF YES, GO PRINT A MESSAGE TO THAT EFFECT & ABORT CMD
 2997
 2998  006202  076  001                MVI     A,01    ;SET BIT FOR 'CONTINUE'
 2999  006204  323  212                OUT     CPUCTL  ;***** I/O WRT 212/1 *****
 3000  006206  315  061  017          CALL    DNF     ;CHECK THAT INSTR FINISHED
 3001  006211  303  336  016          JMP     PCCOM   ;AND GO TO TYPE OUT THE PC
 3002
 3003                           .SBTTL  *** 'CS' CMD ***
 3004                           ;COMMAND TO START THE SM10 CPU CLK RUNNING
 3005  006214  315  255  032   CSCMD:  CALL    SETRN   ;SET CLK 'RUNNING' FLAG
 3006  006217  257                    XRA     A       ;CLR ACCUM TO CLR 'SS MODE'
 3007  006220  323  204                OUT     CRMCTL  ;***** I/O WRT 204/0 *****
 3008  006222  076  003                MVI     A,03    ;SET BITS FOR 'CLK RUN' & 'SINGLE CLK'
 3009  006224  323  206                OUT     CLKCTL  ;***** I/O WRT 206/3 *****
 3010  006226                         ENDCMD          ;DONE..
  (1)  006226  311                    RET             ;RETURN TO CALLER
 3011
 3012                           .SBTTL  *** 'CH' CMD ***
 3013                           ;COMMAND TO HALT THE SM10 CPU CLK
 3014  006227  315  271  032   CHCMD:  CALL    CLRRN   ;CLEAR CLK 'RUNNING' FLAG
 3015  006232  076  010                MVI     A,^O010 ;SET BIT FOR 'SS MODE'
 3016  006234  323  204                OUT     CRMCTL  ;***** I/O WRT 204/010 *****
 3017  006236  257                    XRA     A       ;CLR BITS FOR 'SINGLE CLK' & 'CLK RUN'
 3018  006237  323  206                OUT     CLKCTL  ;***** I/O WRT 206/0 *****
 3019  006241                         ENDCMD          ;DONE..
  (1)  006241  311                    RET             ;RETURN TO CALLER
 3020
 3021
 3022                           .SBTTL  *** 'LF' CMD ***
 3023                           ;COMMAND TO 'LOAD FUNCTION'..SPECIFIES WHICH DIAG FUNCTION WRITE
 3024                           ;TO DO ON THE NEXT 'DF' COMMANDS
 3025  006242  315  244  030   LFCMD:  CALL    ARG16.  ;GO ASSEMBLE 16 BIT ARG(WE ONLY NEED 4 BITS)
 3026
 3027  006245  042  010  040          SHLD    CRMFN   ;PERMANENT HOME FOR DATA
 3028  006250                         ENDCMD          ;DONE..
  (1)  006250  311                    RET             ;RETURN TO CALLER
```

```
                                ;SBTTL  *** 'DF' CMD ***
                                ;ROUTINE WRITES THE DATA TYPED USING THE DIAG FUNCTION
                                ;PREVIOUSLY SPECIFIED BY LF COMMAND
006251                          DFCMD:  RUN.,           ;IS CPU RUNNING??
(1) 006251  347                         RST     4
(1) 006252  006                         .BYTE   6

006253  315  244  030                   CALL    ARG16.  ;GO ASSEMBLE ARG

                                ;NEXT ROUTINE DOES LOTS OF I/O WRTS TO SM10 CPU
                                ;ALL WHILE TRYING TO WRITE DIAGNOSTIC ADDRESS REG FOR CRAM
                                ;LOADING OR READING..
006256  345                             PUSH    H       ;SAVE DATA TO BE DEPOSITED
006257  315  315  014                   CALL    CRM.AD  ;WRITE THE CRAM ADDRESS
006262  341                             POP     H       ;GET DATA TO BE DEPOSITED

006263  175                   WFUNC:    MOV     A,L     ;GET DATA FOR BITS 28-35 INTO ACCUM
006264  323  103                       OUT     A2835   ;***** I/O WRT 103 *****
006266  174                             MOV     A,H     ;GET DATA FOR BITS 20-27
006267  323  105                       OUT     A2027   ;***** I/O WRT 105 *****

006271  257                   WFNC1:    XRA     A       ;CLR ACCUM
006272  323  115                       OUT     BUSARB  ;***** I/O WRT 115/0 *****

006274  076  144                       MVI     A,^O144 ;BITS FOR 'CONS REQ','T ENB FOR COM/ADR','CRA R CLK'
006276  323  210                       OUT     BUSCTL  ;***** I/O WRT 210/144 *****

006300  072  010  040                   LDA     CRMFN   ;GET DIAG FUNCTION
006303  323  205                       OUT     DIAG    ;***** I/O WRT 205/FNC *****
                                ;NOTE THAT 'TRAP EN' WAS JUST ZAPPED, BUT IT IS ONLY USEFUL IF THE
                                ;MICRO-CODE IS RUNNING AND ANYTHING U DO TO GET THE MICRO-CODE RUNNING
                                ;WILL RESTORE THE TRAP ENABLE..THIS KLUDGE SPEEDS UP MICRO-CODE LOAD


006305  076  040                       MVI     A,^040  ;BIT FOR 'CRAM WRT'
006307  323  204                       OUT     CRMCTL  ;***** I/O WRT 204/40
006311  257                             XRA     A       ;BIT TO CLR 'CRAM WRT'
006312  323  204                       OUT     CRMCTL  ;***** I/O WRT 204/0 *****
006314                                  ENDCMD          ;DONE..
(1) 006314  311                         RET             ;RETURN TO CALLER

                                ;SIMPLE LITTLE ROUTINE TO SAVE SOME SPACE..USED IN SEVERAL PLACES
006315  052  004  040           CRM.AD: LHLD    CRMAD   ;LOAD DIAG ADDR TO BE WRITTEN
```

```
 3071                          ;ROUTINE COUNTS ON DATA IN 'H,L',,DESTROYS 'H,L'...
 3072  006320  076  001        CADWR:  MVI     A,01    ;BIT FOR CRAM RESET
 3073  006322  323  204                OUT     CRMCTL  ;***** I/O WRT 204/1 *****
 3074  006324  257                    XRA     A       ;CLR BIT TO CLR CRAM RESET
 3075  006325  323  204                OUT     CRMCTL  ;***** I/O WRT 204/0 *****
 3076
 3077                          ;***** I/O WRT 103,105,107,111,113 *****
 3078  006327  175                    MOV     A,L     ;LO ORDER 8 BITS TO ACCUM
 3079  006330  323  103                OUT     A2835   ;SET IN HARDWARE REG
 3080  006332  174                    MOV     A,H     ;HI ORDER 4 BITS TO ACCUM
 3081  006333  323  105                OUT     A2027   ;SET INTO HARDWARE REG
 3082
 3083  006335  257                    XRA     A       ;CLR ACCUM
 3084  006336  323  107                OUT     A1219   ;CLR OTHER HARDWARE REGS
 3085  006340  323  111                OUT     A0411
 3086  006342  323  113                OUT     A0003
 3087
 3088  006344  323  115                OUT     BUSARB  ;***** I/O WRT 115/0 *****
 3089
 3090  006346  076  144                MVI     A,^O144 ;BITS FOR 'CONS REQ', 'T ENB FOR COM/ADR','CRA R CLK'
 3091  006350  323  210                OUT     BUSCTL  ;***** I/O WRT 210/144 *****
 3092  006352  076  021                MVI     A,^O21  ;BIT FOR 'CRM ADDR LOAD'
 3093  006354  323  204                OUT     CRMCTL  ;***** I/O WRT 204/21
 3094  006356  257                    XRA     A       ;BIT TO CLR CRAM ADDR LOAD
 3095  006357  323  204                OUT     CRMCTL  ;***** I/O WRT 204/0 *****
 3096  006361  311                    RET             ;AND RETURN
 3097
 3098
 3099                          ;ROUTINE TO READ A SINGLE DIAG FUNC WORTH OF STUFF FROM
 3100                          ;THE CRA/CRM PROCESSOR BOARDS
 3101  006362  127            READC:  MOV     D,A     ;SAVE DIAG FUNC FOR A SEC.,
 3102  006363  072  355  040          LDA     TRAPEN  ;GET CURRENT VALUE FOR TRAP ENABLES
 3103  006366  262                    ORA     D       ;MIX TOGETHER
 3104  006367  323  205                OUT     DIAG    ;***** I/O WRT 205/FNC *****
 3105  006371  076  115                MVI     A,^O115 ;BITS 'CONS REQ','CRA T CLK','R CLK ENB','CRA R CLK'
 3106  006373  323  210                OUT     BUSCTL  ;***** I/O WRT 210/115 *****
 3107
 3108  006375  333  000                IN      D2835   ;***** I/O RD 0 *****
 3109  006377  057                    CMA             ;FIX INVERSION
 3110  006400  062  100  040          STA     TMPB2   ;SAVE IN STANDARD BUFFER
 3111
 3112  006403  333  001                IN      D2027   ;***** I/O RD 1 *****
 3113  006405  057                    CMA             ;FIX INVERSION
 3114  006406  346  017                ANI     ^O17    ;KEEP ONLY 12-8
 3115  006410  062  101  040          STA     TMPB2+1 ;SAVE IN STANDARD BUFFER
 3116
 3117  006413  257                    XRA     A       ;CLR ACCUM
 3118  006414  323  210                OUT     BUSCTL  ;***** I/O WRT 210/0 *****
 3119
 3120  006416  311                    RET             ;RETURN
```

```
 3122                           .SBTTL  *** 'RC' ***
 3123  006417                   RCCMD:  RUN..           ;IS CPU RUNNING??
  (1)  006417  347                       RST     4
  (1)  006420  006                       .BYTE   6
 3124
 3125  006421  257              RCINT:  XRA     A       ;CLEAR ACCUM FOR USE AS A COUNTER
 3126  006422  001  341  040             LXI     B,CRMBF+^D31 ;PNTR TO A BUFFER AREA TO SAVE THE 'RC'S' AS READ
 3127  006425  137              RCLP:   MOV     E,A     ;SAVE IN 'E' REG
 3128  006426  315  362  014             CALL    READC   ;READ A DIAG FUNC FROM CRA/CRM BRD
 3129                           ;NOW PRINT WHAT WAS READ
 3130  006431  072  120  040             LDA     NOPNT   ;WE WILL MAKE IT QUICKER IF NOT PRINTING RESULTS
 3131  006434  247                       ANA     A       ;SET FLAGS
 3132  006435  302  062  015             JNZ     RCNOP   ;IF NO PRINT, AVOID TYPING CODE
 3133
 3134  006440  173                       MOV     A,E     ;PUT IN MEM FOR PRINT ROUTINE
 3135  006441  315  304  030             CALL    P8BITA  ;PRINT NAME OF THIS DIAG FUNC
 3136  006444                           PSLASH          ;AND '/'
  (2)  006444  317                       RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (2)  006445  057                       .BYTE   SLASH   ;CHAR TO PRINT
 3137  006446  315  314  030             CALL    P16.    ;AND PRINT IT
 3138  006451                           PCRLF
  (2)  006451  347                       RST     4
  (2)  006452  002                       .BYTE   2
 3139
 3140  006453  305                       PUSH    B       ;SAVE COUPLE REGS WHILE GO DECNET
 3141  006454  325                       PUSH    D
 3142  006455  315  205  034             CALL    DECNET  ;YES.. SEND THIS GROUP OF DATA DOWN THE KLINIK LINE
 3143  006460  321                       POP     D       ;RETRIEVE THOSE REGISTERS
 3144  006461  301                       POP     B
 3145
 3146                           ;CODE FOR SAVING THE RESULTS OF THESE FUNCTION READS IN THE 8080 RAM SPACE
 3147                           ;FOR NOW WE WILL SAVE THE  RESULTS IN THE PLACE WHERE CRAM DATA IS KEPT
 3148  006462  052  100  040   RCNOP:  LHLD    TMPB2   ;FETCH UP THE DATA THAT WAS ACTUALLY READ
 3149  006465  174                       MOV     A,H     ;GET LO ORDER PIECE TO ACCUM
 3150  006466  002                       STAX    B       ;STORE TO PLACE POINTED TO BY 'D,E'
 3151  006467  013                       DCX     B       ;UPDATE THE STORAGE POINTER
 3152  006470  175                       MOV     A,L     ;GET HI ORDER PIECE OF CRAM DATA
 3153  006471  002                       STAX    B       ;SAVE IN STORAGE AREA
 3154  006472  013                       DCX     B       ;AGAIN DOWNDATE POINTER TO BEGINING OF ACTUAL
 3155
 3156  006473  034                       INR     E       ;INCREMENT IT
 3157  006474  173                       MOV     A,E     ;COPY CURRENT COUNT TO ACCUM FOR THE COMPARE
 3158  006475  376  020                 CPI     ^O20    ;REACHED MAX YET??
 3159  006477  302  025  015             JNZ     RCLP    ;BACK IF NOT YET..
 3160
 3161                           ;OTHERWISE
 3162  006502                           ENDCMD          ;DONE...
  (1)  006502  311                       RET             ;RETURN TO CALLER
```

```
 3164                           .SBTTL  *** 'EJ' CMD ***
 3165                           ;CONSOLE COMMAND TO DISPLAY THE FLOW OF THE CONTROL STORE BY PRINTING
 3166                           ;OUT THE CURRENT 'J-FIELD', 'NEXT LOC', 'SUBROUTINE RET REG', & 'CURRENT
 3167                           ; LOCATION'
 3168  006503                   EJCMD:  RUN..           ;IS CPU RUNNING??
  (1)  006503  347                     RST     4
  (1)  006504  006                     .BYTE   6
 3169  006505  041  150  015           LXI     H,EJLST ;FIRST GET A PNTR TO ASCII TEXT
 3170  006510  001  207  004           LXI     B,^B10010000111 ;SET B=4 & C='10,00,01,11'
 3171
 3172  006513  171             EJLP:   MOV     A,C     ;COPY DIAG FUNC STRING TO ACCUM
 3173  006514  346  003                 ANI     3       ;STRIP ALL BUT LO ORDER 2 BITS
 3174
 3175  006516  315  362  014   EJ1:    CALL    READC   ;GO READ DIAG FUNC AS GIVEN BY ACCUM
 3176
 3177  006521  315  160  004           CALL    PLN1    ;PRINT ASCII IDENTIFIER FOR THIS FUNC
 3178
 3179  006524  345                     PUSH    H       ;SAVE 'H,L'
 3180  006525  315  314  030           CALL    P16,    ;AND GO PRINT IT AS 16 BIT OCTAL
 3181  006530                         PCHAR   SPACE
  (1)  006530  317                     RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (1)  006531  040                     .BYTE   SPACE   ;CHAR TO PRINT
 3182  006532                         PCHAR   SPACE
  (1)  006532  317                     RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (1)  006533  040                     .BYTE   SPACE   ;CHAR TO PRINT
 3183  006534  341                     POP     H       ;GET 'H,L' BACK
 3184
 3185  006535  171                     MOV     A,C     ;GET FUNCTION PICKER
 3186  006536  017                     RRC             ;SHIFT FUNCTION LIST
 3187  006537  017                     RRC             ; 2 PLACES
 3188  006540  117                     MOV     C,A     ;PUT BACK FUNCTION
 3189
 3190  006541  005                     DCR     B       ;NOW DOWN THE COUNTER
 3191  006542  302  113  015           JNZ     EJLP    ;AND JUMP TO THE EXECUTING CODE
 3192
 3193  006545                         PCRLF           ;AND A CR-LF
  (2)  006545  347                     RST     4
  (2)  006546  002                     .BYTE   2
 3194                           ;ELSE,, END OF COMMAND
 3195  006547                         ENDCMD          ;RETURN TO CALLER
  (1)  006547  311                     RET             ;RETURN TO CALLER
 3196
 3197  006550  103  125  122   EJLST:  .ASCIZ %CUR/%   ;FUNC 03 IS CURRENT CRAM LOCATION
 3198  006555  116  130  124           .ASCIZ %NXT/%   ;FUNC 01 IS NEXT LOC
 3199  006562  112  057  000           .ASCIZ %J/%     ;FUNC 00 IS J-FIELD
 3200  006565  123  125  102           .ASCIZ %SUB/%   ;FUNC 02 IS SUBROUTINE RETURN REG
```

```
                                .SBTTL  *** 'TR' CMD ***
                                ;THIS CONSOLE COMMAND TRACES THE FLOW OF THE MICRO-CODE BY TYPEING
                                ;THE 4 KNOWN ADDRESSES FROM THE CONTROL RAM ADDRESS BRD, THEN ISSUEING
                                ;A SINGLE CPU PULSE AND CONTINUING THIS UNTIL THE USER TYPES A
                                ;CARRIAGE RETURN.
006572  332  206  015   TRCMD:  JC      TR1     ;IF NO ARG, GO LIKE NORMAL
006575                         ARG16           ;IF WAS ARG, GO GET IT
 (1)  006575  347               RST     4
 (1)  006576  004               .BYTE   4
006577  152  040               .ADDR   BRKDT   ;PLACE TO PUT IT

006601  076  077               MVI     A,^077  ;NOW ANY ARBITRARY,NON-ZERO VALUE
006603  062  151  040           STA     BRKON   ;TO SAY THAT BREAKING IS ON..

006606                 TR1:    RUN..           ;IS CPU RUNNING??
 (1)  006606  347               RST     4
 (1)  006607  006               .BYTE   6
006610                         CLRB    RPEND   ;SO CAN CLR CMD CNTR
006612  072  151  040   TR:     LDA     BRKON   ;CHECK IF BREAK IS ON
006615  247                    ANA     A       ;CHECK FLAG
006616  312  230  015           JZ      TRLP    ;IF ZERO,DONT LOOK AT BREAK STUFF
006621  021  152  040           LXI     D,BRKDT ;PASS PNTR TO THE DESIRED STOPPING ADDRESS
006624  315  171  030           CALL    BREAK   ;IF FLAG SET, CALL TO CHECK ADDRESS
006627  310                    RZ              ;IF RETURN WITH Z-SET, WE ARE AT BREAK PLACE

006630  315  251  015   TRLP:   CALL    PULSE   ;GIVE PULSE
006633                         PCRLF           ;CARRIAGE RETURN LINE FEED
 (2)  006633  347               RST     4
 (2)  006634  002               .BYTE   2

006635  072  157  040           LDA     RPEND   ;GET CMD CNTR
006640  247                    ANA     A       ;IS IT SET??
006641  312  212  015           JZ      TR      ;WELL, CONT LOOP IF NOT YET
                               ;OTHERWISE, END THE COMMAND
006644                         CLRB    BRKON   ;AND CLR THE FLAG
006646                         ENDCMD          ;DONE..
 (1)  006646  311               RET             ;RETURN TO CALLER
```

```
                        .SBTTL  *** 'PM' CMD ***
                        ;CONSOLE COMMAND TO 'PULSE' 'MICRO-CODE',..IE GIVE
                        ;A SINGLE PULSE AND THEN AN 'EJ' COMMAND..
                        ;COMMAND IS EQUIVILANT TO THE 'TR' TRACE COMMAND,
                        ;ONLY EXECUTING THE TRACE ONCE..
006647                  PMCMD:  RUN..           ;IS CLK RUNNING??
(1) 006647  347                 RST     4
(1) 006650  006                 .BYTE   6

006651  315  204  013   PULSE:  CALL    CP1     ;GO DO A SINGLE CLOCK
006654  315  103  015           CALL    EJCMD   ;TYPE CONTROL STORE ADDRESSES & EXIT FROM THERE
006657                          ENDCMD          ;AND OUT
(1) 006657  311                 RET             ;RETURN TO CALLER
```

```
 3245                           .SBTTL  *** 'EC' CMD ***
 3246                           ;ROUTINE TO READ THE C-RAM AND TYPE IT OUT
 3247  006660                   ECCMD:  RUN.,           ;IS CPU RUNNING??
  (1)  006660  347                       RST     4
  (1)  006661  006                       .BYTE   6
 3248  006662  332  302  015            JC      EC2     ;IF NO ARG, DONT GO ASSEMBLE ONE
 3249  006665  315  331  013            CALL    LCCMD   ;FETCH UP DESIRED CRAM ADDRESS
 3250
 3251  006670                   EC1:    CLRRM   TMPB2   ;ZAP A TEMPORARY BUFFER
  (1)  006670  347                       RST     4
  (1)  006671  012                       .BYTE   10.
  (1)  006672  105  040                  .ADDR   TMPB2+5
 3252
 3253  006674  315  315  014            CALL    CRM.AD  ;NOW WRITE DESIRED CRAM ADDRESS
 3254  006677  315  204  013            CALL    CP1     ;AND GIVE A SINGLE CLK PULSE TO LOAD CNTRL REG
 3255
 3256  006702  076  006        EC2:    MVI     A,06    ;SET UP 'EXAMINE NEXT' TYPE COMMANDS
 3257  006704  062  031  040            STA     ENEXT   ;SAVE EXAMINE STUFF IN RAM
 3258
 3259                           ;NOW READY TO READ THE CONTROL REG
 3260  006707  041  041  016            LXI     H,RDLST ;GET PNTR TO DIAG FUNCTIONS TO BE READ
 3261
 3262  006712  176              ECLP:   MOV     A,M     ;GET DIAG FUNCTION TO ACCUM
 3263  006713  043                      INX     H       ;UPDATE PNTR
 3264  006714  247                      ANA     A       ;WAS FNC END-OF-LIST(YES IF WAS MINUS)
 3265  006715  372  344  015            JM      ECBEE   ;JMP IF WAS END OF LIST
 3266
 3267                           ;OTHERWISE, WE MUST DO A DIAG FUNCTION
 3268  006720  315  362  014            CALL    READC   ;GO READ THIS DIAG FUNC,DATA RETURNED IN 'TMPB2'
 3269  006723  042  170  040            SHLD    ECSAV   ;NOW SAVE 'H,L' FOR A MINUTE
 3270
 3271  006726  041  100  040            LXI     H,TMPB2 ;POINTER TO DATA JUST READ
 3272  006731  315  032  031            CALL    OCTAL   ;NOW TURN DATA INTO ASCII OCTAL CHARS
 3273  006734  002                      .BYTE   2       ;TWO BYTES RELEVANT DATA
 3274  006735  004                      .BYTE   4       ;WANT 4 OCTAL CHARS
 3275
 3276  006736  052  170  040            LHLD    ECSAV   ;RESTORE THE 'H,L'
 3277  006741  303  312  015            JMP     ECLP    ;AND CONTINUE TILL READ ALL DIAG FUNCS
```

```
 3279                           ;WHEN U GET TO HERE, YOU'VE READ ALL FUNCS, NOW READ & CMP A & B COPIES
 3280  006744  176              ECBEE:  MOV     A,M     ;GET DIAG FUNC FOR AN 'A' COPY
 3281  006745  043                      INX     H       ;UPDATE MEM PNTR
 3282  006746  247                      ANA     A       ;DID DIAG FUNC HAVE MINUS SIGN??
 3283  006747  372  012  016           JM      PCRAM   ;DONE LIST, JMP IF WAS MINUS
 3284                           ;OTHERWISE, GO AND READ THE 'A' COPY AGAIN
 3285  006752  315  362  014           CALL    READC   ;DATA RETURNED IN 'TMPB2'
 3286  006755                         MOV5B           ;MOVE THAT DATA TO 2ND TMP BUFF
  (1)  006755  347                    RST     4
  (1)  006756  000                    .BYTE   0
 3287  006757  100  040               .ADDR   TMPB2   ;SRC OF DATA
 3288  006761  073  040               .ADDR   TMPBF2  ; PLACE TO PUT IT
 3289
 3290  006763  176                    MOV     A,M     ;GET DIAG FUNC FOR A 'B' COPY
 3291  006764  043                    INX     H       ;UPDATE PNTR
 3292  006765  315  362  014           CALL    READC   ;NOW READ A 'B' COPY
 3293
 3294  006770  345                    PUSH    H       ;SAVE 'H,L'
 3295  006771  315  022  033           CALL    CMP36   ;NOW COMPARE THE 'A' AND 'B' COPIES
 3296  006774  100  040               .ADDR   TMPB2   ;'B' COPY
 3297  006776  073  040               .ADDR   TMPBF2  ;'A' COPY
 3298  007000  341                    POP     H       ;RESTORE 'H,L'
 3299  007001  312  344  015           JZ      ECBEE   ;IF CHECKED OK, BACK TO READ NEXT 'A/B' COPIES
 3300
 3301                           ;FALL THRU TO VERIFY ERROR IF 'Z' NOT SET
 3302  007004                         PLINE   ECVER   ;'?VERIFY ERR'
  (1)  007004  337                    RST     3       ;PRINT LINE OF CHARS
  (1)  007005  104  037               .ADDR   ECVER   ;BUFF TO PRINT
 3303  007007  303  021  015           JMP     RCINT   ;GO PRINT ALL CRAM REGS
 3304
```

```
 3306                           ;IF 'A/B' COPIES VERIFIED, TIME TO PRINT C-RAM CONTENTS
 3307  007012  076  003         PCRAM:  MVI     A,03    ;DIAG FUNC TO READ 'CURRENT CRAM LOCATION'
 3308  007014  315  362  014           CALL    READC   ;GO READ CURRENT CRAM LOC..
 3309  007017  315  314  030           CALL    P16.    ;PRINT IT
 3310  007022                         PSLASH          ; AND '/'
  (2)  007022  317                     RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (2)  007023  057                     .BYTE   SLASH   ;CHAR TO PRINT
 3311
 3312                           ;NOW PRINT THE 32 OCTAL CHARS.......
 3313  007024  006  040                 MVI     B,32.   ;NUM CHARS TO PRINT
 3314  007026  361               PCRLP:  POP     PSW     ;GET A CHAR
 3315  007027  315  010  004           CALL    PCHR    ;PRINT IT
 3316  007032  005                     DCR     B       ;DOWN COUNT OF CHARS PRINTED
 3317  007033  302  026  016           JNZ     PCRLP   ;LOOP TILL DONE
 3318
 3319  007036                         PCRLF           ;NEED CR-LF
  (2)  007036  347                     RST     4
  (2)  007037  002                     .BYTE   2
 3320
 3321  007040                         ENDCMD          ;THEN OUT
  (1)  007040  311                     RET             ;RETURN TO CALLER
 3322  007041  017               RDLST:  .BYTE   ^O17    ;READ 84-95
 3323  007042  016                     .BYTE   ^O16    ;READ 72-83
 3324  007043  015                     .BYTE   ^O15    ;READ 60-71
 3325  007044  014                     .BYTE   ^O14    ;READ 48-59
 3326  007045  012                     .BYTE   ^O12    ;READ 36-47A
 3327  007046  005                     .BYTE   ^O5     ;READ 24-35A
 3328  007047  004                     .BYTE   ^O4     ;READ 12-23
 3329  007050  000                     .BYTE   0       ;READ 0-11
 3330  007051  377                     .BYTE   ^O377   ;END BYTE
 3331  007052  012                     .BYTE   ^O12    ;READ 36-47A
 3332  007053  013                     .BYTE   ^O13    ;READ 36-47B
 3333  007054  005                     .BYTE   ^O5     ;READ 24-35A
 3334  007055  006                     .BYTE   ^O6     ;READ 24-35B
 3335  007056  377                     .BYTE   ^O377   ;END BYTE
 3336
 3337                           .SBTTL  *** 'EN' CMD ***
 3338  007057  052  004  040   ENEC:   LHLD    CRMAD   ;GET CURRENT ADDRESS
 3339  007062  043                     INX     H       ;UPDATE IT
 3340  007063  042  004  040           SHLD    CRMAD   ;PUT IT BACK
 3341  007066  303  270  015           JMP     EC1     ;GO TO COMMON CODE
```

```
 3343                           .SBTTL  *** 'DC' CMD ***
 3344                           .SBTTL  *** 'DN TO DC' CMD ***
 3345                           ;CODE USED IN DEPOSIT NEXT FOR THE CRAM
 3346  007071  052  004  040   DNDC:  LHLD    CRMAD   ;GET CURRENT ADDRESS
 3347  007074  043                  INX     H       ;INCREMENT IT
 3348  007075  042  004  040         SHLD    CRMAD   ;PUT IT BACK
 3349
 3350  007100                 DCCMD:  RUN..           ;IS CPU RUNNING??
  (1)  007100  347                  RST     4
  (1)  007101  006                  .BYTE   6
 3351  007102  315  227  031         CALL    ARG96   ;ASSEMBLE DATA TO DEPOSIT
 3352  007105  322  040              .ADDR   CRMTM   ;PLACE TO PUT IT
 3353
 3354  007107  021  302  040         LXI     D,CRMBF ;PLACE TO PUT THE RESULTS OF THE CRAM SHUFFLE
 3355  007112  041  322  040         LXI     H,CRMTM ;THE OLD 12-BYTE FORMAT WILL ALWAYS BE HERE
 3356  007115  016  004              MVI     C,4     ;LOAD 'C' WITH A 4
 3357
 3358                           ;BEGIN THE UNPACKING
 3359  007117  315  062  032  GENLP:  CALL    PLACE   ;LOCAL ROUTINE THAT TAKES 12 BITS OF 24.
 3360  007122  076  003              MVI     A,3     ;A SHIFT 24 REQUIRES 3 BYTES OF DATA TO SHIFT
 3361  007124  315  156  031         CALL    SHR24   ;SHIFT 12 BITS JUST PACKED INTO OUTER SPACE
 3362  007127  014                  .BYTE   12.     ;TELL ROUTINE 12 PLACES
 3363  007130  315  062  032         CALL    PLACE   ;NOW ROUTINE WILL GET 12 MORE BITS..12+12=24
 3364
 3365  007133  043                  INX     H       ;UPDATE PNTR 3-BYTES(IE 24 BITS)
 3366  007134  043                  INX     H
 3367  007135  043                  INX     H
 3368
 3369  007136  015                  DCR     C       ;DOWN THE COUNTER(THERE ARE 4 GROUPS OF 24=96)
 3370  007137  302  117  016         JNZ     GENLP   ;CONTINUE TILL DONW THE 4 GROUPS
 3371
 3372  007142  315  315  014         CALL    CRM.AD  ;WRITE THE CRAM ADDRESS
 3373  007145  041  302  040         LXI     H,CRMBF ;GET PLACE WHERE INFO WAS JUST PLACED
 3374
 3375  007150  076  006              MVI     A,06    ;NUMBER FOR DEPOSIT NEXT TO USE
 3376  007152  062  033  040         STA     DNEXT   ;STANDARD PLACE TO KEEP IT
 3377  007155  074                  INR     A       ;SET FUNCTION .EQ. 7(INR WORKS BY LUCK)
 3378  007156  001  010  040         LXI     B,CRMFN ;SET AN ADDRESS INTO 'B,C' REGISTER,TO USE AS A POINTER
 3379  007161  002           DCLP:  STAX    B       ;SAVE IT IN THE RAM AT LOC "CRMFN"
 3380
 3381  007162  136                  MOV     E,M     ;GET 8 BITS OF DATA
 3382  007163  043                  INX     H       ;UPDATE PNTR
 3383  007164  126                  MOV     D,M     ;GET 4 MORE BITS OF DATA
 3384  007165  043                  INX     H       ;AND UPDATE PNTR AGAIN
 3385
 3386  007166  353                  XCHG            ;NOW "H,L" CONTAINS THE DATA & "D,E" THE PNTR
 3387  007167  315  263  014         CALL    WFUNC   ;AND DIAG FUNCTION WRT
 3388  007172  353                  XCHG            ;POINTER BACK TO "H,L"
 3389
 3390  007173  012                  LDAX    B       ;GET PARTICULAR DIAG FUNC FROM RAM LOC
 3391  007174  075                  DCR     A       ;DOWN TO NEXT
 3392  007175  362  161  016         JP      DCLP    ;AS LONG AS 0-7, KEEP GOING
 3393  007200                       ENDCMD          ;NOW ALL DONE
  (1)  007200  311                  RET             ;RETURN TO CALLER
```

```
                                ,SBTTL  *** 'SM' CMD ***
                                ;CODE TO START THE MICRO-CODE AT THE ADDRESS SPECIFIED,,
                                ;DEFAULTS TO STARTING AT C-RAM LOC 0 IF NO ADDRESS IS
                                ;GIVEN
007201  332  212  016  SMCMD:  JC      SM1     ;IF NO ARG, SUPPLY ADDRESS OF 0000

                                ;OTHERWISE MUST ASSEMBLE THE GIVEN ADDRESS
007204  315  244  030          CALL    ARG16.  ;ASSEMBLE 16-BITS OF ARGUMENT

007207  303  215  016          JMP     SM1.5   ;OTHERWISE, CONTINUE NORMALLY

007212  041  000  000  SM1:    LXI     H,00    ;IF HERE, DESIRE ADDRESS OF 0000
007215  042  000  040  SM1.5:  SHLD    T80DT   ;SET ADDR

007220  315  225  005          CALL    MRCMD   ;RESET THE MACHINE

007223                         MOV5B           ;SET UP INITIAL DATA
 (1)  007223  347               RST     4
 (1)  007224  000               .BYTE   0
007225  256  022               .ADDR   ONES    ;DATA TO BE IS ALL ONES
007227  047  040               .ADDR   DMDAT   ;PLACE WHERE IT GOES

007231  021  364  016          LXI     D,MAD000 ;GET ADDRESS OF MEM LOC 0
007234  315  237  012          CALL    DMINT   ;'DEPOSIT MEMORY' INTERNAL FORMAT

007237  072  354  040          LDA     PARBT   ;GET PARITY STUFF
007242  346  140               ANI     ^O140   ;ONLY KEEP A LITTLE BIT
007244  323  100               OUT RESET       ;AND TURN OF ALL PARITY STUFF WHILE WE DO THIS

007246  052  000  040          LHLD    T80DT   ;GET START ADDRESS OF MICRO-CODE TO 'H,L'
007251  315  320  014          CALL    CADWR   ;WRITE THE DIAG ADDRESS REG

007254  315  214  014          CALL    CSCMD   ;START THE CPU CLK FREE RUN
```

```
 3427  007257  315  265  033  HLTCM:  CALL    DELAY,  ;NOW WAIT FOR MICRO-CODE TO REACH HALT LOOP
 3428  007262  377                   ,BYTE   -1
 3429  007263  315  346  006         CALL    CLRUSE  ;EXIT FROM USER MODE
 3430  007266  333  300               IN      RUNFP   ;***** I/O RD 300 *****
 3431  007270  057                   CMA             ;AND FIX INVERSION
 3432  007271  346  010               ANI     ^O10    ;IS CPU IN THE HALT LOOP???
 3433  007273  302  305  016         JNZ     SMVER   ;JUMP IF YES,,APPEARED TO START OK
 3434
 3435                         ;FALL TO HERE IF SM10 DID NOT SET HALT LOOP FLAG
 3436  007276                         PLINE   SMERR   ;PRINT ERR MESSAGE
  (1)  007276  337                   RST     3       ;PRINT LINE OF CHARS
  (1)  007277  146  037               ,ADDR   SMERR   ;BUFF TO PRINT
 3437  007301  067                   STC             ;SET C-BIT TO INDICATE AN ERROR EXIT
 3438  007302  303  356  016         JMP     SMFINI  ;AND EXIT VIA RESTORE PARITY PATH
```

```
 3440  007305                 SMVER:  INTON           ;SET INTERAL STATUS FOR THE EXAMINE
  (1)  007305  327                      RST     2       ;GO SET INTERNAL MODE
 3441  007306                         EXAM    0       ;EXAMINE MEM LOC 0(MICRO-CODE STOP CODE)
  (1)  007306  315  216  030            CALL    EXAMSH  ;AND DO EXAM ASSUMING SHORT ADDRESS
  (1)  007311  000  000                 .ADDR   0       ;ADDR TO BE ZAPPED PASSED AS TRAILING ARG
 3442  007313                         INTOFF          ;TURN OFF INTERNAL STATUS
  (1)  007313  367                      RST     6       ;GO EXIT FROM INTERNAL MODE
 3443                           ;IT DID SUCCEED IN SETTING HALT LOOP FLAG, SO MERELY PRINT HALTED
 3444                           ;AND THE DATA IN LOCATION 0..
 3445                           ;****SUBROUTINE 'STOP CODE' ****
 3446  007314  315  255  032            CALL    SETRN   ;JUST A LITTLE KLUDGE..CHEAP WAY TO FIX STATE LIGHT
 3447                                                   ; IF PROGRAM EXECUTED A 'HALT' WHILE LITES WERE BLINKY
 3448
 3449  007317                         PLINE   HLTMS   ;PRINT 'HALTED' MESSAGE
  (1)  007317  337                      RST     3       ;PRINT LINE OF CHARS
  (1)  007320  116  037                 .ADDR   HLTMS   ;BUFF TO PRINT
 3450  007322  041  012  040            LXI     H,EMBUF ;PNTR TO DATA IN LOC 0
 3451  007325  315  016  031            CALL    P18     ;PRINT RIGHT HALF OF 36-BIT DATA
 3452  007330                         CLRB    CHKHLT  ;SET FLAG TO SAY WEVE TYPED HALTED ALREADY
 3453
 3454  007332                         PSPACE          ;PRINT A SPACE
  (2)  007332  317                      RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (2)  007333  040                      .BYTE   SPACE   ;CHAR TO PRINT
 3455  007334                         PSPACE          ;AND ANOTHER
  (2)  007334  317                      RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (2)  007335  040                      .BYTE   SPACE   ;CHAR TO PRINT
 3456  007336                 PCCOM:  INTON           ;SET INTERNAL MODE
  (1)  007336  327                      RST     2       ;GO SET INTERNAL MODE
 3457  007337                         EXAM    1       ;EXAMINE WORD WHICH HOLDS THE PC
  (1)  007337  315  216  030            CALL    EXAMSH  ;AND DO EXAM ASSUMING SHORT ADDRESS
  (1)  007342  001  000                 .ADDR   1       ;ADDR TO BE ZAPPED PASSED AS TRAILING ARG
 3458  007344                         INTOFF          ;CLR INTERNAL MODE
  (1)  007344  367                      RST     6       ;GO EXIT FROM INTERNAL MODE
 3459  007345                         PLINE   PCMSG   ;PRINT 'PC/'
  (1)  007345  337                      RST     3       ;PRINT LINE OF CHARS
  (1)  007346  112  037                 .ADDR   PCMSG   ;BUFF TO PRINT
 3460  007350  315  347  030            CALL    P36.    ;AND PRINT THE PC
 3461  007353                         PCRLF           ;PRINT CARRIAGE RETURN LINE-FEED
  (2)  007353  347                      RST     4
  (2)  007354  002                      .BYTE   2
 3462  007355  247                      ANA     A       ;CLEAR THE C-BIT 'CAUSE ALL OK
 3463
 3464                           ;AND BEFORE WE LEAVE,RESTORE THE PARITY STUFF
 3465  007356  072  354  040    SMFINI: LDA     PARBT   ;GET CURRENT PARITY DEFAULTS
 3466  007361  323  100                 OUT     RESET   ;RESTORE THE PARITY DETECTS
 3467  007363  311                      RET             ;AND OUT
 3468  007364                 ZEROES:
 3469  007364                 MAD000: D       0,0,,0,0 ;MEMORY ADDRESS 0
 3470                                                   ;  FROM MEMORY LOCATION 0
```

```
 3472                           .SBTTL  *** 'PE' CMD ***
 3473                           ;COMMAND TO ENABLE VARIOUS PARITY CHECKS NORMALLY MADE BY THE 8080..
 3474                           ;ACCEPTABLE PARITY COMMANDS ARE:
 3475                           ;       PE      0       ;'DISABLE' ALL PARITY DETECTION
 3476                           ;       PE      1       ;'ENALE' 'DP' PARITY DETECT
 3477                           ;       PE      2       ;'ENABLE' 'CRM' PARITY DETECTION
 3478                           ;       PE      4       ;'ENABLE' 'PE' PARITY DET(CLK FREEZE ON PAR ERR)
 3479                           ;       PE      7       ;'ENABLE' ALL
 3480                           ;BITS ARE WEIGHTED FOR THE THREE TYPES OF PARITY ERRORS
 3481  007371  332  020  017   PECMD:  JC      PARDIS  ;COMMAND REQUIRES ARG
 3482
 3483  007374  315  244  030           CALL    ARG16.  ;ASSEMBLE TYPED ARG
 3484
 3485  007377  175                   MOV     A,L     ;GET NUMBER TYPED INTO THE ACCUM
 3486  007400  346  007               ANI     ^O7     ;KEEP ONLY APPROPRIATE BITS
 3487  007402  027                   RAL             ;BIT 0 INTO BIT1
 3488  007403  027                   RAL             ;BIT 0 INTO BIT 2
 3489  007404  027                   RAL             ;  INTO BIT 3
 3490  007405  027                   RAL             ;  INTO BIT 4
 3491
 3492  007406  157                   MOV     L,A     ;SAVE IN L
 3493  007407  072  354  040         LDA     PARBT   ;NOW GET CURRENT STATUS
 3494  007412  346  217               ANI     ^O217   ;OFF THE OLD CRUMMY PARITY
 3495  007414  265                   ORA     L       ;THROW IN THESE NEW BITS
 3496
 3497  007415  303  361  013           JMP     KS.PAR  ;SAVE IN RAM PLACE & WRITE TO KS
 3498
 3499
 3500                           ;THIS IS THE CODE FOR IF WE WANT TO DISPLAY THE PARITY
 3501  007420  072  354  040   PARDIS: LDA     PARBT   ;GET THE CURRENT PARITY STATUS
 3502  007423  346  160               ANI     ^O160   ;CLR CRUD, JUST SAVING PARITY BITS
 3503  007425  037                   RAR             ;ROTATE TO JUSTIFY THE BITS AT BIT 0
 3504  007426  037                   RAR
 3505  007427  037                   RAR
 3506  007430  037                   RAR
 3507
 3508  007431  315  304  030   P8CRLF: CALL    P8BITA  ;AND GO PRINT THOSE 8 BITS
 3509  007434                           PCRLF           ;TERMINATE ALL WITH A CR-LF
  (2)  007434  347                   RST     4
  (2)  007435  002                   .BYTE   2
 3510  007436                           ENDCMD          ;ALL DONE
  (1)  007436  311                   RET             ;RETURN TO CALLER
 3511
```

```
                        .SBTTL  *** 'EX' CMD ***
                        ;CONSOLE COMMAND TO EXECUTE A SINGLE SM10 'TEN ORDER' INSTRUCTION

007437                  EXCMD:  ARG36           ;GO ASSEMBLE THE INSTR TO BE EXECUTED
(1) 007437 347                  RST     4
(1) 007440 010                  .BYTE   8
007441 012 040                  .ADDR   EMBUF   ;PLACE TO PUT IT

007443 021 012 040      EXINTM: LXI     D,EMBUF ;POINTER TO INSTR INTO 'D,E'

007446 315 143 033      EXINT:  CALL    WDATP   ;***** I/O WRT 102,104,106,101,112 *****

007451 076 002                  MVI     A,2     ;SET BIT FOR 'I/O DATA CYCLE'
007453 323 114                  OUT     DTARB   ;***** I/O WRT 114/2 *****

007455 076 003                  MVI     A,3     ;BITS FOR 'EXECUTE' & 'CONTINUE'
007457 323 212                  OUT     CPUCTL  ;***** I/O WRT 212/3 *****

007461 000              DNF:    NOP
007462 000                      NOP             ;WAIT

007463 333 300                  IN      RUNFP   ;***** I/O RD 300 *****
007465 057                      CMA             ;FIX INVERSION
007466 346 001                  ANI     1       ;IS CONTINUE STILL SET??
007470 310                      RZ              ;IF CLR, WE ARE OK...

                        ;IF CONT STILL SET, WE HAVE AN ERROR
007471                          PLINE   EXMS    ;ERR MESSAGE '?DNF-DID NOT FINISH'
(1) 007471 337                  RST     3       ;PRINT LINE OF CHARS
(1) 007472 125 037              .ADDR   EXMS    ;BUFF TO PRINT
007474 315 346 006              CALL    CLRUSE  ;EXIT FROM USER MODE
007477 057                      CMA             ;ACCUM NOW .EQ. -1
007500 247                      ANA     A       ;SET FLAGS, SO 'JNZ' WILL JUMP
007501 311                      RET             ;AND RETURN
```

```
 3544                        .SBTTL  *** 'ST' CMD ***
 3545                        ;CONSOLE COMMAND TO ISSUE A START TO THE CPU
 3546  007502  315  120  013  STCMD:  CALL    LACMD   ;FIRST GO ASSEMBLE A LEGAL ADDRESS AT WHICH TO START
 3547  007505                       MOV5B           ;MOVE TO TMP BUFF SO DONT KILL 'MEMAD'
  (1)  007505  347                  RST     4
  (1)  007506  000                  .BYTE   0
 3548  007507  017  040             .ADDR   MEMAD   ;SRC
 3549  007511  073  040             .ADDR   TMPBF2  ;TEMP PLACE TO KEEP IT
 3550
 3551
 3552  007513                 STINT:  CLRRM   DMDAT           ;MUST CLR COMM WORDS BEFORE WE START
  (1)  007513  347                  RST     4
  (1)  007514  012                  .BYTE   10.
  (1)  007515  054  040             .ADDR   DMDAT+5
 3553
 3554  007517                       DEPOS   31              ;CLEAR LOC 31(KEEP ALIVE WORD)
  (1)  007517  247                  ANA     A       ;CLR 'C-BIT' FOR USE BY COMMON CODE
  (1)  007520  315  217  030        CALL    DEPSHT  ;AND DO THE DEPOSIT ASSUMING SHORT ADDR
  (1)  007523  031  000             .ADDR   31      ;ADDR TO BE ZAPPED PASSED AS TRAILING ARG
 3555  007525                       DEPOS   32              ;CLEAR LOC 32(THE TTY INPUT WORD)
  (1)  007525  247                  ANA     A       ;CLR 'C-BIT' FOR USE BY COMMON CODE
  (1)  007526  315  217  030        CALL    DEPSHT  ;AND DO THE DEPOSIT ASSUMING SHORT ADDR
  (1)  007531  032  000             .ADDR   32      ;ADDR TO BE ZAPPED PASSED AS TRAILING ARG
 3556  007533                       DEPOS   33              ;AND 33(THE TTY OUTPUT WORD)
  (1)  007533  247                  ANA     A       ;CLR 'C-BIT' FOR USE BY COMMON CODE
  (1)  007534  315  217  030        CALL    DEPSHT  ;AND DO THE DEPOSIT ASSUMING SHORT ADDR
  (1)  007537  033  000             .ADDR   33      ;ADDR TO BE ZAPPED PASSED AS TRAILING ARG
 3557
 3558  007541  041  140  005        LXI     H,^O2540        ;LOAD 'H,L' WITH 'JRST' OPCODE
 3559  007544  042  076  040        SHLD    TMPBF2+3        ;AND PUT INTO THE BUFFER WHERE THE ADDR IS
 3560
 3561  007547  021  073  040        LXI     D,TMPBF2        ;NOW SET POINTER TO THE INSTR
 3562  007552  315  046  017        CALL    EXINT           ;AND GO HANDLE JUST LIKE AN EXECUTE
 3563  007555  300                  RNZ                     ;IF NON ZERO, THE EXECUTE FAILED
 3564
 3565  007556                       LONG,DELAY 1    ;NOW WAIT
  (1)  007556  041  310  000        LXI     H,200. * 1 ;SET UP THE TIMING COUNT IN H,L REG
  (1)  007561  315  134  014        CALL    LTLOOP  ;AND GO DELAY ABOUT 1 SECOND FOR EACH (300 COUNT)
 3566
 3567                        ;AND NOW FALL INTO THE 'CONTINUE' COMMAND
```

```
 3569                           .SBTTL  *** 'CO' CMD ***
 3570                           ;CONSOLE COMMAND TO ISSUE CONTINUE TO CPU
 3571  007564  315  017  007    COCMD:  CALL    SETUSE  ;ENTER USER MODE
 3572  007567  076  005                MVI     A,^O5   ;SET BITS FOR 'CONTINUE' & 'RUN'
 3573  007571  323  212                OUT     CPUCTL  ;***** I/O WRT 212/5 *****
 3574  007573  062  231  040    COINT:  STA     CHKHLT  ;AND GAURANTEE THAT ANY FAST HALTS WILL GET REPORTED
 3575
 3576  007576                         PLINE   KSPRMT  ;WANT TO TELL USER WHEN WE SWITCH MODES
  (1)  007576  337                    RST     3       ;PRINT LINE OF CHARS
  (1)  007577  053  037               .ADDR   KSPRMT  ;BUFF TO PRINT
 3577  007601                         PLINE   U       ;SAY 'USER MODE'
  (1)  007601  337                    RST     3       ;PRINT LINE OF CHARS
  (1)  007602  006  007               .ADDR   U       ;BUFF TO PRINT
 3578
 3579                           ;AND JUMP OFF TO COMMON CODE THAT CHECKS THE CONTINUE BIT
 3580                           ;AND ERRS IF CONTINUE HAS NOT BEEN CLEARED BY THE CPU
 3581  007604  303  061  017            JMP     DNF     ;GO....
```

```
 3583                           .SBTTL  *** 'HA' CMD ***
 3584                           ;CONSOLE COMMAND TO HALT THE SM10 CPU..CPU MICRO-CODE SHOULD ENTER
 3585                           ;THE HALT LOOP
 3586  007607  257              HACMD:  XRA     A       ;CLR ACCUM FOR 'RUN','EXECUTE' & 'CONTINUE'
 3587  007610  323  212                 OUT     CPUCTL  ;*****I/O WRT 212/0 *****
 3588  007612  303  257  016           JMP     HLTCM   ;AND FINISHES UP JUST LIKE 'SM' COMMAND
 3589
 3590                           .SBTTL  *** 'SH' CMD ***
 3591                           ;COMMAND TO CAUSE THE TOPS20 MONITOR TO BEGIN AN ORDERLY SYSTEM SHUTDOWN
 3592  007615                   SHCMD:  MOV5B           ;MOVE US SOME DATA
  (1)  007615  347                     RST     4
  (1)  007616  000                     .BYTE   0
 3593  007617  053  001                 .ADDR   .DSBASE ;FROM HERE (0,,776700)
 3594  007621  047  040                 .ADDR   DMDAT   ;TO HERE, PLACE FOR DEPOSIT TO FIND IT
 3595
 3596  007623                           DEPOS   30      ;AND DO IT
  (1)  007623  247                     ANA     A       ;CLR 'C-BIT' FOR USE BY COMMON CODE
  (1)  007624  315  217  030           CALL    DEPSHT  ;AND DO THE DEPOSIT ASSUMING SHORT ADDR
  (1)  007627  030  000                .ADDR   30      ;ADDR TO BE ZAPPED PASSED AS TRAILING ARG
 3597  007631  315  017  007           CALL    SETUSE  ;NOW BE SURE WE ENTER THE USER MODE AGAIN
 3598  007634  006  200                 MVI     B,^O200 ;SET SIGN BIT, CAUSES 'KEEP-ALIVE' TO BE IGNORED
 3599  007636  315  326  032           CALL    STATEM  ;GO DO THE STUFF
 3600  007641  377                     .BYTE   ^O377   ;WITHOUT CHANGING THINGS
 3601  007642  303  173  017           JMP     COINT   ;AND GO BACK TO USER MODE TO WATCH MONITOR 'SHUTDOWN'
```

```
 3603                           .SBTTL  *** 'KL' CMD ***
 3604                           ;KLINIK COMMAND
 3605  007645  332  300  017   KLCMD:  JC      KLDIS   ;DISPLAY CURRENT STATE IF NOTHING TYPED
 3606
 3607                           ;ELSE ASSEMBLE THE TYPED IN ARG
 3608  007650  315  244  030           CALL    ARG16.  ;PERMIT 16 BIT WIDE TYPE IN
 3609
 3610                           ;NOW VERIFY THAT LOWER HALF IS A LEGAL NUMBER
 3611  007653  175                   MOV     A,L     ;LO HALF TO ACCUM
 3612  007654  247                   ANA     A       ;SET CONDITION CODES
 3613  007655  312  264  017         JZ      KLOFF   ;IF TYPED ZERO, GO TURN OFF KLINIK
 3614
 3615                           ;NOW FALL THRU HERE IF .EQ. 1, MUST TURN ON THE KLINIK
 3616  007660  062  300  040         STA     KLLINE.ON ;SET BIT TO SAY THAT KLINIK IS ON
 3617  007663  311                   RET             ;AND OUT
 3618  007664  062  300  040   KLOFF:  STA     KLLINE.ON ;GET HERE IF ACCUM WAS ZERO.. ZAP THE KLINIK FLAG
 3619
 3620                           ;AND FALL INTO CODE TO SEE IF THE END OF KLINIK MUST FORCE A CHANGE IN THE
 3621                           ;STATE OF THE KLINIK LINE AND USER. IE IF IN MODE 3, WE MUST FORCE USER
 3622                           ;INTO MODE 2
 3623  007667  072  247  040         LDA     CSLMODE ;GET CURRENT MODE
 3624  007672  376  010               CPI     .MODE3  ;IS IT MODE 3??
 3625  007674  314  101  034         CZ      SETM2   ;SET MODE 2 IF NOT IN THERE
 3626  007677  311                   RET             ;AND OUT
 3627
 3628                           ;HERE IF JUST WANT TO DISPLAY CURRENT KLINIK STATE
 3629  007700  072  300  040   KLDIS:  LDA     KLLINE.ON ;GET CURRENT VALUE
 3630  007703  247                   ANA     A       ;SET FLAGS
 3631  007704  303  374  013         JMP     CHOOSE  ;GO DISPLAY THE APPROPRIATE MESSAGE
```

```
 3633                           .SBTTL  *** 'TT' CMD ***
 3634  007707  315  017  007    TTCMD:  CALL    SETUSE  ;ON THIS COMMAND WE DESIRE THAT THE CTY ENTER USER MODE
 3635  007712                           PLINE   KSPRMT  ;'KS10>'
  (1)  007712  337                    RST     3       ;PRINT LINE OF CHARS
  (1)  007713  053  037               .ADDR   KSPRMT  ;BUFF TO PRINT
 3636  007715                           PLINE   U       ; 'USR MOD'
  (1)  007715  337                    RST     3       ;PRINT LINE OF CHARS
  (1)  007716  006  007               .ADDR   U       ;BUFF TO PRINT
 3637  007720  072  247  040           LDA     CSLMODE ;CHECK KLINIK STATUS -  PUT IT INTO MODE 2?
 3638  007723  326  004                SUI     .MODE2  ;IS MODE LESS THAN MODE 2?
 3639  007725  362  101  034           JP      SETM2   ;NO - OK TO SET MODE 2 - SO GO DO IT
 3640  007730                           CLRB    KLNKSW  ;CLEAR KLINIK STATUS WORD - FORCE A REEXAMINATION OF
 3641                                                   ;  OF THE SWITCHES ...
 3642  007732  311                      RET             ;NOW RETURN
 3643
 3644
 3645                           .SBTTL  *** 'PW' CMD ***
 3646                           ;COMMAND FOR SETTING A PASSWORD INTO THE 8080, SO THAT THE KLINIK LINE
 3647                           ;USER WILL HAVE SOMETHING TO MATCH AGAINST WHEN HE TRIES TO GET INTO THE SYSTEM
 3648  007733                   PWCMD:  CLRB    KLNKSW  ;ON ANY PASSWORD COMMAND, FORCE A RE-EXAMINATION
 3649                                                   ;  OF THE KLINIK MODE
 3650
 3651                           ;NOW DO THE NORMAL STUFF U NEED TO DO WITH THIS COMMAND
 3652  007735  332  375  017           JC      PW.CLR  ;IF NO PW TYPED, GO CLR PASSWORD
 3653
 3654                           ;FALL THRU ELSE.. IE MUST SET THE PASSWORD
 3655  007740  052  223  040   PW.SET: LHLD    .ARG1   ;GET POINTER TO THE TYPE-IN BUFFER
 3656  007743  021  262  040           LXI     D,PASSWORD ;POINT TO THE BUFFER AREA WHERE THE PASSWORD WILL BE
 3657  007746  006  372                MVI     B,-6    ;SET A MAX COUNT FOR THE LENGTH OF THE PASSWORD
 3658
 3659  007750  176             PW.LOOP: MOV    A,M     ;COPY A PASSWORD CHARACTER TO THE ACCUM
 3660  007751  376  377                CPI     EOLCH   ;IS IT END OF LINE??
 3661  007753  310                      RZ              ;IF YES,SIMPLE RETURN
 3662
 3663                           ;FALL TO HERE IF MORE TO BE MOVED
 3664  007754  315  214  006           CALL    UP.LO   ;UPPER CASE ONLY
 3665  007757  022                      STAX    D       ;MOVE A CHARACTER TO THE SAVE BUFFER
 3666  007760  023                      INX     D       ;UPDATE DESTINATION POINTER
 3667  007761  043                      INX     H       ;UPDATE SOURCE POINTER
 3668  007762  004                      INR     B       ;UPDATE CHARACTER COUNT
 3669  007763  302  350  017           JNZ     PW.LOOP ;STAY IN THE LOOP
 3670
 3671                           ;FALL THRU WHEN DONE 6 CHRACTERS.. THAT HAD BETTER BE ALL, ELSE ERROR
 3672  007766  176                      MOV     A,M     ;GET 7TH CHARACTER
 3673  007767  376  377                CPI     EOLCH   ;IT BETTER BE END
 3674  007771  310                      RZ              ;IF YES, WAS END OF LINE, THEN OK TO RETURN
 3675
 3676                           ;FALL THRU WHEN USER TYPED TOO MANY CHARACTERS
 3677  007772                           PLINE   PWLEN   ;ERR MESSAGE
  (1)  007772  337                    RST     3       ;PRINT LINE OF CHARS
  (1)  007773  344  037               .ADDR   PWLEN   ;BUFF TO PRINT
 3678  007775                   PW.CLR: CLRRM   PASSWORD+1 ;CLEAR 5 BYTES OF THE 6 BYTE BUFFER
  (1)  007775  347                    RST     4
  (1)  007776  012                    .BYTE   10.
```

```
 (1)  007777  270  040              .ADDR   PASSWORD+1+5
3679  010001  053                   DCX     H       ;POINTER CAME OUT GOOD
3680  010002  066  000              MVI     M,0     ;CLR THE 6TH BYTE OF THE BUFFER
3681  010004  311                   RET             ;THAT'S ALL
3682
```

```
 3684                           .SBTTL  *** 'MK' & 'UM' CMD ***
 3685                           ;CONSOLE COMMAND TO MARK AND UMARK SPECIFIED MICRO-CODE LOCATIONS
 3686  010005  016  000         UMCMD:  MVI     C,0     ;WE ARE CLEARING THE MARK BIT..
 3687  010007  303  014  020           JMP     MRKCM   ;AND GO SAVE IT AS A FLAG FOR WHAT WE ARE DOING
 3688
 3689  010012  016  001         MKCMD:  MVI     C,1     ;A BIT SAYS WE ARE SETTING THE MARK BIT
 3690  010014  305               MRKCM:  PUSH    B       ;SAVE 'B,C', IT HAS DATA FOR SET OR CLEAR
 3691  010015                           RUN..           ;IS CPU RUNNING??
  (1)  010015  347                     RST     4
  (1)  010016  006                     .BYTE   6
 3692  010017  315  331  013           CALL    LCCMD   ;'C-BIT' IS CLR..GO ASSEMBLE A LEGAL RAM-ADDRESS
 3693  010022  315  315  014           CALL    CRM.AD  ;SET DIAGNOSTIC ADDRESS REG
 3694  010025  315  204  013           CALL    CP1     ;GIVE SINGLE PULSE TO GET DATA WHERE I CAN READ
 3695
 3696  010030  076  017               MVI     A,^O17  ;GET FUNCTION READ FOR CRAM DATA THAT INCLUDES MRK
 3697  010032  315  362  014           CALL    READC   ;DO THE DIAGNOSTIC FUNCTION READ
 3698  010035  315  315  014           CALL    CRM.AD  ;SET DIAGNOSTIC ADDRESS REG
 3699  010040  021  100  040           LXI     D,TMPB2 ;GET PNTR TO DATA THAT HAS THE MARK BIT
 3700  010043  301                     POP     B       ;GET INSTR TYPE
 3701
 3702  010044  032                     LDAX    D       ;GET THE ACTUAL DATA
 3703  010045  346  376                ANI     ^O376   ;CLEAR BIT 0
 3704  010047  261                     ORA     C       ;NOW EITHER SET OR CLEAR THE BIT
 3705
 3706  010050  022               MRKRT:  STAX    D       ;BUT DATA BACK, NEW MARK BIT STATUS
 3707
 3708  010051  315  176  033           CALL    ADATP   ;WRITE DATA TO BUS REG
 3709
 3710  010054  076  007               MVI     A,7     ;NOW WISH TO DO FUNCTION WRITE 7
 3711  010056  062  010  040           STA     CRMFN   ;SET INTO FUNC WORD
 3712
 3713  010061  303  271  014           JMP     WFNC1   ;AND FINISH UP BY WRITING DATA BACK
```

```
 3715                           .SBTTL  *** 'ZM' CMD ***
 3716                           ;CONSOLE COMMAND TO ZERO THE SM10 MOS MEMORY..
 3717  010064                   ZMCMD:  CLRRM   MEMAD   ;CLEAR MEMORY ADDRESS BUFFER(TO START AT 0)
  (1)  010064  347                       RST     4
  (1)  010065  012                       .BYTE   10.
  (1)  010066  024  040                  .ADDR   MEMAD+5
 3718  010070  076  002                  MVI     A,2     ;BITS TO SAY WRITE TYPE FUNCTION
 3719  010072  062  023  040             STA     MEMAD+4 ;WRITE INTO THE BUFFER
 3720
 3721  010075                           CLRRM   DMDAT   ;DATA TO DEPOSIT IS ALL ZEROES
  (1)  010075  347                       RST     4
  (1)  010076  012                       .BYTE   10.
  (1)  010077  054  040                  .ADDR   DMDAT+5
 3722
 3723  010101                           INTON           ;INTERNAL MODE ON
  (1)  010101  327                       RST     2       ;GO SET INTERNAL MODE
 3724  010102  315  230  012             CALL    DM1     ;DEPOSIT ZEROES INTO FIRST LOCATION
 3725  010105  315  055  033   ZM1:      CALL    INC36   ;TO NEXT ADDRESS
 3726  010110  017  040                  .ADDR   MEMAD   ;HERE IT IS
 3727
 3728  010112  021  017  040             LXI     D,MEMAD ;DO PART OF THE DEPOSIT HERE, FOR SPEED  SAKE
 3729  010115  315  176  033             CALL    ADATP   ;LOAD UP BUS REGS WITH THE DESIRED DATA
 3730  010120  076  004                  MVI     A,4     ;NOW FUNCTION TYPE BIT INTO ACCUM
 3731  010122  323  115                  OUT     BUSARB  ;***** I/O WRT *****
 3732
 3733  010124  315  274  012             CALL    DMGO    ;NOW GO DO THE DEPOSIT
 3734
 3735                           ;AND CHECK TO SEE IF GOT A NXM
 3736  010127  072  154  040             LDA     ERRCD   ;GET ERROR CODE..
 3737  010132  247                       ANA     A       ;CHECK IF SET
 3738  010133  312  105  020             JZ      ZM1     ;IF NO ERRORS YET, KEEP GOING
 3739
 3740                           ;FALL THRU WHEN HAD 'NO DATA ACNOWLEDGE' ERROR
 3741  010136                           INTOFF          ;CLEAR INTERNAL MODE
  (1)  010136  367                       RST     6       ;GO EXIT FROM INTERNAL MODE
 3742  010137                           ENDCMD          ;AND DONE
  (1)  010137  311                       RET             ;RETURN TO CALLER
```

```
 3744                          .SBTTL  *** 'RP' CMD ***
 3745
 3746                          ;NOTE: THE LIST OF SAVED COMMAND DISPATCHES IS NOT IN THE NORMAL 8080
 3747                          ;ADDRESS FORMAT..IE THE COMMAND LIST IS SAVE IN PAIRS OF BYTES AS
 3748                          ;HI ORDER PIECE FIRST..
 3749                          ;LO ORDER PIECE SECOND..
 3750  010140  322  165  020   RPCMD: JNC     RP1     ;IF ARG, BEGIN AT A SPECIAL PLACE
 3751  010143  257                 XRA     A       ;CLR ACCUM
 3752  010144  062  160  040   RP0:   STA     RPCNTR  ;THERE IS NO REPEAT COUNT
 3753  010147  315  307  020         CALL    RPF00   ;IN THE BEGINNING YOU MUST RESET THE POINTERS
 3754  010152  257                 XRA     A       ;CLR ACCUM
 3755  010153  062  157  040         STA     RPEND   ;CLR THE REPEAT KILLER
 3756  010156  057                 CMA             ;MAKE ACCUM .EQ. -1
 3757  010157  062  213  040         STA     RPTON   ; THAT REPEAT FUNCTION IS TURNED ON
 3758  010162  303  211  020         JMP     RP2     ;CONTINUE...
 3759
 3760  010165  315  244  030   RP1:   CALL    ARG16.  ;FETCH THE ARG THAT WAS TYPED
 3761
 3762  010170  174                 MOV     A,H     ;IT MUST ONLY BE 256 OR LESS
 3763  010171  247                 ANA     A       ;SET PROCESSOR FLAGS
 3764  010172  302  150  032         JNZ     KILNM   ;IF .GT. 256, THEN BAD NUMBER
 3765
 3766                          ;FALL THRU IF ACCUM 0
 3767  010175  175                 MOV     A,L     ;GET REAL ARG INTO ACCUM
 3768  010176  074                 INR     A       ;SET ACCUM 1 GREATER THAN ACTUAL
 3769  010177  303  144  020         JMP     RP0     ;CONTINUE BY INITING FLAGS
 3770
 3771  010202  072  157  040   RPTRTN: LDA     RPEND   ;NEXT THING IS TO SEE IF TIME TO STOP REPEAT
 3772  010205  247                 ANA     A       ;TEST DATA
 3773  010206  302  247  020         JNZ     RP.OUT  ;AND END THE REPEAT IF 'END' FLAG IS SET
```

```
 3775  010211  052  211  040  RP2:    LHLD    RPLST   ;GET POINTER TO COMMAND DISPATCH LIST
 3776  010214  176                   MOV     A,M     ;CHECK BYTE..MAKE SURE ITS NOT THE END-OF-LIST
 3777  010215  074                   INR     A       ;IF IT WAS -1, NOW ITS A ZERO
 3778  010216  302  254  020         JNZ     RP4     ;AND GO BACK TOO
 3779
 3780  010221  072  160  040         LDA     RPCNTR  ;CHECK IF THIS IS A COUNTED REPEAT
 3781  010224  314  307  020         CZ      RPFOO   ;IT WAS END OF LIST IF U GOT HERE, SO FIX POINTERS
 3782  010227  247                   ANA     A       ;SET FLAGS
 3783  010230  312  202  020         JZ      RPTRTN  ;IF .EQ. 0 NO COUNT ON THE REPEAT
 3784
 3785  010233  075                   DCR     A       ;THERE IS A COUNTER, DOWN IT
 3786  010234  062  160  040         STA     RPCNTR  ;SAVE NEW COUNT
 3787  010237  376  001             CPI     1       ;SEE IF AT BOTTOM LINE
 3788  010241  304  307  020         CNZ     RPFOO   ;IF A COUNTED REPEAT, FIX END OF LIST ONLY IF MORE TO DO
 3789  010244  302  202  020         JNZ     RPTRTN  ;JUMP IF NO
 3790
 3791  010247  257           RP.OUT: XRA     A       ;CLEAR ACCUM PRIOR TO LEAVING
 3792  010250  062  213  040         STA     RPTON   ;TURN OFF THE 'ON' FLAG
 3793  010253  311                   RET             ;AND HERE IF YES
 3794
 3795  010254  126           RP4:    MOV     D,M     ;IF IT WAS OK.. START ASSEMBLING THE DISPATCH
 3796  010255  043                   INX     H       ;UPDATE TO LO ORDER PIECE
 3797  010256  136                   MOV     E,M     ;AND DISPATCH IS NOW IN 'D,E'
 3798  010257  043                   INX     H       ;UPDATE POINTER
 3799  010260  042  211  040         SHLD    RPLST   ;SAVE POINTER TO WHERE WE ARE IN CMD LIST
 3800
 3801  010263  041  340  001         LXI     H,NULLW ;'H,L' GETS PLACE WE WANT TO RETURN TO
 3802  010266  345                   PUSH    H       ;PLACE ON STACK SO THAT 'RET' INS COMES HERE
 3803  010267  353                   XCHG            ;DISPATCH ADDRESS INTO 'H,L'
 3804
 3805  010270  174                   MOV     A,H     ;GET HI ORDER PIECE OF ADDR TO SEE IF ARG
 3806  010271  247                   ANA     A       ; WAS TYPED WITH THIS COMMAND..SET FLAGS
 3807  010272  362  301  020         JP      RPGO    ;IF SIGN BIT CLR, CMD GOT NO ARG
```

```
 3809                           ;OTHERWISE MUST SET THE C-BIT TO TELL CMD TO LOOK FOR ARG
 3810  010275  346  177                ANI     ^O177   ;CLR SIGN BIT
 3811  010277  147                      MOV     H,A     ;PUT IT BACK FOR CORRECT DISPATH
 3812  010300  067                      STC             ;SET C-BIT IF NECCESSARY
 3813  010301  077                RPGO: CMC             ;SET C-BIT FOR THIS COMMAND TO SEE
 3814  010302  351                      PCHL            ;AND GO DO IT..
 3815
 3816                           ;IF REACHED END OF THE DISPATCH LIST, THEN THIS CODE RESETS
 3817                           ;THE POINTER BACK TO THE BEGINNING OF THE LIST
 3818  010303  057                RPNEW: CMA             ;RPFOO DOESNT TOUCH ACCUM, SET ACCUM TO -1
 3819  010304  062  131  040           STA     CMDS..  ;NOW ZAP THE FIRST IN LINE FLAG
 3820  010307  041  001  041   RPFOO: LXI     H,RPINI ;BUFFER BEGINNING ADDRESS
 3821  010312  042  211  040           SHLD    RPLST   ;PUT BACK INTO RAM
 3822  010315  041  032  041           LXI     H,RPTBFI ;POINTER TO DATA BUFFER
 3823  010320  042  207  040           SHLD    RPBUFS  ;RESET INTO HOLDING LOCATION
 3824  010323  311                      RET             ;AND RETURN
```

```
 3826                           .SBTTL  *** 'DS' CMD ***
 3827                           ;COMMAND TO SELECT NON DEFAULT DISK UNIT AND UNIBUS ADAPTERS FOR BOOTING
 3828                           ;FROM DISK
 3829  010324                  DSCMD:  PLINE   Q.UBA    ;MESSAGE TO ASK FOR 'UNIBUS ADAPTER' TO BE USED
  (1)  010324  337                        RST     3       ;PRINT LINE OF CHARS
  (1)  010325  252  037                   .ADDR   Q.UBA   ;BUFF TO PRINT
 3830  010327  315  160  021             CALL    PICKUP  ;GO FETCH THE RESPONSE THAT WAS TYPED
 3831  010332  332  345  020             JC      DS1     ;IF NOTHING TYPED, LEAVE UBA AS CURRENTLY SELECTED
 3832
 3833                           ;FALL INTO HERE IF A NEW UBA NUMBER WAS TYPED
 3834  010335  072  100  040             LDA     TMPB2   ;GRAB THE NEW UBA NUMBER AS TYPED
 3835  010340  007                       RLC             ;THE UBA NUMBER IS JUSTIFIED '*4' IN A BYTE
 3836  010341  007                       RLC             ;TAKES 2 ROTATES TO GET IT JUSTIFIED
 3837  010342  062  357  040             STA     DSKUBA  ;AND SAVE THE NEW VALUE IN THE RAM
 3838
 3839  010345                  DS1:    PLINE   Q.RH    ;ASK FOR AN RH11 TO USE
  (1)  010345  337                        RST     3       ;PRINT LINE OF CHARS
  (1)  010346  261  037                   .ADDR   Q.RH    ;BUFF TO PRINT
 3840  010350  315  160  021             CALL    PICKUP  ;GET WHAT WAS TYPED
 3841  010353  332  364  020             JC      DS2     ;IF NOTHING TYPED, DO NOTHING
 3842
 3843  010356                          MOV5B           ;NOW SAVE THIS NEW DISK BASE
  (1)  010356  347                        RST     4
  (1)  010357  000                        .BYTE   0
 3844  010360  100  040                   .ADDR   TMPB2   ;THIS IS WHERE THE DATA SHOULD BE SITTING
 3845  010362  374  040                   .ADDR   DSBASE  ;THIS IS WHERE WE WILL KEEP IT
 3846
 3847  010364                  DS2:    PLINE   Q.UNIT  ;ASK FOR A UNIT NUMBER TO BOOT FROM
  (1)  010364  337                        RST     3       ;PRINT LINE OF CHARS
  (1)  010365  273  037                   .ADDR   Q.UNIT  ;BUFF TO PRINT
 3848  010367  315  160  021             CALL    PICKUP  ;GO FETCH WHAT WAS TYPED
 3849  010372  330                       RC              ;IF NOTHING TYPED, THEN ALL DONE. RETURN FROM THIS CMD
 3850
 3851                           ;FALL TO HERE IF A UNIT WAS TYPED..GO SET THE UNIT TO BE USED
 3852  010373  072  100  040             LDA     TMPB2   ;GET NUMBER TYPED FOR THE NEW UNIT
 3853  010376  062  132  040             STA     UNITNM  ;SET IT INTO RAM AS THE NEW VALUE
 3854  010401  311                       RET             ;ALL DONE THIS COMMAND....
```

```
 3856                           .SBTTL  *** 'MS' CMD ***
 3857                           ;COMMAND TO SELECT WHAT MAGTAPE TO BOOT FROM
 3858  010402                   MSCMD:  PLINE   Q.UBA    ;ASK FOR A UNIBUS ADAPTER TO LOAD FROM
  (1)  010402  337                       RST     3       ;PRINT LINE OF CHARS
  (1)  010403  252  037                  .ADDR   Q.UBA   ;BUFF TO PRINT
 3859  010405  315  160  021            CALL    PICKUP  ;GET WHAT WAS TYPED
 3860  010410  332  023  021            JC      MS1     ;IF NOTHING, LEAVE UBA ALONE.. GO GET THE NEXT THING
 3861
 3862                           ;FALL INTO HERE IF A NEW UBA WAS SELECTED
 3863  010413  072  100  040            LDA     TMPB2   ;GET THE NEW UBA TYPED
 3864  010416  007                      RLC             ;UBA NUMBERS MUST BE JUSTIFIED ON BYTE BOUNDARY *4
 3865  010417  007                      RLC             ;TAKES TWO SHIFTS TO SET THE UBA NUMBER
 3866  010420  062  356  040            STA     MTAUBA  ;SAVE THE NEW UBA VALUE IN THE RAM
 3867
 3868  010423                   MS1:    PLINE   Q.RH    ;ASK FOR A NEW RH11 TO USE
  (1)  010423  337                       RST     3       ;PRINT LINE OF CHARS
  (1)  010424  261  037                  .ADDR   Q.RH    ;BUFF TO PRINT
 3869  010426  315  160  021            CALL    PICKUP  ;FETCH WHAT WAS TYPED
 3870  010431  332  042  021            JC      MS1.5   ;IF NOTHING TYPED, THEN DO NOTHING
 3871
 3872  010434                           MOV5B           ;IF SOMETHING TYPED, GET IT FROM THE BUFFER
  (1)  010434  347                       RST     4
  (1)  010435  000                       .BYTE   0
 3873  010436  100  040                  .ADDR   TMPB2   ;PLACE WHERE THE STUFF WAS PUT
 3874  010440  367  040                  .ADDR   MTBASE  ;PLACE WHERE WE KEEP THE MAGTAPE BASE REG
 3875  010442                   MS1.5:  PLINE   Q.TCU   ;GO ASK FOR A UNIT NUMBER
  (1)  010442  337                       RST     3       ;PRINT LINE OF CHARS
  (1)  010443  303  037                  .ADDR   Q.TCU   ;BUFF TO PRINT
 3876  010445  315  160  021            CALL    PICKUP  ;GO SEE WHAT WAS TYPED
 3877  010450  332  061  021            JC      MS2     ;IF NOTHING TYPE, LEAVE VALUE ALONE.. GO AROUND THIS.
 3878
 3879                           ;FALL TO HERE IF NEED TO SET A NEW UNIT NUMBER
 3880  010453  072  100  040            LDA     TMPB2   ;GET WHAT WAS TYPED
 3881  010456  062  137  040            STA     TAPEUNIT ;SET IN THE NEW UNIT NUMBER
 3882
 3883  010461                   MS2:    PLINE   Q.DEN   ;NOW GO SEE WHAT DENSITY TO SET FOR THE MAGTAPE
  (1)  010461  337                       RST     3       ;PRINT LINE OF CHARS
  (1)  010462  312  037                  .ADDR   Q.DEN   ;BUFF TO PRINT
 3884  010464  315  172  021            CALL    INBUF   ;UPDATE BUFFER POINTER. THIS CASE IS DIFFERENT THAN
 3885                                                   ;THE OTHERS.. WE NEED TO EXAMINE ASCII. NOT OCTAL
 3886  010467  332  131  021            JC      MS3     ;BUT IF NOTHING TYPED, GO ASK FOR A NEW SLAVE
```

```
 3888                          ;FALL THRU IF NEED TO SET A DENSITY....
 3889                          ;NOW H,L REG POINTS TO THE STRING JUST TYPED IN
 3890  010472  345                     PUSH    H       ;SAVE POINTER TO THE TYPED IN BUFFER
 3891  010473  021  147  021           LXI     D,EIGHT0 ;GET POINTER TO THE '800' LIST
 3892  010476  315  302  033           CALL    STRCMP  ;NOW DO A STRING COMPARE
 3893  010501  302  112  021           JNZ     S16CHK  ;IF WAS NOT AN '800', SEE IF ITS A '1600'
 3894
 3895                          ;OK, IT WAS 800..NOW SET UP THE CHANNEL DATA TO SAY 800 BPI TAPE
 3896  010504  076  002               MVI     A,2     ;A 2 IS THE CORRECT CODE FOR 800 BPI
 3897  010506  341                     POP     H       ;IF '800' MATCHED, THEN CLEAN UP THE STACK
 3898  010507  303  126  021           JMP     MS2.5   ;GO TO NEXT CHECK
 3899
 3900  010512  341             S16CHK: POP     H       ;GET POINTER TO THE TYPED IN STUFF
 3901  010513  021  153  021           LXI     D,SIXTN ;MATCH AGAINST '1600'
 3902  010516  315  302  033           CALL    STRCMP  ;DO THE STRING COMPARE
 3903  010521  302  150  032           JNZ     KILNM   ;IF WAS NOT 1600, THEN IT WAS BAD
 3904
 3905                          ;IT WAS 1600, SO SET THE RIGHT THING FOR TAPE BPI
 3906  010524  076  004               MVI     A,4     ;THIS IS THE CODE FOR 1600 BPI
 3907  010526  062  363  040   MS2.5: STA     DEN.SLV+1 ;SET THE BYTE IN THE CHANNEL DATA WORD
 3908
 3909  010531                  MS3:    PLINE   Q.SLV   ;ASK FOR A NEW SLAVE DEVICE
  (1)  010531  337                     RST     3       ;PRINT LINE OF CHARS
  (1)  010532  322  037                .ADDR   Q.SLV   ;BUFF TO PRINT
 3910  010534  315  160  021           CALL    PICKUP  ;FETCH WHAT WAS TYPED
 3911  010537  330                     RC              ;IF NOTHING TYPED, THEN WE ARE ALL DONE
 3912
 3913                          ;ELSE FALL TO HERE TO GET THE SLAVE
 3914  010540  072  100  040           LDA     TMPB2   ;GET THE NUMBER
 3915  010543  062  362  040           STA     DEN.SLV ;SET THE BYTE AS REQUIRED
 3916  010546  311                     RET             ;AND ALL DONE
 3917  010547  070  060  060   EIGHT0: .ASCIZ /800/     ;FOR 800 BPI TAPES
 3918  010553  061  066  060   SIXTN:  .ASCIZ /1600/   ;STRING FOR 1600 BPI TAPES
```

```
 3920                           ;SOME SUBROUTINES FOR USE BY THE DEVICE SELECT COMMANDS
 3921                           ;FIRST A ROUTINE TO READ IN A NUMBER TYPED IN ANSWER TO AN 8080 QUESTION
 3922                           ;AND SAVE THE NUMBER TYPED IN THE 36 BIT BUFFER 'TMPB2', RETURNS C-BIT CLEAR
 3923                           ;IF A NUBER WAS GATHERED AND STORED IN 'TMPB2', RETURNS C-BIT SET IF NOTHING
 3924                           ;WAS TYPED.
 3925  010560  315  172  021   PICKUP: CALL    INBUF   ;SET UP THE INPUT BUFFER TO THE CURRENT TYPEIN.
 3926  010563  330                    RC              ;RETURNS HERE WITH C-BIT SET IF NOTHING TYPED
 3927
 3928                           ;GET HERE IF SOMETHING WAS TYPED.. GO GET IT AND PUT IT IN 'TMPB2'
 3929  010564                         ARG36           ;GATHER A 36-BIT ARGUMENT
  (1)  010564  347                    RST     4
  (1)  010565  010                    .BYTE   8
 3930  010566  100  040              .ADDR   TMPB2   ;PUT IT IN THIS BUFFER
 3931  010570  257                    XRA     A       ;CLEAR C-BIT BECAUSE ALL WAS OK.
 3932  010571  311                    RET             ;ALL DONE
 3933
 3934                           ;SUBROUTINE TO FIX UP THE BUFFER POINTERS IN THE INPUT BUFFER
 3935  010572  041  204  040   INBUF: LXI     H,EOL   ;GET POINTER TO END-OF-LINE COUNTER
 3936  010575  065                    DCR     M       ;SO CAN DECREMENT
 3937  010576  315  160  010          CALL    BFRST   ;RESET TTY INPUT POINTERS
 3938  010601  052  215  040          LHLD    BUF.    ;FIND THE BEGINNING OF BUFFER
 3939  010604  042  223  040          SHLD    .ARG1   ;AND SET IT AS THE POINTER TO THE FIRST ARG
 3940
 3941  010607  041  215  021             LXI  H,INRDY ;PASS A RETURN ADDRESS IN H,L
 3942  010612  303  340  001             JMP  NULLW   ;ENTER TTY INPUT WAIT
 3943
 3944  010615  052  223  040   INRDY: LHLD    .ARG1   ;GET POINTER TO THE START OF THE NEW DATA
 3945  010620  315  245  033   FNDARG: CALL    SEPCHR  ;EAT UP ANY NO-OP SEPARATORS
 3946  010623  042  223  040          SHLD    .ARG1   ;REPLACE THE POINTER
 3947  010626  303  344  032          JMP     EOCML   ;CHECK IF AT END-OF-LINE. C-SET IF YES(IE NO ARG)
```

```
 3949                           .SBTTL  *** 'BT' CMD ***
 3950       000077                      INDIRECT=^O77
 3951  010631                   BOOT:   PLINE   BTMSG1  ;INCLUDE A MESSAGE SO THAT USERS KNOW U R BOOTING
  (1)  010631  337                      RST     3       ;PRINT LINE OF CHARS
  (1)  010632  140  037                 .ADDR   BTMSG1  ;BUFF TO PRINT
 3952  010634                   BTAUT:  PCRLF           ;AND KEEP IT ON ONE LINE
  (2)  010634  347                      RST     4
  (2)  010635  002                      .BYTE   2
 3953  010636  076  010                MVI     A,^O10  ;BIT 32 IN TENLAND TO SAY THIS WAS A BOOT BUTTON LOAD
 3954  010640  062  255  040           STA     GOCODE  ;SAVE IN THE 'GO CODE' PLACE
 3955  010643  067                    STC             ;FALL THRU TO A 'BT'
 3956
 3957  010644  315  326  021  BTCMD:  CALL    BTCHOICE ;GO SELECT MONITOR OR DIAG PRE-BOOT
 3958  010647  315  263  022  BT.SRC: CALL    MICROP ;READ THE PAGE OF FILE POINTERS INTO MEMORY @1000
 3959
 3960                           ;WHEN GET TO HERE, THE PAGE  HAS BEEN READ IN.
 3961  010652  332  175  032           JC      C.BTERR ;ERROR IN BOOT PROCESS DURING MICRO-CODE LOAD
 3962
 3963  010655  315  043  023           CALL    DMEM2CRAM ;LOAD DATA FROM MEMORY INTO CRAM
 3964
 3965
 3966                           ;FALL THRU IF DONE THE CRAM LOADING PORTION
 3967                           ;NOW MUST READ IN THE BOOT CODE ITSELF,START THE SM10 MICRO-CODE
 3968                           ;AND THEN START THE BOOT PROGRAM AT ADDRESS 1000
 3969                           ;NOW SET UP DISK POINTERS TO POINT TO BOOT BLOCK OF DISK,
 3970                           ;IN ORDER THAT WE LOAD THE MONITOR BOOT
 3971  010660  315  277  021  LB.GO:  CALL    LBINT   ;GO READ-IN THE APPROPRIATE BOOTSTRAP
 3972  010663                   LB.GO1: MOV5B           ;SET UP A START ADDRESS
  (1)  010663  347                      RST     4
  (1)  010664  000                      .BYTE   0
 3973  010665  244  022                 .ADDR   MA1000  ;MEMORY ADDRESS 1000 FOR STARTING PROGRAM
 3974  010667  073  040                 .ADDR   TMPBF2  ;SET UP SO START COMMAND CAN FIND THE ADDRESS
 3975
 3976                           ;TEMP CODE FOR FIGURING OUT HOW TO MAKE THE INTERNAL START CODE WORK
 3977  010671  303  113  017           JMP     STINT   ;GO START THE MACHINE WITH MONITOR BOOT
```

```
 3979                          .SBTTL  *** 'LB' CMD ***
 3980  010674  315  326  021   LBCMD:  CALL    BTCHOICE ;IF ARG GIVEN, GO SET UP A CHOICE FOR THE BOOTING
 3981  010677  021  000  002   LBINT:  LXI     D,^O1000 ;ALL POINTERS START AT 1000
 3982  010702  072  172  040           LDA     RM100   ;GET THE OFFSET AS SELECTED(MON OR DIAG PRE-BOOT)
 3983  010705  203                   ADD     E       ;ADD LO ORDER TO THE OFFSET
 3984  010706  137                   MOV     E,A     ;PUT IT BACK
 3985  010707  315  266  022           CALL    FILEINIT ;READ IN POINTERS TO THE 'PRE-BOOTS'
 3986
 3987  010712  332  231  032           JC      L.BTERR ;ERROR IN LOADING THE PRE-BOOT
 3988
 3989  010715  315  347  021           CALL    BT.GO   ;START UP THE MICRO-CODE & INTERNAL OFF
 3990                                ;NOW PASS ADDRESSES OF RH BASE & DRIVE # TO THE PRE-BOOT PROGRAMS
 3991  010720  052  132  040   INFOBT: LHLD    UNITNM  ;UNIT NUMBER INTO HL REGISTER
 3992  010723  303  074  022           JMP     PASSSRC ;ROUTINE WHICH WRITES LOCS 36,37 & 40
```

```
 3994                          ;CUTE LITTLE ROUTINE FOR SELECTING WHICH BOOT TO LOAD
 3995  010726  332  342  021   BTCHOICE: JC    LOAD4   ;IF NO ARG, SET FOR 'BOOT>' INPUT
 3996
 3997  010731  315  244  030           CALL    ARG16.  ;ASSEMBLE THE ARG
 3998                          ;AS LONG AS IT WAS A NUMBER, WE WILL LOAD THE DIAG BOOT
 3999  010734  076  006               MVI     A,6     ;A 6 IS THE OFFSET FOR THE DIAG PRE-BOOT
 4000  010736  062  172  040   EXIT4:  STA     RM100   ;SAVE IT
 4001  010741  311                   RET
 4002  010742  076  004        LOAD4:  MVI     A,4     ;PASS A 4.. TO SAY LOAD MONITOR BOOT
 4003  010744  303  336  021           JMP     EXIT4   ;AND COMMON EXIT
 4004
 4005                          ;ROUTINE TO START UP THE MACHINE(KS10), AND RE-ESTABLISH THE PARITY DEFAULT
 4006  010747  315  212  016   BT.GO:  CALL    SM1     ;START THE MICRO-CODE
 4007  010752  332  173  032           JC      D.BTERR ;IF MICRO-CODE DOES NOT START
 4008  010755                BT.GO1: INTOFF          ;INTERNAL MODE OFF
  (1)  010755  367                   RST     6       ;GO EXIT FROM INTERNAL MODE
 4009
 4010  010756  076  174               MVI     A,DEFLTE ;GET MACHINE DEFAULT VALUE FOR ENABLES
 4011  010760  315  361  013           CALL    KS.PAR  ;SET THEM INTO THE RAM & WRITE INTO KS
 4012  010763  076  020               MVI     A,TRPDEF ;GET MACHINE DEFAULT FOR TRAP ENABLES
 4013  010765  062  355  040   TP.SET: STA     TRAPEN  ;SET DEFAULT INTO THE RAM
 4014  010770  323  205               OUT     DIAG    ;***** I/O WRT 205/TRAPS ENABLE *****
 4015  010772  311                   RET             ;AND OUT
 4016
```

```
 4018                           .SBTTL  *** 'MT' CMD ***
 4019  010773  042  221  040   MTCMD:  SHLD    CMD..   ;SAVE WHAT COMMAND THIS IS, SO RETRYS WILL WORK
 4020  010776  315  224  022           CALL    MTSETUP ;GO TO SOME COMMON CODE
 4021  011001  076  071                 MVI     A,READ.TAPE ;GET THE COMMAND EXECUTION CODE FOR THE TAPE CMD
 4022  011003  315  376  024           CALL    MTXFR   ;AND READ-IN THE MICRO-CODE FROM TAPE
 4023  011006  322  017  022           JNC     MT.1    ;NO NEED TO CHECK IF FATAL ERR IF ALL OK
 4024
 4025  011011  315  154  022           CALL    NONFATAL ;WAS AN ERROR, GO SEE WHAT KIND
 4026  011014  302  201  032           JNZ     A.BTERR ;ERR TYPE 'A' IF INITIAL READ FAILS
 4027
 4028  011017  076  002        MT.1:  MVI     A,MT.BIT ;SET ACCUM .EQ. MAGTAPE BIT
 4029  011021  315  045  023           CALL    MEM2CRAM ;LOAD MICRO-CODE FROM MEMORY TO 'CRAM' SPACE
 4030
 4031  011024  315  035  022           CALL    MBINT   ;NOW LOAD IN THE PRE-BOOT PROGRAM
 4032  011027  315  347  021           CALL    BT.GO   ;START THE MICRO-CODE,REPLACE PARITY & TRAP DEFAULTS
 4033  011032  303  263  021           JMP     LB.GO1  ;AND PROCEED TO START THINGS
```

```
 4035                           ;NOW MUST DO A RE-WIND., SKIP FIRST FILE(MICRO-CODE), THEN READ-IN THE
 4036                           ;SECOND FILE(THE PRE-BOOT)
 4037  011035  076  031        MBINT:  MVI     A,SKP.TAPE ;GRAB A SKIP COMMAND
 4038  011037  315  376  024          CALL    MTXFR   ;ISSUE A REWIND, AND A FILE-SKIP
 4039
 4040                           ;NOW WE EXPECT THERE TO BE A FRAME COUNT ERROR FROM THE SPACE FORWARD
 4041                           ;AND WE WILL DO WHAT WE CAN TO IGNORE IT
 4042  011042  322  053  022          JNC     MTSKOK  ;IF NO ERROR AT ALL, THATS OK TOO
 4043
 4044  011045  315  154  022          CALL    NONFATAL ;CHECK ERROR TYPE IF FALL INTO HERE
 4045  011050  302  231  032          JNZ     L.BTERR ;IF WAS NOT A FRAME COUNT ERROR, IT WAS MORE SERIOUS
 4046
 4047                           ;IF COMPARE RESULT WAS ZERO, THEN THE ERROR WAS A FRAME COUNT ERROR
 4048                           ;AND WE WILL IGNORE IT BY FALLING INTO THE CONTINUE CODE
 4049  011053  076  071        MTSKOK: MVI     A,READ.TAPE ;GET A TAPE READ COMMAND
 4050  011055  315  007  025          CALL    QMXFR   ;EXECUTE TAP COMMAND LIST WITH NO REWIND IN IT
 4051  011060  322  071  022          JNC     PASSME  ;NO ERROR IF NO 'C' BIT
 4052
 4053  011063  315  154  022          CALL    NONFATAL ;SEE WHAT KIND OF ERROR
 4054  011066  302  231  032          JNZ     L.BTERR ;BOOT ERROR IF NO Z
 4055
 4056  011071  052  137  040   PASSME: LHLD    TAPEUNIT ;GET TAPE UNIT FOR CURRENT MAGTAPE SELECTION
```

```
 4058                          ;NOW PASS INFO IN LO MEMORY ADDRESS SPOTS
 4059  011074  345            PASSSRC:  PUSH   H       ;SAVE THE PASSED UNIT NUMBER ON TOP OF STACK
 4060  011075                          MOV5B           ;PASS RH BASE ADDRESS TO INTERNAL BUFFER
  (1)  011075  347                     RST    4
  (1)  011076  000                     .BYTE  0
 4061  011077  242  040               .ADDR  RHBASE  ;FROM HERE
 4062  011101  047  040               .ADDR  DMDAT   ;TO HERE
 4063  011103  041  051  040          LXI    H,DMDAT+2 ;GET POINTER TO PIECE FOR UBA
 4064  011106  072  254  040          LDA    UBANUM  ;GET CURRENT UBA
 4065  011111  266                    ORA    M       ;PUT IT INTO MEMORY
 4066  011112  167                    MOV    M,A
 4067
 4068  011113  345                    PUSH   H       ;SAVE THE POINTER TO 'DMDAT' AREA
 4069  011114                         DEPOS  36      ;DEPOSIT IN MEMORY
  (1)  011114  247                     ANA    A       ;CLR 'C-BIT' FOR USE BY COMMON CODE
  (1)  011115  315  217  030          CALL   DEPSHT  ;AND DO THE DEPOSIT ASSUMING SHORT ADDR
  (1)  011120  036  000               .ADDR  36      ;ADDR TO BE ZAPPED PASSED AS TRAILING ARG
 4070  011122  341                    POP    H       ;GET BACK THE POINTER TO 'DMDAT' AREA
 4071  011123  066  000               MVI    M,0     ;CLEAR BYTE WITH BITS 12-19
 4072  011125  341                    POP    H       ;GET THE UNIT NUMBER THAT WAS SAVED ON THE STACK
 4073  011126  042  047  040          SHLD   DMDAT   ;AND PUT IT INTO THE DEPOSIT MEMORY DATA AREA
 4074  011131                         DEPOS  37      ;DEPOSIT IN MEMORY HERE
  (1)  011131  247                     ANA    A       ;CLR 'C-BIT' FOR USE BY COMMON CODE
  (1)  011132  315  217  030          CALL   DEPSHT  ;AND DO THE DEPOSIT ASSUMING SHORT ADDR
  (1)  011135  037  000               .ADDR  37      ;ADDR TO BE ZAPPED PASSED AS TRAILING ARG
 4075
 4076  011137                         MOV5B          ;FINALLY PASS DENSITY SLAVE INFORMATION
  (1)  011137  347                     RST    4
  (1)  011140  000                     .BYTE  0
 4077  011141  362  040               .ADDR  DEN.SLV ;GET IT FROM HERE
 4078  011143  047  040               .ADDR  DMDAT   ;PUT IT HERE
 4079  011145                         DEPOS  40      ;AND MOS MEMORY HERE
  (1)  011145  247                     ANA    A       ;CLR 'C-BIT' FOR USE BY COMMON CODE
  (1)  011146  315  217  030          CALL   DEPSHT  ;AND DO THE DEPOSIT ASSUMING SHORT ADDR
  (1)  011151  040  000               .ADDR  40      ;ADDR TO BE ZAPPED PASSED AS TRAILING ARG
 4080  011153  311                    RET            ;BACK TO CALLER
```

```
 4082                          ;ROUTINE THAT CHECKS TO SEE WHAT KIND OF ERROR WE HAVE SUFFERED UNDER
 4083                          ;THE MAGTAPE TRANSFER
 4084  011154  076  343       NONFAT: MVI     A,<^O377&FRMERR+2> ;CHK ERROR CODE FOR FATAL OR NON FATAL TYPES
 4085  011156  041  154  040         LXI     H,ERRCD ;NOW POINT TO THE ACTUAL ERROR TYPE THAT WE GOT
 4086  011161  276                   CMP     M       ;COMPARE THE TWO
 4087  011162  365                   PUSH    PSW     ;SAVE FLAGS WHILE WE RESET THE TAPE DRIVE
 4088  011163  314  015  025         CZ      MTRESET ;GO RESET ANY ERRORS ENCOUNTERED IN THE SKIP OPERATION
 4089  011166  361                   POP     PSW     ;GET BACK THE FLAGS
 4090  011167  310                   RZ              ;ONLY RETURN IF ERROR WAS NON FATAL
 4091
 4092                          ;FALL TO HERE IF ERR WAS FATAL TYPE..SEE IF WE CAN RETRY IT
 4093  011170  076  340               MVI     A,<^O377&RETRY.+2> ;'RETRYABLE' ERROR??
 4094  011172  276                   CMP     M       ;COMPARE
 4095  011173  300                   RNZ             ;OUT IF CAN'T EVEN RETRY..DIE
 4096
 4097                          ;AND HERE IF IT WAS RETRYABLE
 4098  011174  061  000  044         LXI     SP,RAMST+^O2000 ;FIRST CLEAR THE STACK
 4099  011177  041  344  003         LXI     H,NORML ;PUT A RETURN ADDRESS ONTO THE STACK
 4100  011202  345                   PUSH    H
 4101
 4102  011203  052  221  040         LHLD    CMD..   ;NOW GET 'WHICH' COMMAND TO RETRY
 4103  011206  351                   PCHL            ;GIVE IT A GO
```

```
 4105                           .SBTTL  *** 'MB' CMD ***
 4106                           ;COMMAND TO LOAD ONLY THE BOOTSTRAP OFF OF THE CURRENTLY SELECTED MAGTAPE
 4107  011207  042  221  040   MBCMD:  SHLD    CMD..   ;SAVE WHICH COMMAND THIS IS
 4108  011212  315  224  022           CALL    MTSETUP ;GO TO SOME COMMON CODE TO SET UP FOR MAGTAPE XFER
 4109  011215  315  035  022           CALL    MBINT   ;AND GO
 4110  011220  315  347  021           CALL    BT.GO   ;START UP THE MICRO-CODE & INTERNAL OFF
 4111  011223  311                    RET             ;BACK TO NULL JOB LOOP
 4112
 4113                           ;SOME COMMON CODE THAT SETS UP PARAMETERS FOR MAGTAPE XFER'S..SAVES A FEW
 4114                           ;BYTES OF 8080 SPACE
 4115  011224  315  015  023   MTSETUP: CALL   BTINT   ;FIRST SET UP FOR THE BOOTING PROCESS
 4116  011227  072  356  040           LDA     MTAUBA  ;GET SELECTED UBA FOR MAGTAPE
 4117  011232  062  254  040           STA     UBANUM  ;PASS TO COMMON SPOT FOR CHANNEL COMMAND LIST TO FIND
 4118
 4119  011235                         MOV5B           ;AND MOVE THE SELECTED MT BASE FOR RH BASE TO FIND
  (1)  011235  347                    RST     4
  (1)  011236  000                    .BYTE   0
 4120  011237  367  040               .ADDR   MTBASE  ;SELECTED MAGTAPE RH BASE ADDRESS
 4121  011241  242  040               .ADDR   RHBASE  ;COMMON RH BASE REGISTER LOCATION
 4122  011243  311                    RET             ;BACK TO MAILINE
 4123
```

```
 4125  011244                MA1000: D 0,0,,1,000
 4126  011251                HOMEWD: D 505,755,,000,000 ;'HOM' MEANS HOME BLOCK
 4127  011256                ONES:   D 777,777,,777,777 ;ALL ONES...
 4128                        .SBTTL  FILE SYSTEM
 4129                        ;--PAGE OF POINTERS FORMAT--
 4130                        ;       +0      POINTER TO FREE
 4131                        ;       +1      LENGTH OF FREE
 4132
 4133                        ;       +2      POINTER TO MICRO-CODE
 4134                        ;       +3      LENGTH OF MICRO-CODE
 4135
 4136                        ;       +4      POINTER TO MONITOR PRE-BOOT
 4137                        ;       +5      LENGTH OF PRE-BOOT
 4138
 4139                        ;       +6      POINTER TO DIAG PRE-BOOT
 4140                        ;       +7      LENGTH OF SAME
 4141
 4142                        ;       +10     POINTER TO BC1 MICRO-CODE
 4143                        ;       +11     LENGTH OF SAME
 4144
 4145                        ;       +12     POINTER TO BC2 PRE-BOOT
 4146                        ;       +13     LENGTH
 4147
 4148                        ;       +14     POINTER TO MONITOR BOOT PROGRAM
 4149                        ;       +15     LENGTH OF SAME
 4150
 4151                        ;       +16     POINTER TO DIAGNOSTIC BOOT
 4152                        ;       +17     LENGTH OF SAME
 4153
 4154                        ;       +20     POINTER TO BC2 ITSELF
 4155                        ;       +21     LENGTH OF SAME
 4156
 4157                        ;       +22     POINTER TO FI-ABLE 0
 4158                        ;       +23     LENGTH OF SAME
 4159                        ;       .
 4160                        ;       .
 4161                        ;       .
 4162                        ;       +776    POINTER TO FI-ABLE 366(8)
 4163                        ;       +777    LENGTH OF SAME
 4164
```

```
 4166                           ;ROUTINE TO 'FIND THE 8080 FILE SYSTEM', WHICH IS REALLY JUST A PAGE
 4167                           ;OF PHYSICAL POINTERS, TO PHYSICAL DISK LOCATIONS
 4168  011263  021  002  002    MICROP: LXI     D,^O1002 ;FOR MICROCODE, ALWAYS GO TO 2ND POINTER
 4169  011266  325              FILEINIT: PUSH  D       ;SAVE POINTER INTO THE FILE PAGE
 4170  011267  315  015  023            CALL    BTINT   ;AND SET UP TO DO A READIN
 4171  011272  321                    POP     D       ;RESTORE 'D,E'..KLUDGY WAY TO MAKE SUBROUTINE HAVE
 4172  011273  325              FILESH: PUSH    D       ;   MULTIPLE ENTRY POINTS
 4173  011274  315  026  023            CALL    DSKDFT  ;FETCH CURRENT DISK DEFAULTS
 4174
 4175  011277  041  000  000            LXI     H,00    ;CLR 'H,L'
 4176  011302  042  112  040            SHLD    BLKADR  ;AND SET THE DESIRED CYLINDER TO 00
 4177  011305  043                    INX     H       ;BUMP H,L TO MAKE IT .EQ. 01
 4178  011306  042  105  040            SHLD    BLKNUM  ;NOW SET THIS INTO THE BLOCK NUMBER(HOME BLOCK)
 4179
 4180  011311  315  372  022            CALL    CHKHOM  ;GO SEE IF THIS PAGE HAS THE 'HOM' BLK I.D.
 4181  011314  312  332  022            JZ      GOODPK  ;IF YES, JUMP TO CONTINUE READ-IN
 4182
 4183                           ;FALL THRU IF FIRST HOME BLOCK NO GOOD
 4184  011317  076  010                MVI     A,^O10  ;TRY ALTERNATE HOME BLOCK
 4185  011321  062  105  040            STA     BLKNUM  ;SET BLOCK NUMBER TO ALTERNATE
 4186
 4187  011324  315  372  022            CALL    CHKHOM  ;TRY ALTERNATE
 4188  011327  302  201  032            JNZ     A.BTERR ;IF THIS ONES BAD, THEN GIVE UP
 4189  011332                  GOODPK: EXAM    1103    ;EXAMINE WORD WITH HOME BLOCK IN IT
  (1)  011332  315  216  030            CALL    EXAMSH  ;AND DO EXAM ASSUMING SHORT ADDRESS
  (1)  011335  103  002                .ADDR   1103    ;ADDR TO BE ZAPPED PASSED AS TRAILING ARG
```

```
 4191                           ;NOW THAT YOUVE READ HOME BLOCK, FIND POINTER FILE
 4192                           ;AND TRANSFER THE ENTIRE MICRO-CODE INTO MOS MEMORY
 4193                           ;SHORT ROUTINE TO MOVE DATA FROM THE EMBUF INTO THE CHANNEL COMMAND LIST
 4194  011337  315  352  022           CALL    BLKRDR  ;READ IN THE PAGE OF POINTERS FROM THE DISK
 4195  011342  332  177  032           JC      B.BTERR ;ERROR IN BOOT PROCESS DURING POINTER PAGE READ-IN
 4196
 4197                           ;NOW EXAM THE REAL DESIRED DATA
 4198  011345  341                     POP     H       ;FETCH UP THE FILE POINTER ADDRESS
 4199  011346  067                     STC             ;SET SIGN WHICH INDICATES AND EXAMINE
 4200  011347  315  225  030           CALL    EXMHL   ;AND READ IT IN TO MEM
 4201
 4202                           ;SHORT ROUTINE TO MOVE DATA FROM THE EMBUF INTO THE CHANNEL COMMAND LIST
 4203  011352  052  015  040   BLKRDR: LHLD    EMBUF+3 ;GET CYLINDER FROM SPECIAL HOME BLK PNTR
 4204  011355  042  112  040           SHLD    BLKADR  ;SET CYLINDER IN CHANNEL COMMAND LIST
 4205  011360  052  012  040           LHLD    EMBUF   ;GET TRACK SECTOR BYTE
 4206  011363  042  105  040           SHLD    BLKNUM  ;SET INFO INTO THE TRCK/SECTOR WORD
 4207
 4208  011366  315  367  024           CALL    DSXFR   ;FINALLY READ THE FIRST PAGE OF THE DESIRED
 4209  011371  311                     RET
 4210
 4211                           ;LITTLE ROUTINE TO READ IN THE HOME BLOCK, CHECK THAT IT IS A HOME BLOCK
 4212                           ; VIA THE 'HOM' ID, AND RETURN Z-BIT SET IF IT IS
 4213  011372  315  367  024   CHKHOM: CALL    DSXFR   ;EXECUTE DISK TRANSFER
 4214  011375  332  201  032           JC      A.BTERR ;BOOT ERROR 'A', IF OOPS
 4215
 4216  011400                         EXAM    1000    ;NOW EXAMINE THE HOME BLOCK ID
  (1)  011400  315  216  030           CALL    EXAMSH  ;AND DO EXAM ASSUMING SHORT ADDRESS
  (1)  011403  000  002                .ADDR   1000    ;ADDR TO BE ZAPPED PASSED AS TRAILING ARG
 4217  011405  315  022  033           CALL    CMP36   ;AND TRY OUT A COMPARE
 4218  011410  251  022                .ADDR   HOMEWD  ;EXPECTED ID
 4219  011412  012  040                .ADDR   EMBUF   ;AGAINST WHAT WAS JUST READ IN
 4220  011414  311                     RET             ;AND OUT
 4221
 4222  011415                 BTINT:  INTON           ;SET INTERNAL MODE ON
  (1)  011415  327                     RST     2       ;GO SET INTERNAL MODE
 4223  011416                         CLRB    PARBT   ;NO PARITYS
 4224  011420                         CLRB    TRAPEN  ;AND NO TRAPS WHILE BOOTING
 4225  011422  315  225  005           CALL    MRCMD   ;AND DONT FORGET MR, RESET
 4226  011425  311                     RET
 4227
 4228                           ;LITTLE ROUTINE TO SET UP DISK DEFAULTS
 4229  011426  072  357  040   DSKDFT: LDA     DSKUBA  ;GET CURRENTLY SELECTED DISK UBA NUMBER
 4230  011431  062  254  040           STA     UBANUM  ;SET INTO COMMAND LIST PLACE
 4231
 4232  011434                         MOV5B           ;AND SEND CURRENTLY SELECTED RHBASE
  (1)  011434  347                     RST     4
  (1)  011435  000                     .BYTE   0
 4233  011436  374  040                .ADDR   DSBASE  ;CURRENT DISK RH BASE
 4234  011440  242  040                .ADDR   RHBASE  ;TO PLACE FOR COMMAND LIST TO FIND IT
 4235  011442  311                     RET             ;OUT
```

```
 4237                           .SBTTL  CRAM LOADER CODE
 4238  011443  076  001        DMEM2CR: MVI     A,BT.BIT ;DO COUPLE ROUTINES A FAVOR, LOAD BIT
 4239  011445  062  121  040   MEM2CR: STA     BT.TYPE ;ACCUM HAD THE BOOTING TYPE..SAVE IT
 4240  011450  041  000  000           LXI     H,00    ;ZEROES TO 'H,L'
 4241  011453  345                   PUSH    H       ;SAVE CURRENT CRAM ADDRESS
 4242  011454  315  320  014          CALL    CADWR   ;AND THEN WRITE IT TO THE CRAM
 4243  011457  076  007              MVI     A,7     ;START WITH FUNCTION 7
 4244  011461  062  010  040          STA     CRMFN
 4245  011464                 NEWPAG: MOV5B           ;INITIALIZE MEM ADDRESS
  (1)  011464  347                   RST     4
  (1)  011465  000                   .BYTE   0
 4246  011466  244  022              .ADDR   MA1000  ;WITH 1000  OCTAL
 4247  011470  017  040              .ADDR   MEMAD   ;STANDARD MEM ADDRESS
 4248
 4249                           ;CODE TO DECIDE IF WE NEED TO READ AN ADDITIONAL DISK SECTOR
 4250  011472  052  017  040          LHLD    MEMAD   ;GET CURRENT MOS MEM ADDRESS
 4251  011475  175             RD.EXM: MOV     A,L     ;GET 8 BITS OF ADDRESS TO BE EXAMINED
 4252  011476  323  103              OUT     A2835   ;SET PIECE OF ADDRESS INTO ADDRESS REGISTER
 4253  011500  174                   MOV     A,H     ;GET COUPLE MORE BITS
 4254  011501  323  105              OUT     A2027   ;SET INTO CSL BOARD ADDRESS REGISTER
 4255  011503  076  004              MVI     A,4     ;SPECIAL KEY TO MAKE 'EXAMINE' WORK CORRECTLY
 4256  011505  315  055  012          CALL    EM.CRM  ;GO DO A MEMORY EXAMINE, OF THE SHORT FLAVOR
 4257
 4258  011510  052  012  040          LHLD    EMBUF   ;GET 16 BITS OF THE MEMORY DATA
 4259  011513  174                   MOV     A,H     ;COPY TO ACCUM, SO THAT WE CAN MAKE IT 12 BITS
 4260  011514  346  017              ANI     ^O17    ;CLR THE BITS
 4261  011516  147                   MOV     H,A     ;PUT BACK INTO THE HI ORDER REG
 4262
 4263  011517  315  263  014          CALL    WFUNC   ;WRITE THE PIECE
 4264  011522  041  010  040          LXI     H,CRMFN ;GET CURRENT DIAG FUNCTION
 4265  011525  065                   DCR     M       ;DOWN COUNT
 4266
 4267  011526  052  013  040          LHLD    EMBUF+1 ;GET 16 BITS OF THE MEMORY READ
 4268
 4269  011531  016  004              MVI     C,4     ;NOW A QUICK LITTLE LOOP
 4270  011533  257                   XRA     A       ;CLR ACCUM TEMP
 4271  011534  174             QQLOOP: MOV     A,H     ;COPY TO ACCUM
 4272  011535  037                   RAR             ;ROTATE INTO THE C-BIT
 4273  011536  147                   MOV     H,A     ;PUT IT BACK
 4274
 4275  011537  175                   MOV     A,L     ;TRY BOTTOM PIECE
 4276  011540  037                   RAR             ;ROTATE C-BIT INTO THE TOP
 4277  011541  157                   MOV     L,A     ;PUT IT BACK
 4278  011542  015                   DCR     C       ;DOWN THE LITTLE COUNTER
 4279  011543  302  134  023          JNZ     QQLOOP  ;CONTINUE
 4280
 4281  011546  315  263  014          CALL    WFUNC   ;WRITE THIS PIECE
 4282  011551  041  010  040          LXI     H,CRMFN ;DOWN THE FUNCTION COUNTER
 4283  011554  065                   DCR     M       ;DECREMENT
 4284  011555  362  203  023          JP      BBLOOP  ;JUMP AROUND THIS STUFF IF NOT AT FUNCTION 0
```

```
 4286                          ;IF DONE FUNCTIONS 0-7, TRY A LITTLE RESET
 4287  011560  076  007               MVI     A,7     ;RESTART AT FUNCTION 7
 4288  011562  062  010  040          STA     CRMFN   ;SAVE IT
 4289
 4290                          ;HERE IF FINSHED A CRAM WORD & NEED TO DO RESET.
 4291  011565  341                     POP     H       ;GET CRAM ADDRESS
 4292  011566  043                     INX     H       ;UPDATE
 4293  011567  345                     PUSH    H
 4294  011570  315  320  014          CALL    CADWR   ;NOW WRITE THIS, THE NEXT CRAM ADDRESS
 4295  011573  174                     MOV     A,H     ;NOW GET HI ORDER PIECE OF CRAM ADDR
 4296  011574  346  010               ANI     ^O10    ;IS IT .EQ. 4000 OCTAL YET??
 4297  011576  312  221  023          JZ      SEEPAGE ;IF NOT 4000 OCTAL YET, CHECK FOR A NXT WORD
 4298  011601  341                     POP     H       ;AND RESTORE STACK BEFORE LEAVING
 4299  011602  311                     RET             ;OTHERWISE ALL DONE
 4300
 4301  011603  052  015  040  BBLOOP: LHLD    EMBUF+3 ;GRAB 16 BITS OF THE MEMORY DATA
 4302  011606  174                     MOV     A,H     ;PASS 8 BITS TO THE ACCUM, SO WE CAN MAKE IT 4 BITS
 4303  011607  346  017               ANI     ^O17    ;OFF UNNEEDED BITS
 4304  011611  147                     MOV     H,A     ;PUT IT BACK
 4305  011612  315  263  014          CALL    WFUNC   ;WRITE THIS DATUM
 4306  011615  041  010  040          LXI     H,CRMFN ;GET FUNCTION
 4307  011620  065                     DCR     M       ;DOWN TO NEXT FUNCTION
 4308
 4309  011621  052  017  040  SEEPAGE: LHLD   MEMAD   ;GET THE CURRENT MEMORY ADDRESS
 4310  011624  043                     INX     H       ;GO TO NEXT ADDRESS
 4311  011625  042  017  040          SHLD    MEMAD   ;SAVE THIS NEXT ADDRESS
 4312  011630  174                     MOV     A,H     ;TEST H FOR AT '2000'
 4313  011631  346  004               ANI     ^O4     ;IF '2000' WEIGHT BIT IS SET, TIME FOR NEW PAGE OF DATA
 4314  011633  312  075  023          JZ      RD.EXM  ;IF MEM ADDRESS .EQ. 2000, THEN FALL THRU TO NEXT READIN
 4315  011636  315  244  023          CALL    NEXTCR  ;ROUTINE TO FETCH NEXT PAGE OF CRAM DATA
 4316  011641  303  064  023          JMP     NEWPAG  ;AND BACK TO BEGINNING
 4317
 4318                          ;ELSE, MUST READ IN ANOTHER PAGE'S WORTH FROM CURRENT BOOT DEVICE
 4319  011644  072  121  040  NEXTCR: LDA     BT.TYPE ;FIND OUT WHAT KIND OF DEVICE WE ARE BOOTING FROM
 4320  011647  376  001               CPI     BT.BIT  ;SEE IF DOING BOOT FROM THE DISK
 4321  011651  302  266  023          JNZ     TAPDEV  ;IF FLAG .NE. BT.BIT, THEN BOOTING FROM MAGTAPE
 4322
 4323                          ;FALL THRU TO DO BOOT FROM DISK
 4324  011654  041  146  036          LXI     H,QXFR  ;SET UP FOR THE QUICK XFR COMMAND LIST
 4325  011657  315  023  025          CALL    CHNXCT  ;READ IN, SHORT FORMAT(TRK/SEC/CYL IS PRESET)
 4326  011662  332  175  032          JC      C.BTERR ;ERROR IN READING CRAM
 4327  011665  311                     RET             ;BACK TO CALLER
 4328
 4329                          ;JUMP TO HERE TO DO BOOT FROM TAPE
 4330  011666  076  071       TAPDEV: MVI     A,READ.TYPE ;PASS TO CHANNEL CMD LIST EXECUTOR THE XFR TYPE
 4331  011670  315  007  025          CALL    QMXFR   ;READ IN A SINGLE PAGE FROM MAGTAPE
 4332  011673  320                     RNC             ;IF NOTHING BAD, RETURN
 4333  011674  315  154  022          CALL    NONFATAL ;IF BADNESS, SEE IF FATAL KIND OR NOT
 4334  011677  302  175  032          JNZ     C.BTERR ;ERROR IN READING THE CRAM
 4335  011702  311                     RET             ;BACK TO CALLER
```

```
 4337                           .SBTTL  *** 'FI' CMD ***
 4338                           ;COMMAND TO READ IN A PARTICULAR PAGE OF THE FILESYSTEM AND TO EXECUTE ITS
 4339                           ;CONTENTS AS IF THEY WERE TYPED IN 8080 COMMANDS
 4340  011703  315  244  030   FICMD: CALL    ARG16,  ;COLLECT IT WHEN ITS THERE
 4341
 4342                           ;BEWARE..IF YOU TYPE FI WITH A BOGUS ARGUMENT, THEN YOU LOSE..
 4343                           ;HE WHO USES THE FI CMD BEST KNOW WHAT HE'S DOING
 4344  011706  021  022  002           LXI     D,^O1022 ;START WITH OFFSET '0' INTO THE 'FI' FILES
 4345  011711  031                     DAD     D       ;ADD THIS TO THE NUMBER TYPED TO GET THE DESIRED
 4346  011712  315  266  022           CALL    FILEINIT ;READ IN THE DESIRED PAGE
 4347  011715  332  231  032             JC    L.BTERR ;IF ERR
 4348
 4349                           ;NOW DO SOMETHING WITH THE STUFF TYPED...
 4350  011720                           MOV5B           ;FIRST MUST GET INFO FROM MOS MEM TO 8080 RAM
  (1)  011720  347                     RST     4
  (1)  011721  000                     .BYTE   0
 4351  011722  244  022               .ADDR   MA1000  ;INFO STARTS AT MOS MEME LOC 1000
 4352  011724  017  040               .ADDR   MEMAD   ;AND WE WILL TELL INTERNAL READER TO START THERE
 4353
 4354  011726  021  253  041           LXI     D,E.BEG+2 ;A PLACE TO STORE ASCII BYTES FROM MEMORY
 4355  011731                  FI.GET: INTON           ;NO PRINTING
  (1)  011731  327                     RST     2       ;GO SET INTERNAL MODE
 4356  011732  315  232  024           CALL    GATHER  ;GO READ IN A WORD FROM MOS MEM
 4357  011735                           INTOFF          ;OK TO PRINT NOW
  (1)  011735  367                     RST     6       ;GO EXIT FROM INTERNAL MODE
 4358  011736  056  004               MVI     L,4     ;NOW ONLY 4 BYTES PER WORD ARE USEFUL
 4359  011740  001  012  040           LXI     B,EMBUF ;AND THIS IS WHERE IN 8080 RAM THE BYTES ARE
 4360  011743  012             FI.MOV: LDAX    B       ;FETCH UP A BYTE
 4361  011744  022                     STAX    D       ;PUT IT IN PLACE
 4362  011745  376  377               CPI     ^O377   ;IS IT END OF STRING??
 4363  011747  302  360  023           JNZ     FI.NXT  ;IF NO, GO MOVE SOME MORE
 4364
 4365  011752  315  351  034           CALL    MV.ALL  ;GOT HERE, MOVE TO EXECUTE BUFFER
 4366  011755  303  222  002           JMP     DCODE   ;AND GO DO IT
 4367
 4368                           ;NOPE.. MOVE SOME MORE
 4369  011760  003             FI.NXT: INX     B       ;UPDATE THE POINTERS
 4370  011761  023                     INX     D
 4371  011762  055                     DCR     L       ;CHECK COUNT TO SEE IF WE'VE DONE 4 YET
 4372  011763  302  343  023           JNZ     FI.MOV  ;IF NOT, GO MOVE THE NEXT BYTE FROM THE CURRENT WORD
 4373  011766  303  331  023           JMP     FI.GET  ;YES, READ NEXT MEM WORD AND TRY IT
```

```
                                ;;**********PROPOSED INSTRUCTION ********************
                                ;;.SBTTL       *** 'B1' CMD ***
                                ;;COMMAND TO READ IN A SECOND TYPE OF MICRO-CODE AND EXECUTE IT..
                                ;;I.E. BOOTCHECK 1 MICRO-CODE
                                ;;B1CMD:       LXI     D,^O1010 ;GET THE CORRECT OFFSET
                                ;;      CALL    FILEINIT ;READ IN THE FIRST PAGE
                                ;;      JC      C,BTERR ;IF ERR
                                ;;
                                ;;      MVI     A,BT.BIT ;SAY THAT THIS IS A LOAD FROM DISK
                                ;;      CALL    MEM2CRAM ;READ IT IN AS MICRO-CODE
                                ;;      CALL    BT.GO   ;START IT UP
                                ;;      RET             ;THAT IT
                                .SBTTL  *** 'B2' CMD ***
                                ;BOOTCHECK 2.. THIS LOADS IN A SEPARATE 'PRE-BOOT', WHICH
                                ;LOADS IN THE BOOT CHECK 2
011771  076  012        B2CMD:  MVI     A,^O12  ;GET THE OFFSET
011773  062  172  040          STA     RM100   ;SAVE IT SO THAT WE CAN USE SOME SUPER COMMON CODE
011776  303  247  021          JMP     BT.SRC  ;THAT'S IT!!!!
```

```
4394                            .SBTTL  *** 'VD' CMD ***
4395                            ;COMMAND TO VERIFY THE CONTENTS OF THE C-RAM AGAINST THE MICRO-CODE
4396                            ;AS IT SITS ON DISK
4397  012001  315  263  022     VDCMD:  CALL    MICROP ;NOW READ IN HOME BLOCKS, THEN 1ST PAGE OF U-CODE
4398  012004  332  175  032             JC      C,BTERR ;IF ERROR, GO TELL WORLD ABOUT IT
4399
4400  012007  076  001                   MVI     A,BT.BIT ;WENT OK., SPECIFY A DISK TYPE OPERATION
4401  012011  303  031  024             JMP     VERCRAM ;AND GO IN TO VERIFY THE CRAM
4402
4403                            .SBTTL  *** 'VT' CMD ***
4404                            ;COMMAND TO VERIFY THE CONTENTS OF THE C-RAM AGAINST THE MICR-CODE
4405                            ;AS IT SITS ON MAG TAPE
4406  012014  315  224  022     VTCMD:  CALL    MTSETUP ;GO TO SOME COMMON CODE TO LOOK AT MAGTAPES
4407  012017  076  071                   MVI     A,READ.TAPE ;TELL CHANNEL LISTER TO DO A READ IN
4408  012021  315  376  024             CALL    MTXFR   ;READ IN FIRST PAGE OF U-CODE OFF THE TAPE
4409  012024  332  201  032             JC      A,BTERR ;IF ERROR, GO REPORT IT
4410
4411  012027  076  002                   MVI     A,MT.BIT ;WENT OK., SPECIFY A TAPE OPERATION
4412                            ;FALL STRAIGHT INTO THE CODE TO VERIFY THE CRAM
```

```
                                .SBTTL  VERIFY CRAM CODE
                                ;ROUTINES THAT DO READ INS FROM THE CURRENTLY SELECTED DEVICE
                                ;AND COMPARES THAT HARD MICRO-CODE DATA AGAINST THE CURRENT
                                ;CONTENTS OF THE CONTROL-STORE
012031                          VERCRAM:
012031  062  121  040                 STA     BT.TYPE ;BEGIN BY SAVING THE DEVICE AGAINST WHICH WE WILL VERIFY

012034                              MOV5B           ;START BY SETTING MEMORY ADDRESS AT 1000
 (1)  012034  347                   RST     4
 (1)  012035  000                   .BYTE   0
012036  244  022                   .ADDR   MA1000  ;A '1000'
012040  017  040                   .ADDR   MEMAD   ;PLACE WHERE MEM ADDRESS IS KEPT
012042  041  000  000               LXI     H,00    ;BEGIN WITH CRAM ADDRESS 00
012045  042  004  040               SHLD    CRMAD   ;SET CRAM ADDRESS TO ZEROES
012050  303  063  024               JMP     V.GO    ;ENTER LOOP AT THE PROPER PLACE

012053  052  004  040  V.DONWD: LHLD   CRMAD   ;GET CURRENT CRAM ADDRESS
012056  043                       INX     H       ;AND UPDATE FOR NEXT TIME AROUND
012057  174                       MOV     A,H     ;PUT HI ORDER PIECE OF IT INTO ACCUM
012060  346  010                   ANI     ^O10    ;SEE IF AT END OF CRAM YET
012062  300                       RNZ             ;IF IT IS, ALL DONE, GO OUT

                                ;HERE IF REALLY READY TO DO A CRAM LOC
012063  315  320  014  V.GO:   CALL    CADWR   ;WRITE IT TO CRAM, BE IT GOOD OR BAD
012066  042  004  040               SHLD    CRMAD   ;NOW SAVE ADDRESS WHILE WE DO SOME STUFF
012071  315  204  013               CALL    CP1     ;SINGLE CLOCK GETS CRAM CONTENTS TO CONTROL REG
012074  315  021  015               CALL    RCINT   ;READ IN CONTENTS OF C-RAM AND SAVE IN 8080 RAM

                                ;HERE WHEN CRAM DATA IS SAFELY TUCKED AWAY IN THE 8080 RAM
012077  001  205  024               LXI     B,VERLST ;B,C PAIR POINTS TO LIST OF DATA OFFSETS
012102  021  302  040               LXI     D,CRMBF ;D,E PAIR POINTS TO ACTUAL DATA LIST(H,L PNTS EXPECTED)

012105  315  232  024  V.NXMEM: CALL   GATHER  ;HERE TO CALL ROUTINE THAT READS IN THE NEXT MEM WORD
012110  076  003                   MVI     A,3     ;AND TAKE TIME OUT TO RESET THE 3 COUNTER
012112  062  253  040               STA     VERCNT  ;SET UP A CLEAN COUNT
```

```
 4448                          ;AND HERE BELOW BEGINS THE ACTUAL DATA COMPARES
 4449  012115  012             V.BLP:  LDAX    B       ;GET THE FIRST INDEX BYTE FROM THE LIST OF BYTES
 4450  012116  346  077                 ANI     ^O77    ;OFF THE SIGNALS, AND CONTINUE
 4451  012120  052  012  040   V.BLP1: LHLD    EMBUF   ;GET EXPECTED DATA INTO H,L REGISTER
 4452  012123  003                     INX     B       ;UPDATE B,C TO POINT AT FUNCTION READ
 4453  012124  203                     ADD     E       ;ADD AS AN OFFSET TO THE D,E PAIR
 4454  012125  137                     MOV     E,A     ;PUT THE GOOD ADDR BACK INTO 'E'
 4455  012126  172                     MOV     A,D     ;AND GRAB THE HI ORDER FOR A SEC
 4456  012127  316  000                 ACI     0       ;ADD IN THE CARRY IF REQUIRED
 4457  012131  127                     MOV     D,A     ;PUT THE HI ORDER BACK AGAIN
 4458
 4459  012132  032                     LDAX    D       ;NOW LOAD IN THE FIRST 'ACTUAL' DATUM
 4460  012133  023                     INX     D       ;UPDATE POINTER TO ACTUAL
 4461  012134  275                     CMP     L       ;COMPARE AGAINST EXPECTED
 4462  012135  302  146  024          JNZ     V.ERR   ;REPORT IF BADNESS
 4463
 4464  012140  174                     MOV     A,H     ;GET UPPER 12 BITS OF THE EXPECTED
 4465  012141  346  017                 ANI     ^O17    ;IF DATA WAS EQUAL, ONLY DISCREPENCY CAN BE IN B7-B4
 4466
 4467  012143  147                     MOV     H,A     ;GET THE 4 BITS OF DATA LEFT AFTER THE 'AND'
 4468  012144  032                     LDAX    D       ;GET THE ACTUAL DATA TO ACCUM
 4469  012145  274                     CMP     H       ;SEE IF SAME
 4470  012146  304  274  024   V.ERR:  CNZ     VERRPT  ; AND B3-B0 SHOULD BE .EQ. 0.. IF NOT REPORT AS ERROR
 4471  012151  033                     DCX     D       ;FIX D TO LOOK AT THE BEGINNG OF THE 'ACTUAL' 2 BYTES
 4472
 4473                          ;NOW NEED TO CHOOSE IF NEED CHECK TWICE(FOR DOUBLE COPIES), OR IF END
 4474                          ;OF LIST FOR THIS CRAM WORD
 4475  012152  003                     INX     B       ;UPDATE POINTER INTO THE INDEXER LIST
 4476  012153  012                     LDAX    B       ;GET THE NEXT INDEX BYTE
 4477  012154  027                     RAL             ;COPY SIGN INTO C-BIT
 4478  012155  332  053  024          JC      V.DONWD ;IF SET, END OF LIST.. GO DO NEXT CRAM WORD
 4479
 4480  012160  027                     RAL             ;WASN'T END OF LIST.. SEE IF A DOUBLE CHECKER
 4481  012161  332  115  024          JC      V.BLP   ;JUMP BACK TO MAIN LOOP WITHOUT UPDATING IF YES
 4482
 4483                          ;IF NOT A DOUBLE, MUST FALL THROUGH TO UPDATE EXPECTED
```

```
 4485                          ;HERE WHEN MUST UPDATE EXPECTED,.NOT A DOUBLE COPY
 4486  012164  041  253  040   V.NXT:  LXI     H,VERCNT;LOAD CURRENT COUNT FOR HOW MANY COMPARE PER MEM WORD
 4487  012167  065                 DCR     M       ;DECREMENT THAT COUNT
 4488  012170  312  105  024          JZ      V.NXMEM ;IF DOWN TO ZERO, GO READ IN THE NEXT MEMORY WORD
 4489
 4490  012173  041  012  040          LXI     H,EMBUF ;TELL SHR36 WHERE IT SHOULD SHIFT
 4491  012176  315  154  031          CALL    SHR36   ;IF NOT DOWN, SHIFT WHAT WE HAVE TO THE NEXT 12 BIT GRP
 4492  012201  014                    .BYTE   12,     ;SPECIFY THE NEXT 12 BIT GROUP IS WHAT WE WANT
 4493  012202  303  115  024          JMP     V.BLP   ;AND CONTINUE IN THE BIG LOOP
 4494
 4495                          ;LIST OF INDEXER BYTES
 4496                          .RADIX  8
 4497  012205  000  017       VERLST: .BYTE   0,17    ;READ FCN 17 (BITS 84-95)
 4498  012207  002  016               .BYTE   2,16    ;READ FCN 16 (BITS 72-83)
 4499  012211  002  015               .BYTE   2,15    ;READ FCN 15 (BITS 60-71)
 4500  012213  002  014               .BYTE   2,14    ;READ FCN 14 (BITS 48-59)
 4501  012215  002  013               .BYTE   2,13    ;READ FCN 13 (BITS 36-47)
 4502  012217  102  012               .BYTE   102,12  ;READ FCN 12 (BITS 36-47) SECOND COPY
 4503  012221  010  006               .BYTE   10,6    ;READ FCN 6  (BITS 24-35)
 4504  012223  102  005               .BYTE   102,5   ;READ FCN 5  (BITS 24-35) SECOND COPY
 4505  012225  002  004               .BYTE   2,4     ;READ FCN 4  (BITS 12-23)
 4506  012227  010  000               .BYTE   10,0    ;READ FCN 0  (BITS 00-11)
 4507  012231  200                    .BYTE   200     ;END OF LIST MARKER
 4508                          .RADIX  10
```

```
 4510                          ;SUBROUTINE TO READ IN THE NEXT WORD FROM MEMORY.
 4511                          ;ALSO CHECKS TO SEE IF AT THE END OF THE MEMORY PAGE(ADDR 1777), AND IF SO
 4512                          ;TO GO AND READ IN THE NEXT PAGE OF MICRO-CODE FROM THE DEVICE AGAINST
 4513                          ;WHICH WE ARE VERIFYING THE MICRO-CODE.
 4514  012232  325             GATHER: PUSH    D
 4515  012233  305                     PUSH    B
 4516
 4517  012234  052  017  040          LHLD    MEMAD   ;GET CURRENT MEMORY ADDRESS
 4518  012237  345                     PUSH    H       ;SAVE CURRENT MEM ADDRESS
 4519
 4520  012240  174                     MOV     A,H     ;GET THE HI ORDER PIECE OF THE MEM ADDRESS
 4521  012241  346  004               ANI     ^O4     ;SEE IF ADDRESS AT '2000' YET
 4522  012243  312  261  024          JZ      G.SKP   ;IF NOT, SIMPLY GO READ IN THE NEXT WORD
 4523
 4524                          ;ELSE MUST READ IN THE NEXT PAGE OF MICRO-CODE
 4525  012246  315  244  023          CALL    NEXTCR  ;DO THE READ IN
 4526
 4527  012251  341                     POP     H       ;GET OLD CRUMMY H,L OFF THE STACK
 4528  012252  041  000  002          LXI     H,^O1000 ;WANT TO RESET MEM ADDRESS TO BEGINNING OF PAGE
 4529  012255  345                     PUSH    H       ;PUT BACK ON STACK
 4530  012256  042  017  040          SHLD    MEMAD   ;AND PASS NEW ADDRESS IN RAM
 4531  012261  315  036  012  G.SKP:  CALL    EM2     ;EXAMINE THE NEXT MEMORY WORD
 4532  012264  341                     POP     H       ;GRAB THE ADDRESS WE WANT TO READ NEXT TIME
 4533  012265  043                     INX     H       ;UPDATE TO NEXT
 4534  012266  042  017  040          SHLD    MEMAD   ;PUT IT BACK
 4535  012271  301                     POP     B       ;RESTORE THE REGS NOW
 4536  012272  321                     POP     D
 4537  012273  311                     RET             ;AND OUT
```

```
 4539                          ;VERIFY ERROR REPORTER SUBROUTINE, REPORTS VERIFY ERRORS AS THEY HAPPEN
 4540                          ;AND THEN PERMITS THE VERIFIER TO CONTINUE VERIFYING THE REST OF THE CRAM
 4541  012274  345             VERRPT: PUSH    H       ;SAVE CONTENTS OF H,L PAIR
 4542  012275  325                     PUSH    D       ;MUST ALSO SAVE D,E
 4543  012276                         INTOFF          ;PRINT ALL THIS GOOD STUFF
  (1)  012276  367                     RST     6       ;GO EXIT FROM INTERNAL MODE
 4544
 4545  012277  041  004  040          LXI     H,CRMAD ;THEN PRINT THE CRAM ADDRESS OF THE FAILING CRAM WORD
 4546  012302  315  317  030          CALL    P16     ;AND PRINT OUT THE ADDRESS
 4547
 4548  012305                         PSLASH          ;THROW OUT A "/"
  (2)  012305  317                     RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (2)  012306  057                     .BYTE   SLASH   ;CHAR TO PRINT
 4549
 4550  012307  012                     LDAX    B       ;FETCH UP THE DIAG FUNC OF THE READ FAILURE
 4551  012310  315  304  030          CALL    P8BITA  ;AND PRINT IT
 4552
 4553  012313                         PCHAR   ':      ;SIMPLE CHARACTERS ARE ": A "
  (1)  012313  317                     RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (1)  012314  072                     .BYTE   ':      ;CHAR TO PRINT
 4554  012315                         PSPACE
  (2)  012315  317                     RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (2)  012316  040                     .BYTE   SPACE   ;CHAR TO PRINT
 4555  012317                         PCHAR   'A
  (1)  012317  317                     RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (1)  012320  101                     .BYTE   'A      ;CHAR TO PRINT
 4556  012321                         PSPACE
  (2)  012321  317                     RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (2)  012322  040                     .BYTE   SPACE   ;CHAR TO PRINT
 4557
 4558  012323  353                     XCHG            ; AND NOW H,L POINTS AT THE ACTUAL
 4559  012324  053                     DCX     H       ;NOW D,E POINTS TO THE ACTUAL
 4560  012325  315  317  030          CALL    P16     ;PRINT THE ACTUAL DATA
 4561  012330  353                     XCHG            ;FIX SO H,L POINTS AT TEMP LOC ONCE AGAIN
```

```
 4563  012331                        PSPACE          ;ANOTHER SPACE
  (2)  012331  317                   RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (2)  012332  040                   .BYTE   SPACE   ;CHAR TO PRINT
 4564  012333                        PCHAR   'E      ;A "W" STANDS FOR "WAS"
  (1)  012333  317                   RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (1)  012334  105                   .BYTE   'E      ;CHAR TO PRINT
 4565  012335                        PSPACE
  (2)  012335  317                   RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (2)  012336  040                   .BYTE   SPACE   ;CHAR TO PRINT
 4566
 4567  012337  052  012  040         LHLD    EMBUF   ;GET THE EXPECTED DATA INTO H,L PAIR
 4568  012342  174                   MOV     A,H     ;WANT TO STRIP ANY BITS ABOVE 12 BITS
 4569  012343  346  017              ANI     ^O17    ;KEEP ONLY RELEVANT BITS
 4570  012345  147                   MOV     H,A     ;PUT IT BACK
 4571  012346  042  100  040         SHLD    TMPB2   ;PUT IT IN THE TEMP PLACE
 4572  012351  315  314  030         CALL    P16.    ;PRINT THE EXPECTED
 4573  012354                        PCRLF
  (2)  012354  347                   RST     4
  (2)  012355  002                   .BYTE   2
 4574
 4575  012356  305                   PUSH    B       ;SAVE B REG TOO
 4576  012357  315  205  034         CALL    DECNET  ;IF THERE IS A HOST, TELL HIM TOO
 4577  012362  301                   POP     B
 4578  012363                        INTON           ;AND BACK TO INTERNAL MODE
  (1)  012363  327                   RST     2       ;GO SET INTERNAL MODE
 4579  012364  321                   POP     D
 4580  012365  341                   POP     H
 4581  012366  311                   RET
```

```
 4583                          ;EXECUTE CHANNEL COMMANDS....
 4584  012367  041  052  036   DSXFR:  LXI     H,DSKSEQ ;PNTR TO COMMAND LIST
 4585  012372  315  023  025   XCTNOW: CALL    CHNXCT  ;EXECUTE CHANNEL LIST
 4586  012375  311                 RET
 4587
 4588  012376  041  215  036   MTXFR:  LXI     H,MTASEQ ;PNTR TO COMMAND LIST
 4589  012401  062  144  040   XCTMTA: STA     SKP,GO  ;ACCUM HAD XFR TYPE..SAVE IT
 4590  012404  303  372  024           JMP     XCTNOW  ;GO EXECUTE THE CHANNEL COMMAND LIST
 4591
 4592  012407  041  275  036   QMXFR:  LXI     H,QTXFR ;PNTR TO COMMAND LIST
 4593  012412  303  001  025           JMP     XCTMTA  ;GO EXECUTE THE CHANNEL COMMAND LIST
 4594
 4595  012415  041  363  036   MTRESET: LXI    H,MTARST ;CHANNEL COMMAND LIST TO CLEAR ERROR FROM MAGTAPE
 4596  012420  303  372  024           JMP     XCTNOW  ;GO EXECUTE THE CHANNEL COMMAND LIST
```

```
                        .SBTTL  CHANNEL COMMAND LIST EXECUTOR
                        ;ROUTINE TO A CHANNEL COMMAND LIST TYPE OPERATION, FOR DATA
                        ;TRANSFERS FROM OUR SELECTED BOOT DEVICE..
                        ;COMMAND LIST IS CODED AS FOLLOWS
                        ; THE LIST IS A SERIES OF 36-BITCOMMANDS.
                        ;WE HAVE FREE USE OF BITS 0-17 AS COMMAND TYPES
                        ; BITS 15,16,17 .EQ. 0 MEAN 'DI' COMMAND
                        ; BITS 15,16,17 .EQ. 1 MEAN 'LI' COMMAND
                        ; BITS 15,16,17 .EQ. 2 MEAN 'EI' COMMAND
                        ; BITS 15,16,17 .EQ. 3 MEAN 'WAIT' COMMAND
                        ; BITS 15,16,17 .EQ. 4 MEAN 'ERRTST' COMMAND
                        ; BITS 15,16,17 .EQ. 5 MEAN 'END' OF COMMAND LIST
                        ; BITS 15,16,17 .EQ. 6 MEAN 'TWAIT' COMMAND
                        ; BITS 15,16,17 .EQ. 7 MEAN 'UBA' COMMAND

                        ;POINTER TO THE CURRENT COMMAND LIST IS ALWAYS STORED IN H,L
012423                  CHNXCT: INTON           ;SET UP FOR INTERNAL MODE
 (1) 012423 327                 RST     2       ;GO SET INTERNAL MODE
012424 305                      PUSH    B
012425 325                      PUSH    D
012426 345                      PUSH    H

012427 021 002 000      DSCON:  LXI     D,2     ;'D,E' GETS THE CONSTANT '2'
012432 031                      DAD     D       ;NOW 'H,L' POINTS TO 'DATA+2'(BITS 12-19)

012433 104                      MOV     B,H     ;COPY 'H,L' INTO 'B,C'
012434 115                      MOV     C,L
012435 345                      PUSH    H       ;SAVE 'H,L'
012436 176                      MOV     A,M     ;GET BITS 12-19 INTO ACCUM

012437 037                      RAR             ;NOW JUSTIFY ACCUM AT BITS 16,17
012440 037                      RAR             ;TAKES 2 SHIFTS
012441 346 017                  ANI     ^O17    ;OFF ALL BUT BITS 14,15,16,17
012443 137                      MOV     E,A     ;NOW PUT INTO LO-ORDER HALF OF DOUBLE REG
012444 041 066 025              LXI     H,DSLST ;GET A PNTR TO THE DISPATCHING LIST
012447 031                      DAD     D       ;CREAT PNTR TO THE COMMAND

012450 136                      MOV     E,M     ;GET LO ORDER PIECE OF CMD DISPATCH
012451 043                      INX     H
012452 126                      MOV     D,M     ;GET HI ORDER PIECE OF CMD DISPATCH
012453 353                      XCHG            ;ASSEMBLED ADDRESS TO 'H,L'
012454 021 061 025              LXI     D,XFRRT ;NOW GET A PSEUDO RETURN PC TO PUT ON STACK
012457 325                      PUSH    D       ;AND PUT IT THERE

012460 351                      PCHL            ;DISPATCH TO THAT ASSEMBLED ADDRESS
```

```
 4644                           ;UPON COMPLETION OF THE COMMAND LIST COMMANDS, YOU GENERALLY
 4645                           ;RETURN HERE IN THE CODE
 4646  012461  341              XFRRT:  POP     H       ;GET THE POINTER TO CURRENT LOCATION IN COMMAND LIST
 4647  012462  043                      INX     H       ;MAKE IT POINT TO NEXT WORD IN THE LIST
 4648  012463  303  027  025            JMP     DSCON   ;AND CONTINUE IN COMMAND LIST EXECUTOR
 4649
 4650                           ;COMMAND LIST DISPATCH SELECTION
 4651  012466  164  025        DSLST:  .ADDR   CMDDI   ;DI CMD .EQ. 0
 4652  012470  112  025                .ADDR   CMDLI   ;LI CMD .EQ. 2
 4653  012472  106  025                .ADDR   CMDEI   ;EI CMD .EQ. 4
 4654  012474  213  025                .ADDR   CMDWAIT ;WAIT CMD .EQ. 6
 4655  012476  262  025                .ADDR   CMDERCHK ;ERRTST CMD .EQ. 10
 4656  012500  302  025                .ADDR   CMDEN   ;END CMD .EQ. 12
 4657  012502  244  025                .ADDR   CMDTWAIT ;WAIT CMD WITH NO TIMEOUT, CHECKS FOR BIT TRUE .EQ. 14
 4658  012504  146  025                .ADDR   CMDUBA  ;LI TYPE CMD, NO OFFSETS, GOOD FOR UBA STUFF .EQ. 16
```

```
 4660                          ;CODE FOR 'EI' COMMAND
 4661  012506  247            CMDEI:  ANA     A       ;CLR THE 'C-BIT'
 4662  012507  303  113  025          JMP     CMDLI1  ;GO TO COMMON CODE FOR LI AND EI COMMAND
 4663
 4664                          ;CODE FOR 'LI' COMMAND
 4665  012512  067            CMDLI:  STC             ;SET THE 'C-BIT'
 4666  012513  365            CMDLI1: PUSH    PSW     ;AND SAVE IT
 4667  012514                          MOV5B           ;PASS THE COMMAND LIST EXECUTOR THE RHBASE ADDRESS
  (1)  012514  347                    RST     4
  (1)  012515  000                    .BYTE   0
 4668  012516  242  040               .ADDR   RHBASE  ;KEPT IN HERE
 4669  012520  024  040               .ADDR   IOAD    ;USED IN HERE
 4670
 4671  012522  041  026  040          LXI     H,IOAD+2 ;'H,L' PNTS TO DEST+2
 4672  012525  072  254  040          LDA     UBANUM  ;CURRENT UBA NUMBER INTO ACCUM
 4673  012530  266                    ORA     M       ;THROW IN THE CURRENT BITS
 4674  012531  167                    MOV     M,A     ;PUT IT ALL BACK
 4675
 4676  012532  053                    DCX     H       ;NOW MAKE 'H,L' POINT TO ADDR +0
 4677  012533  053                    DCX     H
 4678  012534  013                    DCX     B       ;MAKE 'B,C' PAIR POINT TO SELECTED OFFSET FROM BASE
 4679  012535  012                    LDAX    B       ;GET SELECTED OFFSET
 4680  012536  206                    ADD     M       ;ADD OFFSET TO THE BASE
 4681  012537  167                    MOV     M,A     ;AND PUT THE WHOLE MESS BACK
 4682  012540  361                    POP     PSW     ;NOW GET STATE OF PROCESSOR FLAGS
 4683  012541  330                    RC              ;IF 'C' SET,IT WAS AN LI AND WE ARE DONE
 4684
 4685                          ;CALL THRU IF 'C' CLR,,IT WAS EI AND WE MUST FINISH IT
 4686  012542  315  357  012          CALL    EI1     ;EXECUTE 'EI' CMD
 4687  012545  311                    RET             ;ALL DONE
```

```
 4689                           ;CODE FOR LI TYPE COMMAND ONLY USING NO OFFSETS, TAKING THE ADDRESSES
 4690                           ;EXACTLY AS PRESENTED.. GOOD FOR UBA OPERATIONS, WHICH REQUIRE NO OFFSETS
 4691  012546  021  026  040   CMDUBA: LXI     D,IOAD+2 ;'D,E' PNTS TO DEST+2
 4692  012551  325                   PUSH    D       ;SAVE THE ADDRESS OF UBA/RH ADDRESS
 4693  012552  315  323  025          CALL    MOV18B  ;MOVE SOME DATA
 4694  012555  341                   POP     H       ;ADDR OF UBA/RH INTO H,L
 4695  012556  072  254  040          LDA     UBANUM  ;CURRENT UBA NUMBER INTO ACCUM
 4696  012561  266                   ORA     M       ;THROW IN THE CURRENT BITS
 4697  012562  167                   MOV     M,A     ;PUT IT ALL BACK
 4698  012563  311                   RET             ;WE ARE DONE
 4699
 4700                           ;CODE FOR DI COMMAND
 4701  012564  012             CMDDI:  LDAX    B       ;GET DISP CODE TO SEE IF INDIRECT
 4702  012565  247                   ANA     A       ;CHECK THE SIGN BIT
 4703  012566  362  201  025          JP      DILOCL  ;AND JUMP IF NO INDIRECTION
 4704
 4705                           ;FALL TO HERE IF WAS INDIRECT
 4706  012571  151                   MOV     L,C     ;PASS ADDR IN 'B,C' TO 'H,L'
 4707  012572  140                   MOV     H,B
 4708  012573  053                   DCX     H       ;NOW BACK UP PNTR TO HI ORDER PCE OF INDIRECT WRD
 4709  012574  106                   MOV     B,M     ;AND INTO B
 4710  012575  053                   DCX     H       ;NOW TO LO ORDER PIECE OF INDIRECT WORD
 4711  012576  116                   MOV     C,M     ;LO ORDER PIECE TO C AND DONE
 4712  012577  003                   INX     B       ;'B,C' MUST POINT TO SRC + 2
 4713  012600  003                   INX     B
 4714  012601  021  051  040   DILOCL: LXI     D,DMDAT+2 ;'D,E' POINTS TO DEST+2
 4715  012604  315  323  025          CALL    MOV18B  ;MOVE SOME STUFF AROUND
 4716  012607  315  016  013          CALL    DI1     ;EXECUTE THE DEPOSIT
 4717  012612  311                   RET             ;AND BACK TO COMMAND LIST
 4718
```

```
 4720                           ;CODE FOR WAIT COMMAND(FOR WAITING FOR A TAPE TO FINISH, FOR EXAMPLE)
 4721  012613                   CMDWAIT:
 4722  012613  257                     XRA     A       ;CLEAR ACCUM
 4723  012614  127                     MOV     D,A     ;NOW CLR 'D', THE REGISTER WE WILL USE FOR TIMEOUT COUNT
 4724  012615  137                     MOV     E,A     ;CLR 'E' TOO
 4725
 4726  012616  305             WAITLP: PUSH    B       ;SAVE B,C
 4727  012617  315  344  025          CALL    CHKBIT  ;CHECK BITS VERSUS DEVICE STATUS TO SEE IF SET(I.E READY)
 4728  012622  301                     POP     B       ;RESTORE B,C
 4729  012623  300                     RNZ             ;IF READY BIT SET, ITS OK, GO AWAY
 4730
 4731                           ;FALL THRU IF READY NOT SET
 4732  012624  305                     PUSH    B       ;SAVE B,C FROM DESTRUCTION
 4733  012625  325                     PUSH    D       ;SAVE TIME-OUT COUNT
 4734  012626  315  357  012          CALL    EI1     ;DO ANOTHER EXAMINE OF DEVICE STATUS TO SEE IF READY NOW
 4735  012631  321                     POP     D       ;GET THE TIME-OUT COUNT
 4736  012632  301                     POP     B       ;RESTORE B,C
 4737  012633  023                     INX     D       ;INCREMENT
 4738  012634  173                     MOV     A,E     ;SEE IF COUNT DOWN TO ZERO YET
 4739  012635  262                     ORA     D       ;USE TOP HALF TOO
 4740  012636  302  216  025          JNZ     WAITLP  ;GO TRY AGAIN
 4741
 4742                           ;FALL THRU IF DEVICE TIME'S OUT BEFORE GETTING A READY
 4743  012641  303  270  025          JMP     DEVERR  ;GO REPORT '?BT' AND THE FAILED PC
 4744
 4745                           ;CODE FOR TWAIT COMMAND
 4746  012644  305             CMDTWAIT: PUSH  B       ;FIRST SAVE THE BC POINTERS
 4747  012645  315  357  012          CALL    EI1     ;READ THE CURRENT STATE AS IT IS NOW
 4748  012650  301                     POP     B       ;RESET B TO A GOOD VALUE
 4749  012651  305                     PUSH    B       ;AND SAVE IT FROM DESTRUCTION AGAIN
 4750  012652  315  344  025          CALL    CHKBIT  ;NOW SEE IF APPROPRIATE BIT IS SET
 4751  012655  301                     POP     B       ;RESTORE
 4752  012656  300                     RNZ             ;RETURN IF BIT WAS SET AS DESIRED
 4753  012657  303  270  025          JMP     DEVERR  ;ELSE.. GO SAY ERR, BIT WAS NOT SET AS DESIRED
 4754
 4755                           ;CODE FOR ERROR TEST COMMAND
 4756  012662  305             CMDERCHK: PUSH  B       ;SAVE B,C
 4757  012663  315  344  025          CALL    CHKBIT  ;FIRST GO CHECK TO SEE IF ANY OF DESIRED BITS SET
 4758  012666  301                     POP     B       ;AND RESTORE
 4759  012667  310                     RZ              ;IF NONE SET, RETURN CAUSE ALL IS OK
 4760
 4761                           ;FALL TO HERE IF SOME BITS SET..HAD DEVICE ERROR & THEREFORE "BOOT" FAILED
 4762  012670  140             DEVERR: MOV     H,B     ;NOW COPY 'B,C' TO 'H,L' REGISTER
 4763  012671  151                     MOV     L,C
 4764
 4765  012672  042  154  040          SHLD    ERRCD   ;'H,L' NOW HAS FAILING PC
 4766  012675                         INTOFF          ;LET ALL MESSAGES PRINT NOW
  (1)  012675  367                     RST     6       ;GO EXIT FROM INTERNAL MODE
 4767  012676  257                     XRA     A       ;CLR C-BIT, SO THAT IT WILL SET LATER, TO INDICATE ERR
 4768  012677  303  312  025          JMP     DEVEXIT ;GO EXIT WITH MESSING WITH PRINT FLAGS
```

```
 4770                           ;CODE FOR THE END COMMAND
 4771  012702  021  364  016    CMDEN:  LXI     D,MAD000 ;GUARANTEE CSL BUS ADDR REG .EQ. 0 AFTER XFR
 4772  012705  315  176  033            CALL    ADATP   ;WRITE THE CSL BUS ADDRESS REG WITH 0'S
 4773  012710                         INTOFF          ;CLR INTERNAL MODE
  (1)  012710  367                    RST     6       ;GO EXIT FROM INTERNAL MODE
 4774  012711  067                    STC             ;SET THE C-BIT, SO THAT IT WILL BE CLEARED LATER
 4775  012712  077                DEVEXIT: CMC         ;COMPLEMENT C-BIT, SO IT WILL SAY ERR, OR NO ERR
 4776  012713  341                    POP     H       ;THROW OUT PSEUDO RETURN FROM STACK TOP
 4777  012714  341                    POP     H       ;THROW OUT THE SAVED 'H,L'
 4778  012715  341                    POP     H       ;AND RESTORE ALL THE REGISTERS
 4779  012716  321                    POP     D       ;THAT WE SAVED
 4780  012717  301                    POP     B
 4781  012720  076  000              MVI     A,0     ;ACCUM WILL ALWAYS BE 0 ON EXIT FROM CMD LIST EXECUTOR
 4782  012722  311                    RET
 4783                           ;ROUTINE FOR MOVING 3 BYTES OF DATA
 4784                           ;'B,C' HAS SRC+2...'D,E' HAS DST+2
 4785  012723  046  002        MOV18B: MVI     H,2     ;COUNT OF 3 IS HOW MANY TO MOVE
 4786  012725  012                    LDAX    B       ;GET PIECE OF SOURCE
 4787  012726  346  003              ANI     3       ;ONLY INTERESTED IN BITS 18,19
 4788  012730  022                    STAX    D       ;PUT AT DESTINATION
 4789  012731  013                    DCX     B       ;POINT TO NEXT RELEVANT BYTE
 4790  012732  033                    DCX     D       ;FOR SRC AND DST
 4791  012733  012            MOV18X: LDAX    B       ;GET A PIECE OF THE SRC TO ACCUM
 4792  012734  022                    STAX    D       ;AND PUT AT DESTINATION PLACE
 4793  012735  013                    DCX     B       ;DOWN THE POINTER
 4794  012736  033                    DCX     D       ;DOWN THE OTHER POINTER
 4795  012737  045                    DCR     H       ;AND DOWN THE COUNTER..
 4796  012740  302  333  025          JNZ     MOV18X  ;CONTINUE TILL MOVED 3 BYTES
 4797  012743  311                    RET             ;THEN OUT
 4798
 4799                           ;COMMON ROUTINE FOR CHECKING DEVICE STATUS, FOR EITHER DEVICE ERRORS OR
 4800                           ;READY BIT TRUE..MUST BE CALLED IMMEDIATLY AFTER AN 'EI,' COMMAND
 4801  012744  052  012  040    CHKBIT: LHLD    EMBUF   ;GET CURRENT DEVICE STATUS INTO 'H,L'
 4802  012747  013                    DCX     B       ;MAKE B PNT TO +1
 4803  012750  013                    DCX     B       ; AND MAKE IT PNT TO +0
 4804  012751  012                    LDAX    B       ;BYTE OF DESIRED INTO ACCUM
 4805  012752  245                    ANA     L       ;COMPARE AGAINST CURRENT DEVICE
 4806  012753  300                    RNZ             ;IF NON-ZERO, NO NEED TO LOOK FURTHER
 4807
 4808  012754  003                    INX     B       ;IF 1ST BYTE WAS ZERO, BUMP PNTR TO LOOK AT NEXT BYTE
 4809  012755  012                    LDAX    B       ;NEXT BYTE INTO THE ACCUM
 4810  012756  244                    ANA     H       ;COMPARE VERSUS DESIRED
 4811  012757  311                    RET             ;PLAIN RETURN..Z BIT WILL BE SET APPROPRIATELY
```

```
 4813                           .SBTTL  *** 'BC' CMD ***
 4814                           ;CODE WHICH PERFORMS THE VERY FAMOUS BOOT CHECK I
 4815  012760  315  225  005    BCCMD:  CALL    MRCMD   ;MAKE SURE MACHINE IS STOPPED
 4816  012763                           INTON           ;SET UP INTERNAL MODE
 (1)   012763  327                      RST     2       ;GO SET INTERNAL MODE
 4817  012764                           CLRB    ERRCD   ;BEGIN BY CLEARING THE ERROR CODES
 4818  012766                           CLRB    ERRCD+1 ;MUST CLEAR BOTH HALVES
 4819        001                .IF     NDF,NEEDRM
 4820
 4821  012770                           CLRRM   BUSAD   ;WILL GENERATE A STARTING BUFFER OF (400000,,0)
 (1)   012770  347                      RST     4
 (1)   012771  012                      .BYTE   10.
 (1)   012772  047  040                 .ADDR   BUSAD+5
 4822  012774  076  010                MVI     A,^O10  ;THIS TURNS OUT TO BE BIT0 IN 36-BIT LAND
 4823  012776  062  046  040           STA     BUSAD+4 ;  SET IT
 4824
 4825  013001  001  000  001           LXI     B,^O400 ;SET B=1, C=0
 4826  013004  305                BC.ALP: PUSH    B       ;SAVE COUNTERS
 4827  013005  315  240  011           CALL    DBCMD   ;EXECUTE THE DEPOSIT BUS
 4828
 4829                           ;CHECK FOR FAILURE
 4830  013010  072  154  040           LDA     ERRCD   ;FETCH AN ERROR CODE
 4831  013013  247                      ANA     A       ;SET 8080 CONDITION CODES
 4832  013014  302  371  026           JNZ     BCA.ERR ;GO STANDARD ERROR REPORT IF ERR FOUND
 4833
 4834                           ;NO ERROR, GENERATE THE NEXT DATUM
 4835  013017  041  042  040           LXI     H,BUSAD ;POINT TO A BUFFER TO BE SHIFTED
 4836  013022  315  154  031           CALL    SHR36   ;SHIFT 36 BITS(I.E. FLOAT A 1 OR 0)
 4837  013025  001                      .BYTE   1       ;SHIFT ONLY ONE PLACE AT A TIME
 4838
 4839                           ;NOW CHECK FOR END OF TEST
 4840  013026  301                      POP     B       ;GRAB UP THE CURRENT LOOP COUNTERS
 4841  013027  014                      INR     C       ;UP COUNT FOR THIS DATAUM
 4842  013030  171                      MOV     A,C     ;COPY TO ACCUM
 4843  013031  376  044                CPI     36.     ;NOW SEE IF FLOATED DOWN THE ENTIRE 36 BIT WORD
 4844  013033  332  004  026           JC      BC.ALP  ;JUMP IF NOT DONE A GROUP OF 36 YET..
 4845
 4846                           ;HERE WHEN DONE A GROUP OF 36, SEE IF THIS WAS FIRST OR SECOND TIME THROUGH
 4847  013036  005                      DCR     B       ;DECREMENT 'TIMES THROUGH THE LOOP' COUNTER
 4848  013037  372  062  026           JM      BC.2ND  ;IF MINUS, TIME TO GET OUT..GO NEXT PHASE OF TEST
 4849
 4850                           ;HERE WHEN DONE FIRST WORD, TIME TO SET UP TO FLOAT A 0 THROUGH A FIELD
 4851                           ;OF ONES
 4852  013042                           MOV5B           ;MOVE 2ND DATA PATTERN
 (1)   013042  347                      RST     4
 (1)   013043  000                      .BYTE   0
 4853  013044  055  026                .ADDR   BC.DB2  ;INIT 2ND PATTERN TO BE (377777,,777777)
 4854  013046  042  040                .ADDR   BUSAD   ;AND THIS IS THE PLACE THAT THE DB COMMAND USES
 4855
 4856  013050  016  000                MVI     C,00    ;RESET THE COUNTER
 4857  013052  303  004  026           JMP     BC.ALP  ;AND GO ROUND FOR THE SECOND TIME.....
 4858  013055                     BC.DB2: D       377,777,,777,777
```

```
 4860                          ;BOOTCHECK CODE FOR EXECUTING A TEST OF THE CRAM AND ITS ABILITIY TO HOLD
 4861                          ;ALL ONES AND ZEROES, AND TO SEE IF ITS ADDRESSING LOGIC WORKS AS
 4862                          ;IT SHOULD....
 4863  013062  041  000  000   BC.2ND: LXI     H,00    ;START AT CRAM ADDRESS 00
 4864  013065  315  235  026   BC.BLP: CALL    W.CRMZ  ;WRITE THE LOCATION WITH ALL ZEROES
 4865  013070  043                  INX     H       ;UPDATE TO NEXT CRAM ADDRESS
 4866  013071  174                  MOV     A,H     ;NOW CHECK TO SEE IF DONE ALL
 4867  013072  346  010             ANI     ^O10    ;IS ADDRESS AT "4000" YET??
 4868  013074  312  065  026        JZ      BC.BLP  ;BACK AND ZERO MORE IF NOT YET
 4869
 4870                          ;NOW READY FOR COMBINED ADDRESS AND DATA TEST
 4871  013077  046  000             MVI     H,00    ;L IS ALREADY .EQ. 0, NOW MAKE H,L PAIR .EQ. 0
 4872  013101  315  320  014   BC.BL1: CALL    CADWR   ;WRITE CURRENT CRAM ADDRESS
 4873  013104  345                  PUSH    H       ;AND SAVE IT FOR A WHILE
 4874  013105  315  204  013        CALL    CP1     ;CLOCK TO GET THE CONTENTS OF THAT LOC TO C.R.
 4875  013110  315  021  015        CALL    RCINT   ;NOW READ-IN THE CONTENTS OF THE C.R.
 4876  013113  041  000  000        LXI     H,00    ;DATA TO VERIFY AGAINST IS 00
 4877  013116  315  263  026        CALL    V.VER   ;VERIFY CONTENTS OF C.R. TO BE ALL 0
 4878  013121  315  226  026        CALL    A.CRMO  ;NOW WRITE THAT LOCATION WITH ALL ONES
 4879  013124  341                  POP     H       ;RETRIEVE CURRENT CRAM ADDRESS
 4880  013125  043                  INX     H       ;UP TO THE NEXT ADDRESS
 4881  013126  174                  MOV     A,H     ;COPY HI HALF TO ACCUM, SO CAN CHECK FOR 4000
 4882  013127  346  010             ANI     ^O10    ;ADDRESS AT '4000'??
 4883  013131  312  101  026        JZ      BC.BL1  ;BACK INTO LOOP IF NOT YET..
 4884
 4885                          ;WHEN DONE HERE, FALL INTO MEMORY PART OF BOOT CHECK
 4886                          ;A PAGE MOS MEMORY CHECK
 4887  013134  315  064  020   BC.3RD: CALL    ZMCMD   ;FIRST CLEAR THE ENTIRE MOS MEMORY(AT LEAST TRY)
 4888
 4889  013137                        MOV5B           ;SET UP THE INITIAL DATA
  (1)  013137  347                   RST     4
  (1)  013140  000                   .BYTE   0
 4890  013141  256  022             .ADDR   ONES    ;DATA FOR DEPOSITING IS ALL ONES
 4891  013143  047  040             .ADDR   DMDAT   ;.. THE DEPOSIT BUFFER
 4892
 4893  013145                        MOV5B           ;SET UP THE STARTING MEMORY ADDRESS
  (1)  013145  347                   RST     4
  (1)  013146  000                   .BYTE   0
 4894  013147  244  022             .ADDR   MA1000  ;START AT ADDRESS 1000
 4895  013151  017  040             .ADDR   MEMAD   ;.. MEMORY ADDRESS BUFFER
 4896
 4897  013153  315  036  012   BC.CLP: CALL    EM2     ;EXAMINE A LOCATION
 4898  013156  315  022  033        CALL    CMP36   ;THEM COMPARE RESULTS..DATA SHOULD BE ALL ZEROES
 4899  013161  012  040             .ADDR   EMBUF   ;THIS IS THE ACTUAL READ-IN DATA
 4900  013163  364  016             .ADDR   ZEROES  ;VERSUS 36-BITS OF 0'S
 4901  013165  302  342  026        JNZ     BC.CERR ;GO TO ERROR REPORT IF NOT ALL ZEROES
 4902
 4903                          ;FALL THRU IF THAT WENT OK.
 4904  013170  315  234  012        CALL    DM2     ;NOW DEPOSIT ONES INTO THAT LOC AND CONTINUE
 4905
 4906  013173  315  036  012        CALL    EM2     ;WHILE WE ARE AT IT, WE'LL CHECK ALL ONES
 4907  013176  315  022  033        CALL    CMP36   ;DO THE 36-BIT COMPARE
 4908  013201  012  040             .ADDR   EMBUF   ;THIS STUFF JUST READ IN
 4909  013203  256  022             .ADDR   ONES    ;AGAINST ALL ONES
```

```
 4910  013205  302  342  026          JNZ     BC.CERR ;IF BAD, SAY SO..
 4911
 4912  013210  052  017  040          LHLD    MEMAD   ;FETCH UP THE CURRENT MEMORY ADDRESS
 4913  013213  043                    INX     H       ;UPDATE TO THE NEXT
 4914  013214  174                    MOV     A,H     ;COPY HI PIECE TO THE ACCUM
 4915  013215  346  004               ANI     ^O4     ;SEE IF REACHED ADDRESS 2000
 4916  013217  042  017  040          SHLD    MEMAD   ;REPLACE THE UPDATED ADDRESS FIRST
 4917  013222  312  153  026          JZ      BC.CLP  ;AND CONTINUE IF HAD NOT REACHED THE MAX
 4918
 4919  013225  311                    RET             ;RETURN..DONE ALL BOOT CHECK OK
```

```
 4921                           ;SUBROUTINES REQUIRED FOR THE CRAM TESTING
 4922                           ;SUBROUTINES FOR WRITING ALL ONES AND ALL ZEROES INTO A SELECTED CRAM
 4923                           ;LOCATION,, DESIRED ADDRESS PASSED IN THE H,L REGISTER(FOR W.XXX CALLS)
 4924                           ;USES CURRENTLY SELECTED ADDRESS FOR (A.XXX CALLS)
 4925  013226  345              A.CRMO: PUSH    H       ;SAVE H,L
 4926  013227  041  377  377           LXI     H,-1    ;H,L TO ALL ONES INDICATES THE ALL ONES DATA DESIRED
 4927  013232  303  244  026           JMP     W.LOC   ;GO COMMON CODE
 4928
 4929  013235  315  320  014  W.CRMZ: CALL    CADWR   ;WRITE DESIRED ADDRESS
 4930  013240  345              A.CRMZ: PUSH    H       ;SAVE H,L
 4931  013241  041  000  000           LXI     H,00    ;SET TO ZERO, DATA IS ALL ZEROES
 4932  013244  016  007        W.LOC:  MVI     C,7     ;TAKES 8 FUNCTION WRITES TO DO ALL OF ONE CRAM LOC
 4933  013246  171              W.LP:   MOV     A,C     ;GET CURRENT FUNCTION TO ACCUM
 4934  013247  062  010  040           STA     CRMFN   ;PUT FUNCTION INTO LOC USED BY STANDARD ROUTINE
 4935  013252  315  263  014           CALL    WFUNC   ;WRITE ONE OF THE 8 PIECES OD A CRAM LOC
 4936  013255  015                    DCR     C       ;DOWN COUNT
 4937  013256  362  246  026           JP      W.LP    ;AS LONG AS .GE. 0, KEEP GOING
 4938  013261  341                    POP     H       ;HERE WHEN DONE ALL 8
 4939  013262  311                    RET             ;NOW GET OUT OF HERE.
```

```
 4941                          ;ROUTINE TO VERFIY THAT A CRAM LOCATION IS INDEED ALL ONES OR ALL ZEROES.
 4942                          ;NO ARGUMENTS PASSED TO THIS ROUTINE
 4943  013263  042  316  040   V.VER: SHLD    CRMBF+^O14 ;ALL DONT CARE LOCS OF THE 'READ' ARE FUDGED
 4944  013266  042  320  040          SHLD    CRMBF+^O16 ;  TO MATCH EXPECTED DATA
 4945  013271  042  322  040          SHLD    CRMBF+^O20 ;  THERE ARE 12 DONT CARE LOCATIONS
 4946
 4947  013274  042  332  040          SHLD    CRMBF+^O30 ; SOME ARE VARIOUS ADDRESS(NXT/CURRENT/SUBRTN)
 4948  013277  042  334  040          SHLD    CRMBF+^O32 ; AND SOME ARE JUST UNBUFFERED COPIES OF THE
 4949  013302  042  336  040          SHLD    CRMBF+^O34 ; BUS
 4950
 4951  013305  175                    MOV     A,L     ;NOW COPY EXPECTED DATA INTO 'B,C' PAIR
 4952  013306  057                    CMA             ;FIRST COMPLIMENT
 4953  013307  117                    MOV     C,A     ;THEN MOVE
 4954  013310  174                    MOV     A,H     ; NEED BOTH HALVES PLEASE
 4955  013311  057                    CMA             ;COMPLIMENT
 4956  013312  107                    MOV     B,A     ;THEN MOVE
 4957
 4958  013313  041  302  040          LXI     H,CRMBF ;NOW POINT TO BEGINNING OF BUFFER WHERE EXPECTED DATA
 4959  013316  136              V.BCLP: MOV     E,M     ;IS KEPT.. PROCEED TO COPY BUFFER DATA INTO
 4960  013317  043                    INX     H       ;(UPDATE POINTER)
 4961  013320  126                    MOV     D,M     ;THE D,E REGISTER PAIR
 4962  013321  043                    INX     H       ;AND UPDATE MEMORY POINTER AFTER EACH MOVE
 4963  013322  353                    XCHG            ;SWAP, SO NOW 'H,L' HAS BUFFER, 'D,E' HAS POINTER
 4964  013323  011                    DAD     B       ;ADD COMPLIMENT TO EXPECTED...SHOULD GET 0.
 4965  013324  043                    INX     H       ;  TWO'S COMPLIMENT , THAT IS..
 4966  013325  175                    MOV     A,L     ;PIECE TO ACCUM
 4967  013326  264                    ORA     H       ;'OR' IN THE OTHER PIECE
 4968  013327  302  352  026          JNZ     BC.BERR ;AND GO HANDLE ERROR IF RESULTS .NE. 0
 4969
 4970                          ;YOU FELL THROUGH TO HERE IF DATA CHECK WAS OK..
 4971  013332  353                    XCHG            ;SWAP..NOW 'H,L' HAS POINTER, 'D,E' HAS 00
 4972  013333  076  342              MVI     A,<<CRMBF+32,>&^O377> ;CHECK IF DONE
 4973  013335  275                    CMP     L       ;SEE IF AT LAST LOCATION IN LIST
 4974  013336  302  316  026          JNZ     V.BCLP  ;JUMP BACK IF NOT YET
 4975  013341  311                    RET             ;ELSE OK TO RETURN
```

```
 4977                           ;ERROR REPORTING
 4978  013342  052  017  040  BC.CERR: LHLD   MEMAD   ;GRAB UP FAILED MEM ADDRESS
 4979  013345  006  100               MVI    B,^O100 ;GET A BIT TO SET IN ERROR PRINTOUT
 4980  013347  303  356  026          JMP    BCC.ERR ;GO PROCESS STANDARD ERROR TYPEOUTS
 4981  013352  341               BC.BERR:    POP    H       ;CLEAR A RETURN ADDRESS
 4982  013353  006  200               MVI    B,^O200 ;BIT TO SET FOR CRAM FAILURES
 4983  013355  341                    POP    H       ;NOW GATHER UP THE CURRENT CRAM ADDRESS
 4984  013356  042  154  040  BCC.ERR: SHLD   ERRCD   ;SAVE 1 BYTE BY DEPOSITING TWICE
 4985  013361  174                    MOV    A,H     ;HI HALF TO THE ACCUM
 4986  013362  260                    ORA    B       ;THROW IN A WEIGHT SO NUMBER WILL DIFFER
 4987  013363  062  155  040          STA    ERRCD+1 ;PUT NUMBER INTO ERROR CODE LOC
 4988  013366  303  376  026          JMP    BCB.ERR ;AND GO PRINT OUT THE CORRECT ERROR STUFF
```

```
 4990  013371  301            BCA.ERR:  POP   B       ;GET COUNTER OFF THE STACK
 4991  013372  171                      MOV   A,C     ;SUBTEST TO ACCUM
 4992  013373  062  154  040            STA   ERRCD   ;PLACE FROM WHICH TO REPORT ERRORS
 4993  013376                 BCB.ERR:  CLRB  NOPNT   ;GUARANTEE PRINTING ON
 4994  013400                           PLINE ERRMSG  ;'?BC'
  (1)  013400  337                      RST   3       ;PRINT LINE OF CHARS
  (1)  013401  201  037                 .ADDR ERRMSG  ;BUFF TO PRINT
 4995  013403  041  154  040            LXI   H,ERRCD ;POINT AT THE ERROR I.D.
 4996  013406  315  317  030            CALL  P16     ;PRINT
 4997  013411  303  153  032            JMP   MMERR   ;OUT
 4998       000               .ENDC
```

```
 5000                           .SBTTL ****KS10 CONSOLE SUBROUTINE FILE
 5001                           ;SUBROUTINE TO CHECK IF THE CPU IS RUNNING, AND IF IT IS,
 5002                           ;TO ABORT THE COMMAND THAT CALLED IT
 5003  013414  341              .RUN..: POP     H       ;MUST FIX THE STACK
 5004  013415  365                      PUSH    PSW     ;SAVE FLAGS(STATE OF THE 'C-BIT'
 5005  013416  072  165  040           LDA     RNFLG   ;CHECK SOFTWARE RUN FLAG TO SEE IF CPU CLK ON
 5006  013421  247                      ANA     A       ;IS IT ZERO? , OR NOT ZERO
 5007  013422  302  027  027           JNZ     YSRUN   ;IF NOT ZERO, JMP OFF TO PRINT MESSAGE
 5008
 5009  013425  361                      POP     PSW     ;OTHERWISE, RESTORE FLAGS
 5010  013426  311                      RET             ;AND OUT
 5011
 5012  013427                 YSRUN:   PLINE   RN.     ;MESSAGE TO SAY RUNNING..
  (1)  013427  337                      RST     3       ;PRINT LINE OF CHARS
  (1)  013430  206  037                .ADDR   RN.     ;BUFF TO PRINT
 5013  013432  303  153  032           JMP     MMERR   ;NOW GO RESTART OURSELVES
 5014
```

```
 5016                           ;ROUTINE TO REPORT A PARITY ERROR WHEN DETECTED BY THE 8080 CONSOLE
 5017  013435                   RPTPAR: CLRB    NOPNT   ;TURN TYPING ON
 5018  013437  257                      XRA     A       ;ACCUM MUST BE ZERO
 5019  013440  323  206                 OUT     CLKCTL  ;KILL CLK, SO PAR ERR CLRING WONT RELEASE THE CPU CLKS
 5020
 5021        001                 .IF     DF,SCECOD
 5022  013442  072  326  042            LDA     SC.OFF  ;GET FLAG TO SEE IF WE CAN TRY FOR SOFT RECOVER
 5023  013445  247                      ANA     A       ;SET 8080 FLAGS
 5024  013446  302  114  027            JNZ     HRDERR  ;IF FLAG .NE. 0, THEN WE WILL NOT RECOVER
 5025                           ;FIRST THING WE NEED TO DO IS CHECK FOR CRAM/CRA PARITY ERRORS, AND DECIDE
 5026                           ;IF WE CAN RECOVER FROM THEM
 5027  013451                           LONG.DELAY 1    ;LET DISK TRAFFIC STOP
  (1)  013451  041  310  000            LXI     H,200. * 1 ;SET UP THE TIMING COUNT IN H,L REG
  (1)  013454  315  134  014            CALL    LTLOOP  ;AND GO DELAY ABOUT 1 SECOND FOR EACH (300 COUNT)
 5028  013457  076  100                 MVI     A,SMPAR ;GET WHICH REG HAS PARITY INFO IN IT
 5029  013461  315  267  013            CALL    ER.UTL  ;READ DATA,PARITY BITS RETURNED IN ACCUM
 5030  013464  057                      CMA             ;INVERT SO THAT .TRUE. .EQ. HI
 5031  013465  346  022                 ANI     ^O22    ;SEE IF CRA/CRM ERROR TYPE
 5032  013467  312  114  027            JZ      HRDERR  ;IF NO, GO FOR A HARD ERROR
 5033
 5034                           ;WELL, LETS SEE IF MEM BUSY OR I/O BUSY
 5035  013472  333  102                 IN      R.BUSY  ;*****I/O RD 102*****
 5036  013474  057                      CMA             ;FIX HARDWARE COMPLIMENT
 5037  013475  346  060                 ANI     ^O60    ;SEE IF EITHER OF THOSE TWO BITS SET
 5038  013477  107                      MOV     B,A     ;SAVE RESULTS OF THIS 'AND' FOR A BIT
 5039  013500  333  101                 IN      REFERR  ;SEE IF MEM REFRESH ERR EITHER
 5040  013502  057                      CMA             ;FIX HARDWARE INVERSION
 5041  013503  346  001                 ANI     1       ;ONLY KEEP THE REFRESH BIT
 5042  013505  260                      ORA     B       ;THROW THE TWO TOGETHER..IF RESULTS .EQ. 0, THEN ALL OK
 5043  013506  312  251  027            JZ      SOFTERR ;IF NOTHING, GO TRY FOR THE SOFT ERROR RECOVERY
```

```
 5045                           ;GET TO HERE IF ERROR IS CONSIDERED NON-RECOVERABLE
 5046  013511                   NR:     PLINE   NRSCE           ;'?NR-SCE', NON RECOVERABLE-SOFT CRAM ERROR
  (1)  013511  337                       RST     3       ;PRINT LINE OF CHARS
  (1)  013512  232  027                   .ADDR   NRSCE   ;BUFF TO PRINT
 5047       000                 .ENDC
 5048
 5049
 5050  013514                   HRDERR: CLRB    CHKPAR  ;SAY NOT TO REPORT AGAIN & AGAIN
 5051  013516  315  346  006            CALL    CLRUSE  ;EXIT FROM USER MODE
 5052  013521                           PLINE   PARMSG  ;MESSAGE TO CTY
  (1)  013521  337                       RST     3       ;PRINT LINE OF CHARS
  (1)  013522  161  037                   .ADDR   PARMSG  ;BUFF TO PRINT
 5053
 5054  013524  333  100                   IN      SMPAR   ;***** I/O RD PARITY *****
 5055  013526  057                       CMA             ;FIX THE HARDWARE INVERSION
 5056  013527  315  304  030            CALL    P8BITA  ;AND PRINT IT OUT WITH THE ERROR MESSAGE
 5057  013532                           PSPACE          ;SEPARATE THE 8 BIT DATUMS
  (2)  013532  317                       RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (2)  013533  040                       .BYTE   SPACE   ;CHAR TO PRINT
 5058  013534  333  303                   IN      ^O303   ;READ 'DPM PARITY BIT'
 5059  013536  057                       CMA             ;FIX CPU INVERSION
 5060  013537  346  001                   ANI     ^O1     ;ONLY INTERESTED IN BIT 0 'DPM PAR ERR'
 5061  013541  315  304  030            CALL    P8BITA  ;AND THEN PRINT THE 'DPM PAR' DATA
 5062  013544                           PSPACE          ;AGAIN, SEPARATE BY SPACES
  (2)  013544  317                       RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (2)  013545  040                       .BYTE   SPACE   ;CHAR TO PRINT
 5063  013546  333  103                   IN      ^O103   ;READ REG THAT HAS R PAR RIGHT & R PAR LEFT
 5064  013550  057                       CMA             ;FIX THE HARDARE INVERSION
 5065  013551  346  360                   ANI     ^O360   ;KEEP ONLY THE 2 'R PAR' BITS
 5066  013553  315  031  017            CALL    P8CRLF  ;PRINT IT
 5067  013556  315  271  032            CALL    CLRRN   ;CLEAR THE SOFTWARE 'RUNNING' FLAG TOO
 5068  013561  315  315  032            CALL    LTFLT   ;PARITY ERROR LIGHTS THE FAULT LIGHT
 5069  013564  303  255  001            JMP     REINI   ;AND GO RE-INIT..PAR ERRS ARE FATAL
 5070       001                 .IF     DF,SCECOD
```

```
 5072                           ;COMMAND LIST FOR READING UBA INFO
 5073  013567                   UBA.RD: UBA, 763,001
  (1)  013567  001                      .BYTE   001 & 377
  (1)  013570  346                      .BYTE   Q,1
  (1)  013571  073                      .BYTE   Q,2
 5074  013572                         ENDLST
  (1)  013572  000                      .BYTE   0
  (1)  013573  000                      .BYTE   0
  (1)  013574  050                      .BYTE   <12*4>
 5075
 5076                           ;RH11 COMMAND LIST FOR CHECKING FOR RECOVERABLE & NON RECOVERABLE
 5077                           ;STATES OF THE CONTROLLER
 5078  013575                   RH.TST: EI, D776,P.00          ;EXAM CONTROLLER STATUS REG
  (1)  013575  374                      .BYTE   Q,1
  (1)  013576  000                      .BYTE   P.00 & 377
  (1)  013577  023                      .BYTE   Q,2
 5079  013600                         ERRTST 60000            ;TEST FOR FATAL ERRS
  (1)  013600  000  140                 .ADDR   60000
  (1)  013602  040                      .BYTE   <10*4>
 5080  013603                         EI, D776,P.12           ;EXAM DRIVE STATUS REG
  (1)  013603  374                      .BYTE   Q,1
  (1)  013604  012                      .BYTE   P.12 & 377
  (1)  013605  023                      .BYTE   Q,2
 5081  013606                         ERRTST 40000            ;CHECK THAT GUY FOR ERRS
  (1)  013606  000  100                 .ADDR   40000
  (1)  013610  040                      .BYTE   <10*4>
 5082  013611                         ENDLST                  ;HERE IF ALL OK
  (1)  013611  000                      .BYTE   0
  (1)  013612  000                      .BYTE   0
  (1)  013613  050                      .BYTE   <12*4>
 5083
 5084  013614                   RH.EXE: EI, D776,P.00          ;A TEMPLATE FOR RH EXAMINES
  (1)  013614  374                      .BYTE   Q,1
  (1)  013615  000                      .BYTE   P.00 & 377
  (1)  013616  023                      .BYTE   Q,2
 5085  013617                         ENDLST                  ;THAT'S ALL WE NEED FOR THIS
  (1)  013617  000                      .BYTE   0
  (1)  013620  000                      .BYTE   0
  (1)  013621  050                      .BYTE   <12*4>
 5086
 5087  013622  000              SAVLST: .BYTE   P.00           ;READ 776700
 5088  013623  002                      .BYTE   P.02           ;READ 776702
 5089  013624  004                      .BYTE   P.04           ;READ 776704
 5090  013625  006                      .BYTE   P.06           ;READ 776706
 5091  013626  010                      .BYTE   P.10           ;READ 776710
 5092  013627  032                      .BYTE   P.32           ;READ 776732
 5093  013630  034                      .BYTE   P.34           ;READ 776734
 5094  013631  377                      .BYTE   -1             ;END OF LIST MARKER
 5095                           ;THE ASCII MESSAGES REQUIRED FOR PARITY RECOVERY
 5096  013632  077  116  122    NRSCE:  .ASCIZ/?NR-SCE /       ;NOT RECOVERABLE SOFT CRAM ERROR
 5097  013643  045  123  103    OKSCE:  .ASCIZ/%SCE /          ;RECOVERABLE SOFT CRAM ERROR
```

```
 5099                          ;NOW GET THE CURRENT CRAM ADDRESS & CHECK FOR HARD CRAM ERRORS.
 5100                          ;(NOTE: A HARD CRAM ERROR IS A CRAM PARITY ERROR OCCURRING AT
 5101                          ;THE SAME CRAM ADDRESS MORE THAN ONCE IN A ROW)
 5102  013651  021  344  040   SOFTERR:       LXI     D,SCEADR        ;'D,E' PAIR WILL POINT AT THE DESIRED
 5103  013654  315  171  030           CALL    BREAK           ;GO CHECK IF CURRENT .EQ. DESIRED..
 5104  013657  312  114  027           JZ      HRDERR          ;IF YES, JUMP 'CAUSE ITS A HARD ERROR
 5105
 5106                          ;SOFT CRAM ERRO RECOVERY CONTINUED....
 5107                          ;RECOVERY BEGINS BY ZAPPING THE PE(1) FLOP SO WE MAY CATCH ANY ADDITIONAL
 5108                          ;PARITY ERRORS
 5109  013662  076  001               MVI     A,1             ;BIT TO RESET CRAM C.R. & PE(1)
 5110  013664  323  204               OUT     CRMCTL          ;***** I/O WRT/204 *****
 5111
 5112                          ;FALL THRU IF ADDRESS NOT THE SAME
 5113  013666  042  344  040   NOTSAME: SHLD   SCEADR          ;SAVE IT AS THE NEW 'PREVIOUS'
 5114
 5115                          ;NOW CHECK RH11 TO SEE IF THIS FAILURE IS RECOVERABLE FOR THE
 5116                          ;MONITOR. 8080 WILL SIMPLY EXECUTE A CHANNEL COMMAND LIST
 5117                          ;OF 'ERRTST' , WITH THE CORRECT ERROR BITS CHECKED FOR
 5118  013671  041  175  027           LXI     H,RH.TST        ;POINT TO ERROR CHECKER COMMAND LIST
 5119  013674  315  023  025           CALL    CHNXCT          ;EXECUTE THAT LIST
 5120  013677  332  111  027           JC      NR              ;IF BAD, SAY NOT RECOVERABLE
 5121                          ;ELSE...FALL INTO THE SOFT RECOVERY CODE
```

```
 5123                          ;FINALLY GET TO HERE IF THIS THING LOOKS RECOVERABLE, NOW WE MUST
 5124                          ;BEGIN SAVING THINGS...
 5125                          ;FIRST,, GET THE CURRENT DISK UBA NUMBER TO SET UP THE RH11 REGISTER
 5126                          ;SAVING, THEN WE MUST SET UP THE RH11 BASE REGISTER ITSELF
 5127                          ;WHILE WE ARE AT IT, MIGHT AS WELL PRINT A LITTLE MESSAGE SAYING WHAT WE ARE
 5128                          ;DOING IN HERE
 5129  013702                  RECVR:  PLINE   OKSCE           ;ERR MSG "%SCE" SOFT CRAM ERROR
  (1)  013702  337                        RST     3       ;PRINT LINE OF CHARS
  (1)  013703  243  027                   .ADDR   OKSCE   ;BUFF TO PRINT
 5130  013705  041  344  040              LXI     H,SCEADR        ;NOW POINT TO THIS BAD ADDRESS
 5131  013710  315  317  030              CALL    P16             ;AND PRINT IT
 5132  013713                           PCRLF                   ;TERMINATE EVERYTHING WITH A CR-LF
  (2)  013713  347                        RST     4
  (2)  013714  002                        .BYTE   2
 5133  013715  315  026  023              CALL    DSKDFT          ;NOW GET THOSE DEFAULTS JUST MENTIONED
 5134  013720  257                        XRA     A               ;CLEAR ACCUM
 5135  013721  323  100                   OUT     RESET           ;*****I/O WRT 100***** NO PAR CHECKING
 5136
 5137                          ;FINALLY READY TO BEGIN THE ACT OF SAVING SOME REGISTERS
 5138                          ;FIRST GET UBA LOCATION 763001
 5139  013723  041  167  027              LXI     H,UBA.RD        ;POINTER TO UBA READ-IN LIST
 5140  013726  315  023  025              CALL    CHNXCT          ;READ-IN THE UBA INFO,, NOW ITS IN EMBUF
 5141  013731                           INTON                   ;DON'T PRINT THIS
  (1)  013731  327                        RST     2       ;GO SET INTERNAL MODE
 5142  013732  315  357  012              CALL    EI1             ;ACTUALLY DO THE READ-IN
 5143  013735                           INTOFF                  ;OK NOW
  (1)  013735  367                        RST     6       ;GO EXIT FROM INTERNAL MODE
 5144
 5145                          ;BEFORE WE SAVE IT, WE WILL SET UP FOR SAVING RH REGISTERS,, THAT WAY, WE
 5146                          ;CAN USE SOME COMMON CODE FOR PUTTING DATA INTO OUR SAVE BUFFER.
 5147  013736  001  214  027              LXI     B,RH.EXE        ;NOW POINT TO A CHANNEL COMMAND LIST
 5148  013741  021  172  040              LXI     D,RM100         ;AND POINT TO A PLACE TO PUT THE LIST
 5149  013744  076  006                   MVI     A,6             ;IT TAKES SIX BYTES TO MOVE THE ENTIRE LIST
 5150  013746  315  011  033              CALL    M5B             ;MOVE THE STUFF TO RAM
```

```
 5152                           ;THE CODE TO MOVE BYTES REQUIRES 'B,C' POINTING TO THE SOURCE
 5153                           ;AND 'D,E' POINTING TO THE DESTINATION
 5154                           ;AND 'A' HAVING HOW MANY BYTES TO BE MOVED.. 'MOVREG' DOES 5 BYTES
 5155                           ;FREE OF CHARGE..B,C AND D,E ARE UPDATED BY THE NUMBER OF BYTES MOVED.
 5156  013751  041  222  027          LXI     H,SAVLST        ;POINT TO A LIST OF THINGS WHICH WE MUST SAVE
 5157  013754  345                   PUSH    H               ;PLACE IT IN THE RAM FOR SAFE KEEPING
 5158  013755  021  327  042          LXI     D,RHSAVE        ;D,E GETS THE POINTER TO THE SAVE AREA
 5159  013760  001  012  040  RH.LP: LXI     B,EMBUF         ;WE WILL ALWAYS BE MOVING STUFF FROM 'EMBUF'
 5160  013763  315  007  033          CALL    MOVREG          ;MOVE 5 BYTES, PLEASE
 5161
 5162                           ;TO SAVE THE RH REGISTERS, WE WILL PUT A TINY CHANNEL COMMAND
 5163                           ;LIST INTO RAM SPACE, THEN EXECUTE IT, CHANGING THE DESIRED REGISTERS
 5164                           ;BETWEEN READS, WHICH WILL GIVE US A CHANCE TO SAVE THE RESULTS
 5165                           ;OF THE READ
 5166  013766  341                   POP     H               ;GET POINTER INTO 'REG TO BE SAVED' BUFFER
 5167  013767  176                   MOV     A,M             ;GET CURRENT BYTE INTO ACCUM
 5168  013770  062  173  040          STA     RM100+1         ;PUT BYTE INTO BUFFER SO CAN BE EXECUTED
 5169  013773  074                   INR     A               ;UP BY ONE
 5170  013774  312  014  030          JZ      SCE.GO          ;IF THAT MAKES .EQ. 0, THEN OUT
 5171
 5172  013777  043                   INX     H               ;ELSE UPDATE POINTER
 5173  014000  345                   PUSH    H               ;SAVE THE POINTER TOO
 5174  014001  041  172  040          LXI     H,RM100         ;PREPARE TO EXECUTE THE READIN BUFFER
 5175  014004  325                   PUSH    D               ;NOW SAVE THE SPOT WE ARE IN IN THE RHSAVE AREA
 5176  014005  315  023  025          CALL    CHNXCT          ;DO IT..ONE RH REGISTER IS IN
 5177  014010  321                   POP     D               ;RETRIEVE POINTER TO THE DATA SAVE SPACE
 5178  014011  303  360  027          JMP     RH.LP           ;SAVE INFO, AND READ IN NEXT RH REGISTER
```

```
 5180                           ;NOW DATA HAS BEEN SAVED, WE CAN ACTUALLY BEGIN TO RESTORE THE MICRO-CODE
 5181  014014                   SCE.GO: INTON                   ;SET INTERNAL MODE
  (1)  014014  327                      RST     2       ;GO SET INTERNAL MODE
 5182  014015  021  002  002           LXI     D,^O1002        ;PLACE IN DISK PAGE OF POINTERS TO MICRO-CODE
 5183  014020  315  273  022           CALL    FILESH          ;GO READ IN THE FIRST PAGE OF MICRO-CODE
 5184  014023  332  175  032             JC    C.BTERR         ;IF ERR ITS ALL OVER
 5185
 5186  014026  315  043  023           CALL    DMEM2CR         ;GO LOAD CRAM
 5187
 5188  014031  052  344  040           LHLD    SCEADR          ;GET ADDRESS AT WHICH TO CONTINUE
 5189  014034  315  320  014           CALL    CADWR           ;SET THE CRAM ADDRESS TO THE GUY THAT SLIPPED
 5190                           ;WE ARE NEARING THE END OF THIS RECOVERY STUFF.. WE MUST RESTORE THE STATE
 5191                           ;OF THE RH11 AND UBA TO WHAT IT WAS BEFORE WE STARTED, THEN WE CAN TURN THE
 5192                           ;CLOCKS ON AGAIN
 5193  014037  041  167  027           LXI     H,UBA.RD        ;POINT TO A UBA READ CHANNEL COMMAND LIST
 5194  014042  315  023  025           CALL    CHNXCT          ;SET THE I/O ADDRESS TO A UBA PAGE REGISTER
 5195
 5196                           ;NOW FIX UP THE FORMAT BETWEEN A READ OF THE UBA PAGING REG, AND
 5197                           ;THE WRITE WE WISH TO DO TO THE PAGING RAM
 5198  014045  072  332  042           LDA     RHSAVE+3        ;GET THE BYTE THAT HAS THE CURRENT CNTRL BITS
 5199  014050  346  170                ANI     ^O170           ;OFF JUNK, KEEP ONLY 4 RELEVANT BITS
 5200  014052  117                      MOV     C,A             ;SAVE IN THE C REG
 5201  014053  041  327  042           LXI     H,RHSAVE        ;NOW POINT TO OUR BUFFER WITH THE DESIRED INFO
 5202  014056  315  154  031           CALL    SHR36           ;SHIFT DATA RIGHT, 4 PLACES
 5203  014061  004                      .BYTE   4
 5204
 5205  014062  171                      MOV     A,C             ;GET OUR CONTROL BITS BACK
 5206  014063  062  331  042           STA     RHSAVE+2        ;PLOP THEM INTO THE 36 BIT WORD
 5207  014066  315  154  031           CALL    SHR36           ;NOW SHIFT THE WHOLE MESS 5 MORE PLACES
 5208  014071  005                      .BYTE   5
 5209
 5210                           ;AND THATS IT.. MOVE STUFF TO A DEPOSIT BUFFER
 5211  014072                           MOV5B                   ;A "MOVE"
  (1)  014072  347                      RST     4
  (1)  014073  000                      .BYTE   0
 5212  014074  327  042                .ADDR   RHSAVE          ;FROM THE SAVE BUFFER
 5213  014076  047  040                .ADDR   DMDAT           ; TO THE DEPOSIT BUFFER
 5214  014100  315  016  013           CALL    DI1             ;WRITE THIS MESS BACK TO UBA PAGING RAM
```

```
 5216                          ;NOW WE WANT TO WRITE BACK THE RH11 REGISTERS THAT WE SAVED, THEN DESTROYED
 5217  014103  041  214  027          LXI     H,RH.EXE        ;NOW SET THE RH11 REGISTER I.D. INTO THE
 5218  014106  315  023  025          CALL    CHNXCT          ;INTERNAL BUFFER 'IOAD',INCLUDING UBA NUMBER
 5219
 5220  014111  041  222  027          LXI     H,SAVLST        ;POINT TO LIST OF REGS TO BE RESTORED
 5221  014114  345                    PUSH    H               ;SAVE THIS INFO ON THE STACK
 5222  014115  001  334  042          LXI     B,RHSAVE+5      ;WILL BEGIN MOVING STUFF WE SAVED FROM RH
 5223  014120  021  047  040  DI.LP: LXI     D,DMDAT         ;ALWAYS MOVE THE STUFF TO THE DEPOSIT BUFFER
 5224  014123  315  007  033          CALL    MOVREG          ;MOVE THE STUFF INTO 'DMDAT'
 5225  014126  341                    POP     H               ;GET OUR LITTLE LIST POINTER
 5226  014127  072  024  040          LDA     IOAD            ;GET CURRENT OFFSET INTO RH
 5227  014132  346  300               ANI     ^O300           ;THROW AWAY CURRENT OFFSET
 5228  014134  266                    ORA     M               ;THROW OUR DESIRED OFFSET INTO THE WORD
 5229  014135  062  024  040          STA     IOAD            ;PUT IT BACK INTO THE IOAD BUFFER
 5230  014140  176                    MOV     A,M             ;GET THE OFFSET WE JUST MESSED WITH
 5231  014141  074                    INR     A               ;TEST TO SEE IF END OF LIST
 5232  014142  312  157  030          JZ      CONT.I          ;IF END OF LIST, FINISH THE RECOVERY
 5233
 5234                          ;NOT END OF LIST, MUST SAVE SOME MORE RH REGISTERS
 5235  014145  043                    INX     H               ;UPDATE THE LIST POINTER
 5236  014146  345                    PUSH    H               ;SAVE THE POINTER
 5237  014147  305                    PUSH    B               ;SAVE POINTER TO THE SAVED DATA IN 'RHSAVE'
 5238  014150  315  016  013          CALL    DI1             ;NOW RESTORE THIS RH REGISTER
 5239  014153  301                    POP     B               ;RESTORE POINTER INTO BUFFER
 5240  014154  303  120  030          JMP     DI.LP           ;CONTINUE
 5241
 5242  014157  315  356  016  CONT.I: CALL    SMFINI          ;GET CURRENT PARITY DEFAULTS & WRITE THEM OUT
 5243  014162  315  214  014          CALL    CSCMD           ;TURN THE CLOCK BACK ON
 5244  014165                         INTOFF                  ;NO MORE INTERNAL MODE
  (1)  014165  367                    RST     6       ;GO EXIT FROM INTERNAL MODE
 5245  014166  303  335  001          JMP     NULLJ           ;GO
 5246       000               .ENDC
```

```
 5248                           ;ROUTINE TO CHECK IF CURRENT RAM ADDRESS IS THE DESIRED BREA
 5249                           ;ADDRESS..
 5250  014171  325              BREAK:  PUSH    D       ;'D,E' HAS POINTER TO DESIRED ADDR, SAVE IT PLEASE
 5251  014172  076  003                 MVI     A,3     ;DIAG FUNC TO READ CURRENT RAM ADDR
 5252  014174  315  362  014           CALL    READC   ;GO DO FUNCTION READ
 5253  014177  321                     POP     D       ;NOW MAKE 'D,E' POINT AT DESIRED AGAIN
 5254
 5255  014200  052  100  040           LHLD    TMPB2   ;GET CURRENT ADDRESS..
 5256  014203  174                     MOV     A,H     ;ALSO MAKE SURE THE CURRENT JUST READ IS 11 BITS
 5257  014204  346  007                 ANI     ^O7     ;8 BITS LO HALF, PLUS 3 BITS HI HALF
 5258  014206  147                     MOV     H,A     ;NOW PUT THE WHOLE MESS BACK
 5259
 5260  014207  032                     LDAX    D       ;GET LO ORDER PIECE TO ACCUM
 5261  014210  275                     CMP     L       ;CHECK VERSUS JUST READ
 5262  014211  300                     RNZ             ;IF .NE. 0, THEN NO MATCH, SO OUT..
 5263  014212  023                     INX     D       ;OK, SO UPDATE PNTR TO READ-IN
 5264  014213  032                     LDAX    D       ;GET HI ORDER OF DESIRED PIECE
 5265  014214  274                     CMP     H       ;COMPARE, SIGNS TAKE CARE OF THEMSELVES
 5266  014215  311                     RET             ;IF RESULT OF ADD WAS ZERO, GOOD.IF NOT,OK TOO..
 5267
 5268                           ;ROUTINE TO DO SHORT FORM OF EXAMINE MEMORY
 5269                           ;ENTER WITH 'D,E' CONTAINING SHORT ADDRESS
 5270  014216  067              EXAMSH: STC             ;SET C-BIT FOR LATER USE IN COMMON CODE
 5271  014217  343              DEPSHT: XTHL            ;SWAP SO H,L POINTS TO TRAILING ARG
 5272  014220  315  050  033           CALL    TARG1   ;COLLECT TRAILING ARG INTO 'D,E'
 5273  014223  343                     XTHL            ;SWAP BACK SO THAT THINGS ARE RIGHT
 5274  014224  353                     XCHG            ;AND NOW MAKE 'H,L' HOLD THE TRAILING ARG
 5275  014225  042  235  040   EXMHL:  SHLD    SHRTAD  ;STORE SHORT ADDRESS IN THE RAM
 5276  014230  021  235  040           LXI     D,SHRTAD ;DE, GETS REPLACE WITH A POINTER TO SHORT ADDRESS
 5277  014233  365                     PUSH    PSW     ;SAVE THE C-BIT FOR LATER USE
 5278  014234  334  041  012           CC      EMINT   ;IF C WAS SET, GO DO AN EXAMINE
 5279  014237  361                     POP     PSW     ;GET FLAGS AS THEY WERE
 5280  014240  324  237  012           CNC     DMINT   ;IF C WAS CLR DO A DEPOSIT
 5281  014243  311                     RET             ;NOW OK TO RETURN
 5282
 5283                           ;ROUTINE THAT EXECUTES AN 'ARG16', THEN RETURNS THE DATA IN 'H,L'
 5284  014244                   ARG16.: ARG16           ;ARGUMENT ASSEMBLER
  (1)  014244  347                     RST     4
  (1)  014245  004                     .BYTE   4
 5285  014246  000  040                .ADDR   T8ODT   ;USE A TEMP LOCATION
 5286
 5287  014250  052  000  040           LHLD    T8ODT   ;GATHER DATA INTO H,L
 5288  014253  311                     RET             ;AND BACK
```

```
 5290                        ;SUBROUTINE TO PRINT A SINGLE 8-BIT BYTE AS OCTAL DATA
 5291                        ;OF THE FORM: XXX
 5292                        ;NO REGS DESTROYED.,PNTR TO 8-BIT BYTE PASSED IN 'H,L'
 5293  014254  345           P8BIT:  PUSH    H       ;SAVE ALL REGISTERS
 5294  014255  305           P8BIT1: PUSH    B
 5295  014256  325                   PUSH    D
 5296  014257  365                   PUSH    PSW
 5297  014260  315  032  031        CALL    OCTAL   ;CREATE 8-BIT BUFFER AS A 3 OCTAL CHARACTERS
 5298  014263  001                   .BYTE   1       ;ONE BYTE OF BINARY DATA INVOLVED
 5299  014264  003                   .BYTE   3       ;WANT ONLY 3 OCTAL CHARS
 5300  014265  016  003             MVI     C,03    ;NUM CHARS TO PRINT
 5301  014267  361           P8LP:   POP     PSW     ;CHAR OFF TOP OF STACK
 5302  014270  315  010  004        CALL    PCHR    ;AND GO PRINT IT
 5303  014273  015                   DCR     C       ;DOWN COUNT
 5304  014274  302  267  030        JNZ     P8LP    ;AND CONTINUE TILL DONE ALL 3
 5305
 5306  014277  361                   POP     PSW     ;RESTORE REGS
 5307  014300  321                   POP     D
 5308  014301  301                   POP     B
 5309  014302  341                   POP     H
 5310  014303  311                   RET             ;AND ALL DONE
 5311
 5312
 5313                        ;WHEN DOING A 'P8BIT' WITH THE DATA PASSED IN THE ACCUM
 5314                        ;INSTEAD OF BEING POINTED TO BY H,L THEN COME HERE
 5315  014304  345           P8BITA: PUSH    H       ;MUST SAVE H,L HERE, SO WE CAN MESS IT UP
 5316  014305  041  122  040        LXI     H,P8.TMP ;KEEP A PLACE FOR PRINTING DATA
 5317  014310  167                   MOV     M,A     ;PUT THE THING TO BE PRINTED IN THE RAM SPACE
 5318  014311  303  255  030        JMP     P8BIT1  ;GO TO COMMON CODE
```

```
 5320                           ;SUBROUTINE TO PRINT 16-BITS WORTH OF DATA..
 5321                           ;POINTER TO THAT DATA IS PASSED TO THE ROUTINE IN
 5322                           ;REGISTER 'H,L'
 5323  014314  041  100  040    P16.:   LXI     H,TMPB2 ;IN THIS TYPE CALL, WE LOAD H,L AUTOMATICALLY
 5324  014317  365              P16:    PUSH    PSW     ;SAVE ALL THE REGISTERS
 5325  014320  305                      PUSH    B
 5326  014321  325                      PUSH    D
 5327  014322  345                      PUSH    H
 5328
 5329  014323  315  032  031            CALL    OCTAL   ;CREATE OCTAL CHARS FROM THE 16-BIT DATA
 5330  014326  002                      .BYTE   2       ;WE HAVE 2 BYTES OF RELEVANT DATA
 5331  014327  006                      .BYTE   6       ;AND WE WANT 6 OCTAL CHARS TO PRINT
 5332
 5333  014330  006  006                MVI     B,6     ;ON RETURN WE WANT TO PRINT 6 CHARS
 5334  014332  361              P16LP:  POP     PSW     ;GET OCTAL CHAR OFF STACK
 5335  014333  315  010  004            CALL    PCHR    ;AND GO PRINT IT
 5336  014336  005                      DCR     B       ;DOWN THE COUNT
 5337  014337  302  332  030            JNZ     P16LP   ;BACK TO PRINT MORE TILL ALL DONE
 5338
 5339  014342  341                      POP     H       ;DONE..NOW RESTORE ALL REGS
 5340  014343  321                      POP     D
 5341  014344  301                      POP     B
 5342  014345  361                      POP     PSW
 5343  014346  311                      RET             ;AND RETURN
```

```
 5345                           ;SUBROUTINE PRINT 36-BIT BINARY DATA AS
 5346                           ;A 12-OCTAL DIGIT CHARACTER STRING IN THE FORM:
 5347                           ;       XXXXXX,,XXXXXX -NO REGS DESTROYED-BIN DATA PNTR PASSED IN 'H,L'
 5348  014347  041  012  040   P36.:   LXI     H,EMBUF ;IN THIS CALL, WE LOAD H,L AUTOMATICALLY
 5349  014352  365             P36:    PUSH    PSW     ;WILL SAVE ALL REGS IN HERE
 5350  014353  305                     PUSH    B
 5351  014354  325                     PUSH    D
 5352  014355  345                     PUSH    H
 5353  014356  315  032  031          CALL    OCTAL   ;CREATE 36-BIT BUFFER AS AN OCTAL CHARACTER STRING
 5354  014361  005                     .BYTE   5       ;5BYTES REQUIRED BY 36-BITS
 5355  014362  014                     .BYTE   ^D12    ;WANT 12 OCTAL DIGITS
 5356
 5357  014363  315  002  031          CALL    PHALF   ;PRINT 18 BITS
 5358                           ;IF HERE, JUST FINISHED FIRST PASS, NEED 2 COMMAS.
 5359  014366                         PCHAR   COMMA
  (1)  014366  317                     RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (1)  014367  054                     .BYTE   COMMA   ;CHAR TO PRINT
 5360  014370                         PCHAR   COMMA   ;PRINT ',,'
  (1)  014370  317                     RST     1       ;GO PRINT CHAR IN TRAILING BYTE
  (1)  014371  054                     .BYTE   COMMA   ;CHAR TO PRINT
 5361  014372  315  002  031   P36RH:  CALL    PHALF   ;PRINT 18 MORE BITS...
 5362  014375  341                     POP     H       ;NOW RESTORE ALL THE REGS
 5363  014376  321                     POP     D
 5364  014377  301                     POP     B
 5365  014400  361                     POP     PSW
 5366  014401  311                     RET             ;RETURN
 5367
 5368                           ;ROUTINE PRINTS 18 BITS AS 6 OCTAL CHARS
 5369  014402  341             PHALF:  POP     H       ;GET A RETURN ADDR OFF STACK,SO STACK IS CLR
 5370  014403  006  006                 MVI     B,6     ;'B' WILL BE A COUNTER, IN IT WITH 3
 5371  014405  361             P36L2:  POP     PSW     ;CHARACTERS TO ACCUM
 5372  014406  315  010  004          CALL    PCHR    ;PRINT IT
 5373
 5374  014411  005                     DCR     B       ;DOWN COUNT
 5375  014412  302  005  031          JNZ     P36L2   ;CONTINUE IF NOT DONE 6 CHARS YET
 5376  014415  351                     PCHL            ;RETURN
 5377
 5378
 5379                           ;ROUTINE TO ALONE PRINT 18 BITS
 5380  014416  365             P18:    PUSH    PSW     ;WILL SAVE ALL REGS IN HERE
 5381  014417  305                     PUSH    B
 5382  014420  325                     PUSH    D
 5383  014421  345                     PUSH    H
 5384  014422  315  032  031          CALL    OCTAL   ;CREATE 6 OCTAL CHARS
 5385  014425  003                     .BYTE   3       ;3 BYTES HAVE RELEVANT DATA
 5386  014426  006                     .BYTE   6       ;AND DESIRE 6 OCTAL CHARS
 5387
 5388  014427  303  372  030          JMP     P36RH   ;GO TO RIGHT HALF PRINTER FROM 'P36'
```

```
 5390                           ;UNIVERSAL BINARY DATA TO ASCII CHARACTER SUBROUTINE
 5391                           ;ROUTINE DESTROYS THE CONTENTS OF ALL REGISTERS
 5392                           ;PASS POINTER TO BINARY DATA IN 'H,L' ,THEN CALL APPROPRIATE
 5393                           ;COVERSION DESIRED WITH 2 TRAILING PARAMETERS-CHARS PLACED ON STACK, MSB-LSB
 5394                           ;       CALL    ROUTINE
 5395                           ;       DB      XX      ;NUMBER OF BYTES HOLDING RELEVANT BINARY DATA
 5396                           ;       DB      YY      ;NUMBER OF ASCII CHARACTERS TO BE GENERATED
 5397                           ;ROUTINES ARE 'OCTAL', 'BINRY', AND 'HEXIDECIMAL'
 5398
 5399
 5400                           ;CODE WAS NEVER USED,,KEEP TEXT HERE JUST IN CASE WE EVER NEED TO ADD IT,,
 5401                           ;THIS WAY WE WON'T HAVE TO FIGURE IT ALL OUT AGAIN
 5402                           ;BINRY: LXI     D,^O401 ;LOAD D=1,E=1
 5403                           ;       JMP     COMEN   ;GO TO COMMON CODE
 5404
 5405  014432  021  007  003   OCTAL: LXI     D,^O1407        ;LOAD D=3,E=7
 5406  014435  353                 XCHG            ;SWAP 'D,E' WITH 'H,L'
 5407  014436  042  202  040         SHLD    BTMSK   ;SET DATA INTO 'BTMSK' & 'BTNUM'
 5408
 5409                           ;POINTER TO BIN DATA NOW IN 'D,E'
 5410  014441  343                 XTHL            ;'H,L' NOW POINTS TO TRAILING ARGS
 5411  014442  106                 MOV     B,M     ;BOMB 'B' REGISTER, NOW CONTAINS 'NUM' BYTES
 5412  014443  043                 INX     H       ;UPDATE PNTR
 5413  014444  116                 MOV     C,M     ;NUM CHARS INTO C
 5414  014445  043                 INX     H       ;AND UPDATE PNTR TO RET ADDR
 5415
 5416  014446  042  217  040         SHLD    HLSAVE  ;SAVE RETURN IN RAM
 5417  014451  341                 POP     H       ;CLEAR OLD STUFF OFF STACK
 5418  014452  305                 PUSH    B       ;SAVE 'B,C' JUST TEMPORARILY
 5419
 5420  014453  041  073  040         LXI     H,TMPBF2        ;'H,L' NOW POINTS TO TEMPORARY BUFFER
 5421                           ;IF FALL TO HERE, MUST MOVE DATA FROM BINARY BUFFER TO TEMP BUFFER
 5422  014456  032             OCTL1: LDAX    D       ;DATA POINTED TO BY 'D,E' TO ACCUM
 5423  014457  023                 INX     D       ;BUMP POINTER
 5424  014460  167                 MOV     M,A     ;SET THAT DATA IN RAM
 5425  014461  043                 INX     H       ;BUMP POINTER
 5426  014462  005                 DCR     B       ;DONE ALL BYTES YET?
 5427  014463  302  056  031         JNZ     OCTL1   ;BACK TILL MOVED ALL.
 5428
 5429  014466  301                 POP     B       ;RESTORE 'B,C' AND GO
 5430
 5431  014467  041  073  040         LXI     H,TMPBF2        ;'H,L' NOW PNTS TO TMP BUFF
 5432  014472  257                 XRA     A       ;CLEAR ACCUM
 5433  014473  127                 MOV     D,A     ;CLEAR 'D' REG
 5434  014474  130                 MOV     E,B     ;BYTE COUNT TO 'E'
 5435  014475  035                 DCR     E       ;BUFF IS ALWAYS 1 LESS THAN BYTE COUNT
 5436  014476  031                 DAD     D       ;'H,L' GETS BUFF ADDR PLUS BYTE COUNT
 5437  014477  042  233  040         SHLD    OCTSV   ;AND SAVE THIS ADDR IN THE RAM
```

```
 5439                           ;HERE WILL BEGIN TRANSLATION FROM BINARY TO CHARACTERS
 5440  014502  041  073  040  OCTLC: LXI     H,TMPBF2        ;'H,L' NOW POINTS TO TEMPORARY BUFFER
 5441  014505  072  202  040         LDA     BTMSK   ;NOW GET BIT MASK
 5442  014510  246                  ANA     M       ;AND TO KEEP ONLY DESIRED CHARS
 5443
 5444  014511  306  060            ADI     ^O60    ;MAKE INTO ASCII
 5445                           ;NOW MUST ROTATE ENTIRE BUFFER 3 PLACES TO RID OURSELVES OF CHAR JUST PROCESSED
 5446  014513  365                  PUSH    PSW     ;NOW SAVE CHARACTER WEVE JUST CREATED
 5447  014514  015                  DCR     C       ;DOWN THE CHAR COUNT
 5448  014515  312  150  031         JZ      OCTL5   ;JUMP OUT IF PROCESSED ALL CHARS
 5449  014520  072  203  040         LDA     BTNUM   ;GET NUM BITS INTO ACCUM
 5450  014523  127                  MOV     D,A     ;'D' GETS INITAL COUNT OF BITS
 5451
 5452  014524  130            OCTL3: MOV     E,B     ;'E' GETS BYTE COUNT
 5453  014525  052  233  040         LHLD    OCTSV   ;GET UPDATED BUFF PNTR TO 'H,L'
 5454  014530  247                  ANA     A       ;CLEAR 'C-BIT'
 5455
 5456  014531  176            OCTL4: MOV     A,M     ;GROUP OF BINARY BITS TO ACCUM
 5457  014532  037                  RAR             ;BIT 0 INTO 'C' BIT
 5458  014533  167                  MOV     M,A     ;AND SHIFTED DATA BACK INTO MEM
 5459  014534  053                  DCX     H       ;STEP UP IN THE BUFFER (UPSIDE-DOWN BUFFER)
 5460  014535  035                  DCR     E       ;DOWN BYTE COUNT
 5461  014536  302  131  031         JNZ     OCTL4   ;CONTINUE WITH BUFFER
 5462  014541  025                  DCR     D       ;DONE BUFFERS WORTH, SEE IF DONE ALL 3 BITS WORTH
 5463  014542  302  124  031         JNZ     OCTL3
 5464                           ;DONE THE 3-BITS, NOW CONTINUE WITH NEXT CHARACTER
 5465  014545  303  102  031         JMP     OCTLC   ;GO PROCESS NEXT CHARACTER
 5466
 5467                           ;HERE WHEN DONE ALL CHARS.
 5468  014550  052  217  040  OCTL5: LHLD    HLSAVE  ;GRAB THE RETURN ADDRESS
 5469  014553  351                  PCHL            ;AND RETURN
```

```
 5471                           ;SUBROUTINE TO SHIFT 36-BIT DATA BUFFER SOME
 5472                           ;NUMBER OF PLACES TO THE RIGHT.. ADDRESS OF BUFFER TO BE
 5473                           ;SHIFTED IS PASSED IN 'H,L'..NUMBER OF PLACES FOR IT
 5474                           ;TO BE SHIFTED IS PASSED AS A TRAILING PARAMETER
 5475                           ;IN A BYTE TRAILING THE SUBROUTINE CALL
 5476                           ;       CALL    SHR36
 5477                           ;       .BYTE   XX      ;NUM PLACES TO SHIFT
 5478
 5479  014554  076  005        SHR36:  MVI     A,5     ;A SHIFT 36 REQUIRES 5 BYTES TO BE MOVED
 5480  014556                  SHR24:
 5481  014556  343             SHRGO:  XTHL            ;POINTER TO TRAILING BYTE INTO 'H,L'
 5482  014557  305                     PUSH    B       ;SAVE ALL THE REGISTERS
 5483  014560  106                     MOV     B,M     ;NUMBER PLACES  TO SHIFT IN 'B'
 5484
 5485  014561  063                     INX     SP      ;BUMP STACK POINTER AROUND THE SAVED 'B,C'
 5486  014562  063                     INX     SP
 5487  014563  043                     INX     H       ;NOW BUMP RETURN ADDRESS PAST THE TRAILING ARG
 5488  014564  343                     XTHL            ;AND PUT IT BACK ONTO THE STACK
 5489  014565  073                     DCX     SP      ;NOW FIX STACK SO THAT SAVED 'B,C' IS BACK ON TOP
 5490  014566  073                     DCX     SP
 5491  014567  325                     PUSH    D
 5492  014570  345                     PUSH    H
 5493
 5494  014571  137                     MOV     E,A     ;NUMBER OF BYTES INVOLVED WAS IN A, SAVE IN E
 5495
 5496  014572  075                     DCR     A       ;FIX COUNT
 5497  014573  205                     ADD     L       ;AND NOW ADD COUNT TO 'H,L' ADDRESS
 5498  014574  157                     MOV     L,A     ;PUT BACK INTO  LO ORDER
 5499  014575  174                     MOV     A,H     ;GET HI PIECE
 5500  014576  316  000                ACI     0       ;AND ADD A CARRY TO H IF THERE WAS ONE
 5501  014600  147                     MOV     H,A     ;NOW PUT IT BACK
 5502
 5503  014601  345                     PUSH    H       ;AND PUT ON STACK FOR REUSE
 5504
 5505  014602  341             S36BL:  POP     H       ;FETCH SAVED, UPDATED ADDRESS
 5506  014603  345                     PUSH    H       ;NOW SAVE 'H,L'
 5507  014604  113                     MOV     C,E     ;GET NUMBER BYTE INVOLVED INTO C AGAIN
 5508  014605  247                     ANA     A       ;CLEAR THE C-BIT
 5509
 5510  014606  176             S36LP:  MOV     A,M     ;GET BYTE FROM BUFF TO ACCUM
 5511  014607  037                     RAR             ;SHIFT IT RIGHT
 5512  014610  167                     MOV     M,A     ;NOW PUT IT BACK INTO THE BUFFER
 5513  014611  053                     DCX     H       ;NEXT BYTE
 5514
 5515  014612  015                     DCR     C       ;WAIT!..HAVE WE DONE ALL BYTES YET??
 5516  014613  302  206  031          JNZ     S36LP   ;BACK INTO LOOP IF NOT YET
```

```
 5518                           ;FALL THRU WHEN DONE THE 5 BYTES
 5519  014616  005                     DCR     B       ;DONE NUMBER OF TIMES YET??
 5520  014617  302  202  031           JNZ     S36BL   ;JUMP IF YES DONE ALL
 5521
 5522                           ;HERE WHEN ALL DONE
 5523  014622  341                     POP     H       ;RESTORE THE 'H,L'+X
 5524  014623  341                     POP     H       ;RESTORE 'B,C'
 5525  014624  321                     POP     D       ;RESTORE 'D,E'
 5526  014625  301                     POP     B       ;RESTORE 'H,L'
 5527  014626  311                     RET
```

```
 5529                           ;ROUTINE TO ASSEMBLE 16-BIT ARGUMENT
 5530                           ;PLACE TO PUT 16-BIT DATA PASSED AS A TRAILING
 5531                           ;ARG... 'B,C' IS MESSED UP BY THIS ROUTINE
 5532
 5533                           ;       CALL    ARG16
 5534  014627  076  014        ARG96:  MVI     A,12.   ;NUMB BYTES USED IN ARG96 IS 12
 5535  014631  303  244  031          JMP     ARGBG1
 5536
 5537  014634  076  005        .ARG36: MVI     A,5     ;NUMB BYTES USED IN ARG36 IS 5
 5538  014636  303  243  031          JMP     ARGBEG
 5539
 5540  014641  076  002        .ARG16: MVI     A,2     ;NUMB BYTES USED IN ARG16 IS 2
 5541  014643  341              ARGBEG: POP     H       ;GET H,L FROM TRAP HANDLER
 5542  014644  062  214  040   ARGBG1: STA     CHRCNT  ;SAVE IT
 5543  014647  072  213  040          LDA     RPTON   ;IS THIS A REPEAT??
 5544  014652  247                   ANA     A       ;CHECK THE REPEAT FLAG
 5545  014653  302  005  032          JNZ     CLEAN   ;JUMP IF YES A REPEAT....
 5546
 5547                           ;ELSE FALL THRU AND 'GET' CHARACTERS AS BINARY DATA
 5548  014656  052  223  040          LHLD    .ARG1   ;GET PNTR TO ARG
 5549  014661  001  000  000          LXI     B,00    ;CLR REGS 'B,C'
 5550
 5551  014664  176              GETLP: MOV     A,M     ;GET AN ASCII CHARACTER
 5552  014665  326  060                SUI     ^O60    ;OTHERWISE, OFF ASCII STUFF
 5553  014667  346  370                ANI     ^O370   ;NOW BE SURE IT WAS NUM AND NOT CHAR
 5554  014671  302  305  031          JNZ     GETEN   ;IF HAD BITS GO SEE IF PROPER ENDING
 5555  014674  176                    MOV     A,M     ;MESSED UP CHAR, GET IT BACK
 5556  014675  326  060                SUI     ^O60    ;OFF THE ASCII AGAIN
 5557  014677  365                    PUSH    PSW     ;SO WE CAN STACK IT
 5558
 5559  014700  014                    INR     C       ;WAS OK.. SO UP COUNT
 5560  014701  043                    INX     H       ;AND UPDATE TO NEXT CHAR
 5561  014702  303  264  031          JMP     GETLP   ;AND CONTINUE TILL DONE
 5562
 5563                           ;JUMP TO HERE WHEN STACKED ALL THE CHARS
 5564  014705  315  245  033   GETEN: CALL    SEPCHR  ;THROW OUT TRAILING SPACES & TABS
 5565  014710  042  223  040          SHLD    .ARG1   ;SAVE 'H,L'
 5566  014713  315  344  032          CALL    EOCML   ;END OF LINE??
 5567  014716  322  150  032          JNC     KILNM   ;IF NOT WE HAVE A PROBLEM
 5568
 5569  014721  052  207  040          LHLD    RPBUFS  ;GET PNTR TO REPEAT DATA BUFFER
 5570  014724  257                    XRA     A       ;CLR ACCM
 5571  014725  167                    MOV     M,A     ;CLR THE 'BYTE' COUNTER
 5572  014726  043                    INX     H       ;UPDATE POINTER
 5573
 5574  014727  135                    MOV     E,L     ;COPY POINTER INTO 'D,E' REG
 5575  014730  124                    MOV     D,H
 5576  014731  015                    DCR     C       ;MAKE C START AT COUNT-1
 5577
```

```
 5579  014732  325            RPINCB: PUSH    D       ;H,L SHOULD BE SAME AS D,E
 5580  014733  341                    POP     H       ;SO DO IT USING STACK
 5581  014734  023                    INX     D       ;EXCEPT D,E SHOULD BE
 5582  014735  023                    INX     D       ;3 GREATER
 5583  014736  023                    INX     D
 5584
 5585  014737  345                    PUSH    H       ;SAVE H,L FOR A MINUTE
 5586  014740  052  207  040          LHLD    RPBUFS  ;WHILE THE 'BYTE' COUNTER GETS UPDATED
 5587  014743  176                    MOV     A,M     ;COPY CURRENT COUNT
 5588  014744  306  003              ADI     3       ;UPDATE BY 3
 5589  014746  167                    MOV     M,A     ;NOW PUT IT BACK
 5590  014747  341                    POP     H       ;AND FIX UP H,L
 5591
 5592  014750  006  010              MVI     B,8.    ;B GETS A COUNT OF 8 FOR OUR LOOP
 5593  014752  361            A16PK:  POP     PSW     ;GET 3-BIT BINARY
 5594
 5595  014753  022            ARGQQ:  STAX    D       ;AND PUT INTO TMP BUFFER
 5596
 5597                         ;DONT MESS UP 'H,L', ITS NEEDED BY 'SHR36' ROUTINE
 5598  014754  315  154  031          CALL    SHR36   ;SHIFT THING 36 PLACES
 5599  014757  003                    .BYTE   3       ;SHIFT 3 PLACES
 5600
 5601  014760  015                    DCR     C       ;DOWN CHAR COUNT
 5602  014761  362  374  031          JP      NOTRK   ;FOOL THE 8-TIME LOOP IF GOES MINUS
 5603  014764  257                    XRA     A       ;CLR ACCUM,IN ORDER TO PAD WITH ZEROES
 5604  014765  005                    DCR     B       ;DOWN OUR '8' COUNTER
 5605  014766  312  000  032          JZ      CHKSTK  ;AND OUT IF ZERO
 5606  014771  303  353  031          JMP     ARGQQ   ;ELSE CONTINUE LOOPING
 5607
 5608  014774  005            NOTRK:  DCR     B       ;DOWN COUNT THE 8-TIME LOOP
 5609  014775  302  352  031          JNZ     A16PK   ;IF STILL DOING 8-TIMES
 5610
 5611                         ;DONE 8-TIMES..NOW CHECK IF C HAS GONE TO ZERO
 5612  015000  171            CHKSTK: MOV     A,C     ;COPY C TO ACCM TO CHECK IF ZERO
 5613  015001  247                    ANA     A       ;,EQ, 0??
 5614  015002  362  332  031          JP      RPINCB  ;JUMP IF NOT YET..
 5615
```

```
 5617                          ;ELSE FALL THRU..MUST NOW MOVE ASSEMBLED NUMBER TO DESTINATION
 5618  015005  052  207  040   CLEAN:  LHLD    RPBUFS  ;GET BYTE COUNTER FOR THIS DATA
 5619  015010  106                  MOV     B,M     ;NOW B HAS COUNT OF NUMBER BYTES ASSY'D
 5620  015011  043                  INX     H       ;FIX H,L UP AGAIN..
 5621                          ;NOW H,L HAS SRC..D,E HAS FF.. STACK HAS PC
 5622  015012  343                  XTHL            ;SWAP-EEE
 5623  015013  315  050  033        CALL    TARG1   ;ASSEMBLE DEST ADDR INTO D,E
 5624  015016  343                  XTHL            ;SWAP-EEE BACK
 5625
 5626  015017  072  214  040        LDA     CHRCNT  ;GET NUMB CHARS DESIRED BY THIS ROUTINE
 5627  015022  117                  MOV     C,A     ;C NOW HAS DESIRED..B HAS NUMB CHAR ASSY'D
 5628  015023  176            MOVLP:  MOV     A,M     ;START MOVING CHARS TO DESTINATION
 5629  015024  022                  STAX    D       ;CHAR TO DEST..
 5630  015025  043                  INX     H       ;UPDATE SRC POINTER
 5631  015026  023                  INX     D       ;UPDATE DEST POINTER
 5632  015027  015                  DCR     C       ;DOWN THE DESIRED COUNT
 5633  015030  312  052  032        JZ      FIXPNT  ;IF GOT DESIRED NUMBER,DONT PASS ANY MORE
 5634
 5635  015033  005                  DCR     B       ;DOWN THE ASSY'D COUNT
 5636  015034  302  023  032        JNZ     MOVLP   ;KEEP ON KEEPIN' ON
 5637
 5638                          ;WHEN FALL THRU WEVE MOVED ALL THAT WE CAN..NOW PAD THE BUFFER
 5639  015037  170                  MOV     A,B     ;CLR ACCUM..B MUST BE ZERO
 5640  015040  015            PADLP:  DCR     C       ;DOWN THE DESIRED COUNT
 5641  015041  372  056  032        JM      FINARG  ;IF THAT'S MINUS, WERE ALL DONE..
 5642  015044  022                  STAX    D       ;OTHERWISE STACK A ZERO
 5643  015045  023                  INX     D       ;UPDAT DEST POINTER
 5644  015046  303  040  032        JMP     PADLP   ;AND CONTINUE TILL DONE
 5645
 5646  015051  043            FIXLP:  INX     H       ;MUST UPDATE BUFFER POINTER
 5647  015052  005            FIXPNT: DCR     B       ;DOWN THE ASSEMBLED COUNT
 5648  015053  302  051  032        JNZ     FIXLP   ;AND GO BACK TO ADJUST POINTER IF NOT ZERO
 5649
 5650  015056  042  207  040   FINARG: SHLD    RPBUFS  ;NOW PUT BACK OUR LITTLE POINTER
 5651  015061  311                  RET             ;AND GET OUT
```

```
 5653                          ;ROUTINE FOR SHUFFLING BITS FOR A NICE CRAM FORMAT
 5654  015062  176            PLACE:  MOV     A,M     ;GET PIECE OF SRC BYTE
 5655  015063  022                    STAX    D       ;PLACE AT DESTINATION
 5656  015064  043                    INX     H       ;UPDATE SRC POINTER
 5657  015065  023                    INX     D       ;UPDATE DESTINATION POINTER
 5658  015066  176                    MOV     A,M     ;GET UPPER 4 BITS OF 12 BIT CHUNK
 5659  015067  346  017                ANI     ^O17    ;MAKE SURE ONLY 4 BITS WORTH
 5660  015071  022                    STAX    D       ;AND PLACE AT THE DESTINATION
 5661  015072  023                    INX     D       ;DESTINATION UPDATE
 5662  015073  053                    DCX     H       ;BACKUP THE SRC POINTR TO BEGIN OF 24 BITS
 5663  015074  311                    RET             ;AND RETURN
 5664
```

```
 5666                          ;ROUTINE TO COMPLETE THE TRAP HANDLING TYPE OPERATION WHICH CHANGES
 5667                          ;A NORMAL 3 BYTE SUBROUTINE CALL INTO A 2 BYTE TRAP TYPE CALL..
 5668                          ;IT COSTS 3 BYTES TO ADD ANY SUBROUTINE TO THE TRAP CALL,SO THAT YOU SAVE
 5669                          ;AT LEAST ONE BYTE FOR ANY SUBROUTINE THAT IS CALLED 3 TIMES..AND YOU SAVE
 5670                          ;ONE BYTE FOR EACH ADDITIONAL TIME IT IS CALLED
 5671  015075  041  117  032   RTNDIS: LXI     H,DLIST ;GET POINTER TO DISPATCH LIST
 5672  015100  365                   PUSH    PSW     ;SAVE STATE OF PROCESSOR FLAGS
 5673  015101  325                   PUSH    D       ;SAVE 'D,E'..TRAP CANT DESTROY REGS
 5674  015102  205                   ADD     L       ;ADD OFFSET IN ACCUM TO ADDRESS.
 5675  015103  157                   MOV     L,A     ;PUT ADDR PLUS OFFSET BACK
 5676  015104  174                   MOV     A,H     ;GET HI ORDER PIECE
 5677  015105  316  000               ACI     0       ;NOW ADD IN A CARRY IF THERE WAS ONE
 5678  015107  147                   MOV     H,A     ;PUT IT BACK
 5679
 5680  015110  136                   MOV     E,M     ;NOW GO FETCH ADDR TO BE DISPATCHED TO
 5681  015111  043                   INX     H       ;UPDATE TO NEXT
 5682  015112  126                   MOV     D,M     ;NOW FETCH HI ORDER PIECE OF ADDR TO BE DISPATCHED TO
 5683  015113  353                   XCHG            ;GET DISPATCH ADDR INTO H,L
 5684  015114  321                   POP     D       ;RESTORE D,E...NOW ONLY H,L //RET ADDR ON STACK
 5685  015115  361                   POP     PSW     ;RESTORE PROCESSOR FLAGS
 5686  015116  351                   PCHL            ;DISPATCH TO APPROPRIATE SUBROUTINE
 5687
 5688  015117  371  032        DLIST: .ADDR   .MOV5B  ;+0
 5689  015121  202  004               .ADDR   .CRLF   ;+2
 5690  015123  241  031               .ADDR   .ARG16  ;+4
 5691  015125  014  027               .ADDR   .RUN..  ;+6
 5692  015127  234  031               .ADDR   .ARG36  ;+8.
 5693  015131  224  033               .ADDR   .CLRRM  ;+10.
```

```
 5695                           .SBTTL **** CLRB EXECUTE CODE ****
 5696  015133  046  040        CLRBYT: MVI     H,^O40  ;THIS HALF GENERATES THE '20000'S' WEIGHT OF ADDRESS
 5697  015135  157                    MOV     L,A     ;THIS GENERATES THE REST OF THE RAM ADDRESS
 5698  015136  066  000                MVI     M,0     ;CLEAR THAT LOCATION
 5699  015140  341                    POP     H       ;FIX H,L
 5700  015141  311                    RET             ;OUT & DONE
 5701
 5702                           .SBTTL **** SOME ERROR CODES ****
 5703  015142                 RRARG:  PLINE   RAG     ;'?REQUIRES ARG'
  (1)  015142  337                    RST     3       ;PRINT LINE OF CHARS
  (1)  015143  242  037                .ADDR   RAG     ;BUFF TO PRINT
 5704  015145  303  344  003          JMP     NORML   ;ERROR MUST RESET THE STACK
 5705  015150                 KILNM:  PLINE   BB1     ;?BN BAD NUMBER
  (1)  015150  337                    RST     3       ;PRINT LINE OF CHARS
  (1)  015151  246  037                .ADDR   BB1     ;BUFF TO PRINT
 5706  015153  072  251  040  MMERR:  LDA     MMFLG   ;SEE IF IN MAINTENACE MODE
 5707  015156  247                    ANA     A       ;SET 8080 FLAGS
 5708  015157  312  255  001          JZ      REINI   ;IF NO MM MODE, OUT
 5709  015162  315  205  034          CALL    DECNET  ;FINISH UP ANY MESSAGES
 5710  015165  315  147  014  MMERR1: CALL    MMCMD   ;IF YES, RESET MODE
 5711  015170  303  255  001          JMP     REINI   ;ERROR MUST RESET THE STACK
```

```
 5713                           ;ERRORS INCURRED DURING THE BOOT PROCESS
 5714  015173  306  002        D.BTERR: ADI    1*2     ;FAILURE WHEN TRIED TO START MICRO-CODE AFTER A BOOT
 5715
 5716  015175  306  002        C.BTERR: ADI    1*2     ;FAILURE DURING THE READING OF THE MICRO-CODE
 5717
 5718  015177  306  002        B.BTERR: ADI    1*2     ;FAILURE DURING THE READING OF THE PAGE OF POINTERS
 5719
 5720  015201  306  002        A.BTERR: ADI    1*2     ;FAILURE DURING THE READING OF THE HOME BLOCK
 5721
 5722  015203  315  315  032           CALL    LTFLT   ;THESE BOOT ERRORS ARE FATAL
 5723  015206  062  155  040   BTERR1: STA     ERRCD+1 ;THIS BIT OF CODE GOES IN THE HI ORDER BYTE OF THE NUMB
 5724  015211                         CLRB    NOPNT   ;RESTORE PRINTING
 5725  015213                         PLINE   BTFAIL  ;PRINT MESSAGE '?BT '
  (1)  015213  337                    RST     3       ;PRINT LINE OF CHARS
  (1)  015214  133  037               .ADDR   BTFAIL  ;BUFF TO PRINT
 5726  015216  041  154  040         LXI     H,ERRCD ;POINT TO THE ERROR CODE
 5727  015221  315  317  030         CALL    P16     ;PRINT THE 16-BIT NUMBER
 5728  015224                         PCRLF           ;PUT A CRLF AT THE END OF THIS LINE
  (2)  015224  347                    RST     4
  (2)  015225  002                    .BYTE   2
 5729  015226  303  255  001         JMP     REINI   ;KILL THE PROCESS
 5730
 5731                           ;CODE FOR WHEN ONLY BOOTSTRAP FAILS TO READIN
 5732  015231  041  360  040   L.BTERR: LXI    H,STATE ;GET POINTER TO STATE LIGHT
 5733  015234  076  001                 MVI     A,^O01  ;SET FAULT LIGHT, BUT DONT CHANGE STATE
 5734  015236  266                       ORA     M       ;THROW CURRENT STATE WITH FAULT BIT
 5735  015237  167                       MOV     M,A     ;PUT STUFF BACK
 5736  015240  076  020                 MVI     A,8*2   ;FAILURE DURING THE LOADING OF THE PRE-BOOT PROGRAM
 5737  015242  303  206  032           JMP     BTERR1  ;AVOID SOME CODE
```

```
 5739                           ;SUBROUTINE TO CHECK IF A COMMAND FROM THE CSL BOARD HAS BEEN
 5740                           ;GRANTED THE BUS., WHICH IT MUST ALWAYS BE GRANTED BECAUSE IT IS THE
 5741                           ;BUS MASTER
 5742  015245  343             BUSRESP: XTHL           ;GET POINTER TO TRAILING ARG
 5743  015246  333  301                 IN      SMSTS   ;***** I/O RD 301 *****
 5744  015250  057                       CMA             ;FIX INVERSION
 5745  015251  246                       ANA     M       ;"AND" READ STUFF VS. TRAILING ARG
 5746  015252  043                       INX     H       ;UPDATE TO RETURN ADDR
 5747  015253  343                       XTHL            ;SWAP RETURN BACK TO STACK
 5748  015254  311                       RET             ;RETURN.."Z-BIT" CORRESPONDS TO "AND" RESULTS
 5749
 5750                           ;LITTLE ROUTINE TO SET AND OR CLR THE SOFTWARE RUN FLAG
 5751  015255  006  004        SETRN:  MVI     B,STBIT ;WE WANT TO SET THE RUN LIGHT
 5752  015257  315  326  032           CALL    STATEM  ;GO DO IT
 5753  015262  017                     .BYTE   ^O17    ;AND DONT MASH ANYTHING
 5754  015263  257                     XRA     A       ;CLEAR ACCUM
 5755  015264  057                     CMA             ;ACCUM = -1
 5756  015265  062  165  040   RNCOM:  STA     RNFLG   ;DATA TO RUN FLAG
 5757  015270  311                     RET             ;AND OUT
 5758
 5759  015271  006  000        CLRRN:  MVI     B,0     ;WE DONT WANT TO SET ANYTHING
 5760  015273  315  326  032           CALL    STATEM  ;JUST GO AND CLEAR SOME THINGS
 5761  015276  013                     .BYTE   ^O13    ;BITS TO KEEP
 5762  015277  257                     XRA     A       ;CLEAR ACCUM
 5763  015300  303  265  032           JMP     RNCOM   ;AND OUT.
 5764
 5765  015303                  NOREFRESH: CLRB NOPNT   ;TURN TYPING ON
 5766  015305                          CLRB    CHKREF  ;SAY NOT TO REPORT OVER AND OVER
 5767  015307  315  346  006          CALL    CLRUSE  ;EXIT FROM USER MODE
 5768  015312                          PLINE   MOSMSG  ;MESSAGE TO CTY
  (1)  015312  337                    RST     3       ;PRINT LINE OF CHARS
  (1)  015313  173  037               .ADDR   MOSMSG  ;BUFF TO PRINT
 5769
 5770  015315  365             LTFLT:  PUSH    PSW     ;MUST SAVE ACCUM TO GET CORRECT "BT ERR MSG"
 5771  015316  006  001                 MVI     B,1     ;WE MERELY WANT TO SET FAULT LIGHT
 5772  015320  315  326  032           CALL    STATEM  ;GO SET THE LIGHTS
 5773  015323  012                     .BYTE   ^O12    ;BITS TO FLUSH WITH THIS
 5774  015324  361                     POP     PSW     ;RESTORE ACCUM
 5775  015325  311                     RET             ;AND NOW SAFE TO RETURN
```

```
 5777                           ;ROUTINE TO CLEAR AND SET BITS IN THE STATE WORD, THEN TO LIGHT
 5778                           ;THE LIGHTS ON THE FRONT PANEL AS SPECIFIED BY THE STATE WORD
 5779  015326  343              STATEM: XTHL            ;GET POINTER TO MASK
 5780  015327  072  360  040           LDA     STATE   ;NOW FETCH CURRENT STATE OF THE MACHINE
 5781  015332  246                    ANA     M       ;MASK AS SPECIFIED
 5782  015333  043                    INX     H       ;UPDATE RETURN POINTER
 5783  015334  343                    XTHL            ;AND PUT IT BACK ON THE STACK
 5784
 5785  015335  260                    ORA     B       ;NOW THROW IN ANY NEW BITS
 5786  015336  062  360  040           STA     STATE   ;NOW SAVE IT
 5787  015341  323  101              OUT     LIGHTS  ;CHANGE THE LIGHTS
 5788  015343  311                    RET             ;OUT
```

```
 5790                          ;SUBROUTINE TO DECIDE IF 'FIRST' POINTS TO
 5791                          ;AN END-OF-COMMAND CHARACTER,,C-BIT SET IF YES, 'FIRST'
 5792                          ;DOES POINT TO END-OF-COMMAND,,ACCUMULATOR IS DESTROYED
 5793  015344  345             EOCML:  PUSH    H       ;SAVE 'H,L'
 5794  015345  052  223  040           LHLD    .ARG1   ;GET CURRENT POINTER FOR COMMAND BUFFER
 5795
 5796  015350  176                     MOV     A,M     ;GET CHARACTER
 5797  015351  376  377               CPI     EOLCH   ;END-OF-COMMAND??
 5798  015353  312  366  032           JZ      EOLYS   ;JUMP IF YES
 5799
 5800  015356  376  054               CPI     COMMA   ;OR, END-OF-COMMAND??
 5801  015360  312  366  032           JZ      EOLYS   ;JUMP IF YES
 5802
 5803                          ;HERE IF NOT... CLR 'C-BIT' & LEAVE
 5804  015363  247                     ANA     A       ;CLR 'C-BIT'
 5805
 5806  015364  341                     POP     H       ;RESTORE 'H,L'
 5807  015365  311                     RET             ;RETURN
 5808
 5809                          ;HERE IF YES, AT END-OF-COMMAND
 5810  015366  067             EOLYS:  STC             ;SET CARRY
 5811  015367  341                     POP     H       ;RESTORE 'H,L'
 5812  015370  311                     RET             ;RETURN
```

```
 5814                           ;SUBROUTINE TO MOVE 5 CONTIGUOUS BYTES BEGINNING
 5815                           ;WITH A SPECIFIED SOURCE ADDRESS,TO ANOTHER BUFFER AREA, ITS
 5816                           ;ADDRESS ALSO PASSED AS A TRAILING ARG. SOURCE ADDRESS IS
 5817                           ;FIRST TRAILING PARAMETER, DESTINATION IS SECOND TRAILING PARAMETER
 5818  015371  341             .MOV5B: POP     H       ;GET H,L FROM TRAP HANDLER
 5819  015372  343                     XTHL            ;SWAP STACK TOP WITH 'H,L'
 5820  015373  325                     PUSH    D       ;SAVE 'D,E'
 5821  015374  305                     PUSH    B       ;SAVE 'B,C'
 5822  015375  315  044  033          CALL    TARG2   ;ASSEMBLE ARGS INTO 'B,C' AND 'D,E'
 5823
 5824  015400  315  007  033          CALL    MOVREG  ;MOVE THE DATA, ARGS PASSED IN REGISTERS
 5825  015403  301                     POP     B       ;RESTORE 'B,C'
 5826  015404  321                     POP     D       ;RESTORE 'D,E'
 5827
 5828  015405  343                     XTHL            ;RESTORE STACK
 5829  015406  311                     RET             ;AND RETURN
 5830
 5831  015407  076  005        MOVREG: MVI     A,5     ;SET COUNTER TO 5
 5832  015411  075             M5B:    DCR     A       ;DOWN COUNTER
 5833  015412  304  011  033          CNZ     M5B     ; AND BE RECURSIVE TILL DOWN COUNTED
 5834
 5835  015415  012                     LDAX    B       ;BYTE TO ACCUM
 5836  015416  022                     STAX    D       ;STORE AT DESTINATION
 5837  015417  023                     INX     D       ;UP BOTH PNTRS
 5838  015420  003                     INX     B       ; TO NEXT BYTE
 5839  015421  311                     RET             ;AND BACK TO CALLER
 5840
```

```
 5842                           ;SUBROUTINE TO COMPARE 2 36-BIT VALUES.
 5843                           ;IF THE ADDRESSES OF THE 2 36-BIT BUFFERS ARE
 5844                           ;PASSED AS TRAILING PARAMETERS TO THE ROUTINE
 5845                           ;IF BOTH BUFFERS ARE THE SAME, THE 'C-BIT' IS
 5846                           ;CLR UPON RETURN.  IF THEY ARE DIFFER, THE 'C-BIT' IS
 5847                           ;SET ON RETURN.
 5848  015422  343              CMP36:  XTHL            ;SWAP STACK TOP WITH 'H,L'
 5849  015423  315  044  033            CALL    TARG2   ;GET THE 2 TRAILING ARGS INTO 'B,C' & 'D,E'
 5850  015426  343                      XTHL            ;PUT RETURN BACK ON STACK
 5851  015427  353                      XCHG            ;SWAP 'D,E' & 'H,L'
 5852  015430  026  005                 MVI     D,5     ;SET COUNTER TO 4.
 5853
 5854  015432  012              CMPLP:  LDAX    B       ;GET A BYTE OF DATA
 5855  015433  276                      CMP     M       ;COMPARE
 5856  015434  300                      RNZ             ;RETURN WITH Z-CLR IF HAD ERR..
 5857  015435  003                      INX     B       ;BUMP POINTER
 5858  015436  043                      INX     H       ;BUMP OTHER POINTER
 5859  015437  025                      DCR     D       ;DOWN COUNT
 5860  015440  302  032  033            JNZ     CMPLP   ;CONTINUE TILL DONE
 5861  015443  311                      RET             ;NORMAL RETURN
 5862
```

```
                        ;SUBROUTINE TO ASSEMBLE TRAILING ARGS INTO REGISTER PAIRS.
                        ;ROUTINE USED TO SAVE CORE ONLY BECAUSE THIS SEQUENCE OF CODING
                        ;IS REPEATED SO OFTEN.,'H,L' POINTS TO THE TRAILING ARG OF THE
                        ;ORIGINAL CALLER.,'D,E' AND 'B,C' MUST HAVE BEEN SAVED BEFORE
                        ;THIS ROUTINE IS CALLED OR THEY WILL BE DESTROYED.,IF A SINGLE
                        ;TRAILING ARG IS TO BE GATHERED UP, IT WILL BE PUT INTO THE
                        ;REG PAIR 'D,E' VIA THE CALL 'TARG1'.,IF 2 TRAILING ARGS TO BE
                        ;GATHERED UP, THE FIRST WILL BE PUT INTO 'B,C' AND THE SECOND
                        ;WILL BE PUT INTO 'D,E'.,'H,L' IS UPDATED TO POINT TO THE BYTE
                        ;FOLLOWING THE TRAILING ARGS.,
015444  116             TARG2:  MOV     C,M     ;LO ORDER SOURCE TO 'C'
015445  043                     INX     H
015446  106                     MOV     B,M     ;HI ORDER SOURCE TO 'B'
015447  043                     INX     H

015450  136             TARG1:  MOV     E,M     ;LO ORDER SOURCE TO 'E'
015451  043                     INX     H
015452  126                     MOV     D,M     ;HI ORDER SOURCE TO 'D'
015453  043                     INX     H

015454  311                     RET             ;AND RETURN
                        ;SUBROUTINE TO ADD 1 TO A 36-BIT BUFFER AND GUARANTEE
                        ;THAT THE CARRY PROPAGATES CORRECTLY. BUFFER TO BE
                        ;INCREMENTED IS PASSED AS A TRAILING ARG.
015455  343             INC36:  XTHL            ;GET POINTER TO TRAILING ARG
015456  315  050  033            CALL    TARG1   ;ASSEMBLE ARG INTO 'D,E'

015461  343                     XTHL            ;PUT RETURN BACK ON THE STACK
015462  353                     XCHG            ;NOW H,L PNTS TO BUFFER TO BE INCREMENTED

015463  257                     XRA     A       ;CLR THE ACCUM
015464  067                     STC             ; AND SET 'C-BIT'

015465  216             INCLP:  ADC     M       ;ADD PIECE OF DATA BUFF, WITH CRY
015466  167                     MOV     M,A     ;AND PUT IT BACK, WITH THE ADDITION
015467  320                     RNC             ;RETURN IF FINALLY STOPPED CRY'S INTO NEXT BYTE
015470  043                     INX     H       ;NEXT PIECE TO INC
015471  303  065  033            JMP     INCLP   ;AND CONTINUE IF THERE WAS A CRY
```

```
 5904                           ;SUBROUTINE 'RDATT'
 5905
 5906                           ;ROUTINE READS I/O REGISTERS 0,1,2,3,103 AND MOVES THE
 5907                           ;DATA IN THOSE BUFFERS (BUS,BITS 0-35) INTO A RAM AREA
 5908                           ;WHOSE ADDRESS IS SPECIFIED BY A TRAILING PARAMETER
 5909                           ;USED WITH THE CALL TO THIS ROUTINE
 5910                           ;CALL IS:
 5911                           ;       CALL RDATT
 5912                           ;       DW XXX  ;XXX IS PLACE TO MOVE THE 36 BITS OF DATA
 5913                           ;ACCUMULATOR IS DESTROYED, REG PAIR 'D,E' IS INCREMENTED BY 5.
 5914  015474                   RDATT:
 5915  015474  343                      XTHL            ;SWAP STACK TOP WITH 'H,L'
 5916  015475  315  050  033           CALL    TARG1   ;ASSEMBLE TRAILING ARG INTO 'D,E'
 5917  015500  343                      XTHL            ;PUT BACK THE STACK
 5918
 5919                           ;THE REAL READING CODE BEGINS HERE & ALSO SERVES AS AN
 5920                           ;ALTERNATE ENTRY IF YOU CHOOSE TO PASS THE BUFFER ADDRESS
 5921                           ;IN REGISTER 'D,E'
 5922  015501                   RDATP:
 5923  015501  325                      PUSH    D       ;SAVE 'D,E'
 5924  015502  333  000                 IN      D2835   ;***** I/O RD '0' (BITS 28-35) *****
 5925  015504  057                      CMA
 5926  015505  022                      STAX    D       ;SAVE IN RAM
 5927  015506  023                      INX     D       ;UP PNTR TO NEXT BYTE
 5928  015507  333  001                 IN      D2027   ;***** I/O RD '1' (BITS 20-27) *****
 5929  015511  057                      CMA
 5930  015512  022                      STAX    D       ;SAVE IN RAM
 5931  015513  023                      INX     D       ;UP PNTR TO NEXT BYTE
 5932  015514  333  002                 IN      D1219   ;*****I/O RD '2' (BITS 12-19) *****
 5933  015516  057                      CMA
 5934  015517  022                      STAX    D       ;SAVE IN RAM
 5935  015520  023                      INX     D       ;UP PNTR
 5936  015521  333  003                 IN      D0411   ;***** I/O RD '3' (BITS 4-11) *****
 5937  015523  057                      CMA
 5938  015524  022                      STAX    D       ;SAVE
 5939  015525  023                      INX     D       ;UP PNTR
 5940  015526  333  103                 IN      D0003   ;***** I/O RD '103' (BITS 0-03) *****
 5941  015530  057                      CMA
 5942  015531  346  017                 ANI     ^O17    ;OFF TRASH IN D BITS 7-4
 5943  015533  022                      STAX    D       ;SAVE
 5944  015534  321                      POP     D       ;RESTORE 'D,E'
 5945  015535  311                      RET             ;RETURN
```

```
 5947                           ;SUBROUTINE "WDATT"
 5948                           ;ROUTINE WRITES I/O REGISTERS 102,104,106,110,112 AND GETS ADDR
 5949                           ;EITHER PASSED AS A TRAILING PARAMETER, OR PASSED IN "D,E"
 5950                           ;CALL IS:
 5951                           ;       CALL WDATT
 5952                           ;       DW      XXX     ;XXX IS SOURCE OF DATA TO BE WRITTEN
 5953
 5954  015536  343              WDATT:  XTHL            ;SWAP STACK TOP WITH "H,L"
 5955  015537  315  050  033            CALL    TARG1   ;ASSEMBLE TRAILING ARG INTO "D,E"
 5956  015542  343                    XTHL            ;SWAP STACK BACK TO ORIGINAL STATE
 5957                           ;ALTERNATE ENTRY FOR WHEN PASSING DATA POINTER IN "D,E"
 5958
 5959  015543  325              WDATP:  PUSH    D       ;SAVE "D,E"
 5960  015544  032                    LDAX    D       ;DATA 28-35 TO ACCUM
 5961  015545  323  102                OUT     W2835   ;***** I/O WRT "102" (BITS 28-35) *****
 5962  015547  023                    INX     D       ;NEXT DATUM
 5963
 5964  015550  032                    LDAX    D       ;DATA 20-27 TO ACCUM
 5965  015551  323  104                OUT     W2027   ;***** I/O WRT "104" (BITS 20-27) *****
 5966  015553  023                    INX     D       ;NEXT DATUM
 5967
 5968  015554  032                    LDAX    D       ;DATA 12-19 TO ACCUM
 5969  015555  323  106                OUT     W1219   ;***** I/O WRT "106" (BITS 12-19) *****
 5970  015557  023                    INX     D       ;NEXT DATUM
 5971
 5972  015560  032                    LDAX    D       ;DATA 4-11 TO ACCUM
 5973  015561  323  110                OUT     W0411   ;***** I/O WRT "110" (BITS 04-11) *****
 5974  015563  023                    INX     D       ;NEXT DATUM
 5975
 5976  015564  032                    LDAX    D       ;DATA 0-3 TO ACCUM
 5977  015565  323  112                OUT     W0003   ;***** I/O WRT "112" (BITS 00-03) *****
 5978  015567  321                    POP     D       ;RESTORE "D,E"
 5979  015570  311                    RET             ;RETURN
```

```
 5981                           ;SUBROUTINE 'ADATT'
 5982                           ;ROUTINE WRITES I/O REGISTERS 103,105,107,111,113 AND GETS ADDR
 5983                           ;EITHER PASSED AS A TRAILING PARAMETER, OR PASSED IN 'D,E'
 5984                           ;CALL IS:
 5985                           ;       CALL ADATT
 5986                           ;       DW      XXX     ;XXX IS SOURCE OF DATA TO BE WRITTEN
 5987
 5988  015571  343              ADATT:  XTHL            ;SWAP STACK TOP WITH 'H,L'
 5989  015572  315  050  033            CALL    TARG1   ;ASSEMBLE TRAILING ARG INTO 'D,E'
 5990  015575  343                    XTHL            ;SWAP STACK BACK TO ORIGINAL STATE
 5991                           ;ALTERNATE ENTRY FOR WHEN PASSING DATA POINTER IN 'D,E'
 5992
 5993  015576  325              ADATP:  PUSH    D       ;SAVE 'D,E'
 5994  015577  032                    LDAX    D       ;DATA 28-35 TO ACCUM
 5995  015600  323  103                OUT     A2835   ;***** I/O WRT '103' (BITS 28-35) *****
 5996  015602  023                    INX     D       ;NEXT DATUM
 5997
 5998  015603  032                    LDAX    D       ;DATA 20-27 TO ACCUM
 5999  015604  323  105                OUT     A2027   ;***** I/O WRT '105' (BITS 20-27) *****
 6000  015606  023                    INX     D       ;NEXT DATUM
 6001
 6002  015607  032                    LDAX    D       ;DATA 12-19 TO ACCUM
 6003  015610  323  107                OUT     A1219   ;***** I/O WRT '107' (BITS 12-19) *****
 6004  015612  023                    INX     D       ;NEXT DATUM
 6005
 6006  015613  032                    LDAX    D       ;DATA 4-11 TO ACCUM
 6007  015614  323  111                OUT     A0411   ;***** I/O WRT '111' (BITS 04-11) *****
 6008  015616  023                    INX     D       ;NEXT DATUM
 6009
 6010  015617  032                    LDAX    D       ;DATA 0-3 TO ACCUM
 6011  015620  323  113                OUT     A0003   ;***** I/O WRT '113' (BITS 00-03) *****
 6012  015622  321                    POP     D       ;RESTORE 'D,E'
 6013  015623  311                    RET             ;RETURN
```

```
 6015                           ;LOCAL SUBROUTINE TO CLR
 6016  015624  341              .CLRRM: POP     H       ;FIX REG AS MESSED UP BY RST INSTR
 6017  015625  343                      XTHL            ;POINTER TO THE TRAILING PARAM
 6018  015626  315  050  033            CALL    TARG1   ;ASSY ARG INTO 'D,E'
 6019  015631  343                      XTHL            ;FIX 'H,L' AND REPLACE FOR RETURN
 6020  015632  353                      XCHG            ;PUT 'D,E' STUFF INTO 'H,L'
 6021  015633  076  005                 MVI     A,5     ;AND SET STARTING COUNT TO 5
 6022  015635  053              CLRT1:  DCX     H       ;DOWN THE MEM ADDRESS
 6023  015636  066  000                 MVI     M,0     ;0 DATA TO MEM
 6024  015640  075                      DCR     A       ;DOWN THE COUNTER
 6025  015641  302  235  033            JNZ     CLRT1   ;BACK TILL DONE
 6026  015644  311                      RET             ;RETURN
 6027
```

```
 6029                           ;SUBROUTINE TO SWALLOW SEPARATOR CHARACTERS FROM THE ADDRESS POINTED
 6030                           ;TO BY 'H,L', UP TO THE FIRST NON-SEPARATOR CHARACTER.
 6031                           ;SEPARATORS ARE:
 6032                           ;       'SPACE'
 6033                           ;       'TAB'
 6034                           ;ONLY THE 'H,L' REGISTER SHOULD BE CHANGED BY THIS ROUTINE
 6035
 6036  015645  365              SEPCHR: PUSH    PSW     ;SAVE ACCUM AND STATUS
 6037  015646  053                      DCX     H       ;DOWN COUNT H,L SO NEXT INSTR WILL MAKE IT EVEN
 6038  015647  043              SEPYS:  INX     H       ;UP THE COUNT
 6039  015650  176                      MOV     A,M     ;COPY CHARACTER INTO ACCUM
 6040  015651  376  040                 CPI      ' '    ;IS THE CHAR A 'SPACE'
 6041  015653  312  247  033           JZ      SEPYS   ;GO UPDATE 'H,L' IF YES..
 6042
 6043                           ;ELSE SEE IF ITS A TAB
 6044  015656  376  011                 CPI     '       ;IS THE CHAR A 'TAB"
 6045  015660  312  247  033           JZ      SEPYS   ;GO UPDATE 'H,L' IF YES
 6046
 6047                           ;ELSE NO MORE SEPARATORS-TIME TO RETURN
 6048  015663  361                      POP     PSW     ;RESTORE ACCUM AND STATUS
 6049  015664  311                      RET             ;ALL DONE RETURN
 6050
 6051                           .SBTTL  SUBROUTINE TIME DELAY
 6052                           ;EACH UNIT OF DELAY COUNTED IN THE TRAILING BYTE IS WORTH 1.02 MICRO-SEC
 6053                           ;THIS SUBROUTINE WASTES SOME AMOUNT OF TIME..THE GREATER THE TRAILING
 6054                           ;ARGUEMENT, THE MORE TIME IS WASTED...
 6055  015665  343              DELAY.: XTHL            ;GET POINTER TO TRAILING ARG INTO 'H,L'
 6056  015666  365                      PUSH    PSW     ;NOW SAVE ACCUM
 6057  015667  176                      MOV     A,M     ;GET THE TRAILING ARG INTO ACCUM
 6058  015670  043                      INX     H       ;UP DATE TO CORRECT RETURN LOCATION
 6059  015671  075              DLYLP:  DCR     A       ;DOWN THE COUNTER
 6060  015672  365                      PUSH    PSW     ;ADD MORE DELAY IN THE LOOP
 6061  015673  361                      POP     PSW     ; BECAUSE PUSHES AND POPS TAKE LONG TIME
 6062  015674  302  271  033           JNZ     DLYLP   ;LOOP TILL ZERO
 6063  015677  361                      POP     PSW     ;RESTORE ACCUM
 6064  015700  343                      XTHL            ;PUT RETURN BACK ONTO THE STACK
 6065  015701  311                      RET             ;AND DONE
```

```
 6067                           .SBTTL  STRING COMPARE ROUTINE
 6068                           ;ROUTINE TO COMPARE A TYPED IN ASCII STRING VERSUS SOME EXPECTED
 6069                           ;STRING. ENTER WITH 'H,L' POINTING TO THE BEGINNING OF THE TYPE-IN
 6070                           ;BUFFER AND WITH D,E POINTING TO THE EXPECTED STRING.
 6071                           ;RETURN Z-BIT CLR IF NO MATCH...Z-BIT SET IF MATCH
 6072  015702  032              STRCMP: LDAX    D       ;GET FIRST EXPECTED CHARACTER
 6073  015703  247                      ANA     A       ;SET FLAGS TO SEE IF ZERO BYTE
 6074  015704  312  316  033           JZ      STREND  ;IF ZERO BYTE, END OF EXPECTED STRING.. OUT
 6075
 6076  015707  276                      CMP     M       ;IF A REAL BYTE, COMPARE AGAINST THE TYPE-IN
 6077  015710  300                      RNZ             ;IF NO MATCH, TAKE ERROR RETURN
 6078  015711  023                      INX     D       ;IF MATCH , UPDATE TO NEXT EXPECTED
 6079  015712  043                      INX     H       ;AND UPDATE TO NEXT TYPED IN.
 6080  015713  303  302  033           JMP     STRCMP  ;LOOP
 6081
 6082  015716  042  223  040  STREND: SHLD    .ARG1   ;PASS CURRENT POINTER TO ROUTINE THAT CHECKS FOR EOL
 6083  015721  315  344  032           CALL    EOCML   ;CHECK THAT TYPE IN WAS TERMINATED
 6084  015724  330                      RC              ;IF YES, Z-BIT IS SET,,, OK TO RETURN
 6085
 6086  015725  264                      ORA     H       ;CLR Z-BIT FLAG.. H WILL BE NON-ZERO
 6087  015726  311                      RET             ;AND OUT.....
```

```
 6089                           ;ROUTINE CALLED WHENEVER KLINIK SWITCH CHANGES STATE
 6090                           ;THE ROUTINE EXAMINES THE NEW STATE OF KLINIK,ZAPS THE LIGHTS AS REQUIRED
 6091                           ;THEN SETS THE KLINIK LINE INTO THE APPROPRIATE STATE. IF KLINIK WAS
 6092                           ;ESTABLISHED, GOING TO ENABLE POSITION WILL CHANGE NOTHING, BUT ANY SWITCH
 6093                           ;CHANGE THAT INCREASES THE AMOUNT OF PROTECTION WILL FORCE CHANGE THE
 6094                           ;MODE OF THE KLINIK LINE.
 6095                           ;THE ROUTINE IS ENTERED WITH 'B' HOLDING THE NEW KLINIK SWITCH STATE, AND
 6096                           ;'KLNKSW' HOLDING THE OLD STATE. VALUES ARE AS FOLLOWS:
 6097                           ;       ENABLE = 2
 6098                           ;       PROTECT = 6
 6099                           ;       DISABLE = 4
 6100  015727  072  360  040   KLNKLT: LDA     STATE   ;MUST GUARANTEE DTR HAS BEEN RESTORED
 6101  015732  323  101                OUT     DTR     ;DO IT
 6102  015734  170                      MOV     A,B     ;COPY KLINIK STATE INTO THE ACCUM
 6103  015735  062  124  040           STA     KLNKSW  ;SAVE THE NEW STATUS
 6104  015740  376  004                CPI     4       ;IS SWITCH NOW IN DISABLED POSITION
 6105  015742  312  016  034           JZ      SETM0   ;GO SET MODE 0 IF YES
 6106
 6107  015745  376  006                CPI     6       ;IS SWITCH NOW IN THE PROTECT POSITION?
 6108  015747  312  371  033           JZ      .SETM1  ;IF YES, GO SET MODE 1
 6109
 6110                           ;FALL THRU IF NEW SWITCH POSITION IS THE 'ENABLE' POSITION
 6111                           ;.. FIRST CHECK CURRENT MODE. IF IN MODE 3 ALREADY, WE MAKE NO CHANGE
 6112  015752  072  247  040           LDA     CSLMODE ;GET CURRENT CSL MODE
 6113  015755  376  010                CPI     .MODE3  ;IS IT MODE 3
 6114
 6115                           ;FLAGS ARE SET, FALL INTO CODE THAT DOES THE RIGHT THING IF IN MODE 3
 6116
 6117  015757  304  101  034           CNZ     SETM2   ;IF WAS NOT MODE 3, THIS WILL SET MODE 2
 6118
 6119                           ;AND FALL INTO KL.LON CODE
 6120
 6121  015762  006  002        KL.LON: MVI     B,2     ;GET A BIT FOR SETTING THE REMOTE LIGHT ON
 6122
 6123                           ;AND FALL INTO CODE FOR SETTING THE LIGHTS
 6124  015764  315  326  032   KL.LAMP: CALL   STATEM  ;SET LIGHTS AS SPECIFIED IN B REG
 6125  015767  375                      .BYTE   ^O375   ;KEEP ALL LIGHTS, 'CEPT REMOTE
 6126  015770  311                      RET             ;AND DONE WITH THIS MESS
```

```
 6128                           .SBTTL  UART MODE MODIFICATIONS
 6129                           ;CODE FOR SETTING THE KLINIK LINE INTO MODE 1
 6130  015771  072  262  040    .SETM1: LDA     PASSWORD ;GET CURRENT PASSWORD
 6131  015774  247                    ANA     A       ;SET FLAGS TO SEE IF ANY PASSWORD EXISTS
 6132  015775  312  016  034          JZ      SETM0   ;IF NO PASSWORD, THEN SET INTO MODE 0
 6133  016000  304  006  034          CNZ     SETM1   ;IF PASSWORD EXISTS, SET THINGS INTO MODE 1
 6134  016003  302  362  033          JNZ     KL.LON  ;IF WE WENT MODE 1, THEN MUST TURN ON LIGHT
 6135
 6136                           ;CODE FOR ACTUALLY SETTING THE KLINIK LINE MODE TO 1
 6137  016006  076  002        SETM1: MVI     A,.MODE1 ;GET MODE 1 FLAG
 6138  016010  041  213  010          LXI     H,MODE1 ;GET THE MODE 1 DISPATCH
 6139  016013  303  123  034          JMP     SETM    ;SET UP RAM
 6140
 6141                           ;CODE THAT SETS BOTH MODE 0 AND THE APPROPRIATE LIGHTS
 6142  016016                   SETM0: CLRB    KLLINE.ON  ;DISABLING KLINIK KILLS CTY AVAILABILITY
 6143  016020  006  000                MVI     B,0     ;THE PASS LIGHTS OFF IN REGISTER 'B'
 6144  016022  315  364  033          CALL    KL.LAMP ;AND GO DO THE LIGHTS
 6145  016025  315  134  034          CALL    HANGUP  ;CLEAR KLINIK LINE
 6146  016030  076  001                MVI     A,.MODE0 ;GET THE MODE 0 FLAG
 6147  016032  041  200  010          LXI     H,MODE0 ;GET THE MODE 0 DISPATCH
 6148  016035  303  123  034          JMP     SETM    ;SET UP RAM
 6149
 6150                           ;CODE TO SET US INTO MODE 3
 6151  016040  076  010        SETM3: MVI     A,.MODE3 ;GET MODE 3 FLAG
 6152  016042  041  373  005          LXI     H,MODE3 ;GET THE MODE 3 DISPATCH
 6153  016045  303  123  034          JMP     SETM    ;SET UP RAM
```

```
 6155                           ;CODE TO SET US INTO MODE 4
 6156  016050  072  156  040  SETM4:  LDA     USRMD   ;SEE IF USER, IF WHICH CASE WE WONT DO 'MODE4'
 6157  016053  247                  ANA     A       ;SET 8080 FLAGS
 6158  016054  300                  RNZ             ;AND OUT IF USER MODE
 6159
 6160                           ;ACCUM MUST .EQ. 0 IF FELL TO HERE
 6161  016055  062  252  040        STA     MAILFG  ;BETTER CLEAR THIS FLAG TOO
 6162  016060  062  301  040        STA     E.CNT   ;USE FASTEST WAY TO CLEAR THIS LOCATION
 6163  016063  041  250  041        LXI     H,E.BEG-1 ;AND RESET ENVELOPER
 6164  016066  042  247  041        SHLD    E.BUF
 6165  016071  076  020             MVI     A,.MODE4 ;GET MODE 4 FLAG
 6166  016073  041  007  035        LXI     H,MODE4 ;GET THE MODE 4 DISPATCH
 6167  016076  303  123  034        JMP     SETM    ;SET UP RAM
 6168
 6169                           ;SET LINE TO MODE 2
 6170  016101  072  247  040  SETM2:  LDA     CSLMODE ;BEFORE ANYTHING ELSE, SEE WHAT WE ARE DOING NOW
 6171  016104  346  003             ANI     .MODE0!.MODE1   ;IF MODES 0 OR 1, MUST INTERRUPT KS10
 6172  016106  312  116  034        JZ      SETM2X  ;IF NOT, DONT BOTHER KS10 AT ALL
 6173
 6174  016111  076  002             MVI     A,KL.ACTIVE ;MUST INFORM THE TEN THAT WE ARE ENTERING KLINIK
 6175  016113  315  156  034        CALL    WRD34   ;CALL ROUTINE THAT WRITES WORD 34
 6176  016116  041  004  011  SETM2X: LXI     H,MODE2 ;GET DISPATCH FOR MODE 2
 6177  016121  076  004             MVI     A,.MODE2 ;SET MOE TWO TO THE STATE FLAG ALSO
 6178  016123  062  247  040  SETM:   STA     CSLMODE
 6179  016126  042  346  040  SETDIS: SHLD    MODDIS  ;AND SET TO KLINIK DISPATCHER
 6180  016131  311                  RET             ;AND ALL DONE
```

```
 6182                           ;LITTLE ROUTINE TO HANG UP THE KLINIK LINE
 6183  016132                   KILL.KLINIK: CLRB KLNKSW  ;FORCE A RELOOK AT THE REMOTE SWITCH
 6184  016134  072  360  040    HANGUP: LDA     STATE   ;GET CURRENT STATE
 6185  016137  346  007                ANI     ^07     ;OFF THE 'DTR' SIGNAL
 6186  016141  323  101                OUT     DTR     ;CLR DTR
 6187  016143  076  003                MVI     A,CARRLOSS ;TELL KS10 THAT KLINIK CARRIER HAS GONE AWAY
 6188  016145  315  156  034          CALL    WRD34   ;DEPOSIT INTO WORD 34
 6189  016150  041  220  001          LXI     H,200. * 2 ;SET A TIMEING DELAY OF 2 SECONDS
 6190  016153  303  134  014          JMP     LTLOOP  ;GO DO DELAY, AND USE HIS RETURN TO EXIT
 6191
 6192                           ;ROUTINE FOR DOING SIMPLE DEPOSIT INTO KS10 MEMORY AT WORD 34, AND
 6193                           ;THEN INTERRUPTING THE 10
 6194  016156  365              WRD34:  PUSH    PSW     ;SAVE ACCUM & STATUS
 6195  016157                           CLRRM   DMDAT   ;CLEAR A BUFFER
  (1)  016157  347                      RST     4
  (1)  016160  012                      .BYTE   10.
  (1)  016161  054  040                 .ADDR   DMDAT+5
 6196  016163  361                      POP     PSW     ;FETCH THE ACCUM'S CONTENTS AGAIN
 6197  016164  043                      INX     H       ;BUMP H,L(VALUE AFTER A CLRRM IS .EQ. 1ST LOC OF BUFF)
 6198  016165  167                      MOV     M,A     ;STORE DATA AT 'DMDAT+1'
 6199  016166                           DEPOS   34      ;DEPOSIT
  (1)  016166  247                      ANA     A       ;CLR 'C-BIT' FOR USE BY COMMON CODE
  (1)  016167  315  217  030            CALL    DEPSHT  ;AND DO THE DEPOSIT ASSUMING SHORT ADDR
  (1)  016172  034  000                 .ADDR   34      ;ADDR TO BE ZAPPED PASSED AS TRAILING ARG
 6200  016174  303  057  010          JMP     POKE10  ;INTERRUPT THE KS10 & USE HIS RETURN
```

```
 6202                           ;CODE USED IN ADDING UP THE CHECKSUMS ON ENVELOPES TO BE SENT
 6203  016177  200              CHKADD: ADD     B       ;HERE TO ADD NEW CHAR TO THE CURRENT SUM
 6204  016200  107                      MOV     B,A     ;AND KEEP THE RESULTS IN 'B'
 6205  016201  043                      INX     H       ;BUMP UP TO LOOK AT THE NEXT CHAR
 6206  016202  303  231  034            JMP     TSKLP   ;BACK TO LOOP
 6207
 6208                           ;THIS IS THE APT ENVELOPE SENDER.. WHEN WE HAVE A BUFFER OF INFO TO SEND TO
 6209                           ;THE APT HOST SYSTEM, THIS IS THE CODE THAT GETS CALLED
 6210  016205  072  252  040    DECNET: LDA     MAILFG  ;ONLY DO SOMETHING HERE IF THE MAILING FLAG SET
 6211  016210  247                      ANA     A       ;SET 8080 FLAGS
 6212  016211  310                      RZ              ;NO FLAG, NO SENDY....
 6213
 6214  016212  373                      EI              ;ABSOLUTLY MUST ALLOW INTERRUPTS, IN CASE HOST DIES
 6215  016213  072  011  042            LDA     ENVMNO  ;FIRST THING TO DO IS COMPLIMENT THE MESSAGE NUMBER
 6216  016216  057                      CMA             ;FLIP
 6217  016217  346  177                ANI     ^O177   ;NO SIGN BITS ALLOWED
 6218  016221  062  011  042            STA     ENVMNO  ;PUT IT BACK
 6219
 6220  016224  041  013  042            LXI     H,ENVBUF ;FIRST THING TO DO IS COMPUTE CHECKSUM FOR THE ENVELOPE
 6221  016227  006  000                MVI     B,0     ;'B' WILL HOLD THE CURRENT SUM
 6222  016231  176              TSKLP:  MOV     A,M     ;GRAB A CHARACTER
 6223  016232  376  015                CPI     CRCHR   ;SEE IF END OF THE ENVELOPE CHARACTER
 6224  016234  312  243  034            JZ      TSKGO   ;IF YES, GO TO THE ACTUAL SENDER
 6225
 6226  016237  247                      ANA     A       ;MAYBE THE CHAR WAS A 0, BECAUSE THERE IS NO CRCHR
 6227  016240  302  177  034            JNZ     CHKADD  ;IF NOT, GO ADD THE CHARACTER TO THE SUM
```

```
 6229                                ;HERE WHEN TIME TO ACTUALLY MAIL AN ENVELOPE
 6230  016243  043            TSKGO:  INX     H       ;UPDATE PAST THE "CR" CHARACTER
 6231  016244  066  000               MVI     M,0     ;NOW GUARANTEE THAT WE END WITH "CR","0" PAIR
 6232  016246  170                    MOV     A,B     ;GRAB THE CURRENT SUM
 6233  016247  057                    CMA             ;COMPLIMENT
 6234  016250  074                    INR     A       ;MAKE TWOS COMPLIMENT
 6235  016251  346  077               ANI     ^O77    ;AND ONLY SIX BITS COUNT
 6236
 6237                                ;NOW MUST DECIDE IF YOU NEED TO ASCII-IZE THE CHECKSUM
 6238  016253  376  075               CPI     ^O75    ;75,76,77 DONT GET ASCII-ED
 6239  016255  362  262  034         JP      TSKGO1  ;SO JUMP IF ANY OF THOSE THREE
 6240  016260  366  100               ORI     ^O100   ;HAD TO ASCII-IZE, SO DO IT WITH A 100
 6241  016262  062  012  042  TSKGO1: STA     ENVCHK  ;SAVE IN THE APPROPRIATE PLACE IN THE BUFFER
 6242
 6243  016265                TSK2TSK:        CLRB    APTANS  ;CLEAR THE ANSWER
 6244  016267                        KCHAR   SYNC    ;2 SYNCS START EVERY MESSAGE
  (1)  016267  315  117  004         CALL    KCHR    ;GO PRINT THE CHARACTER
  (1)  016272  001                   .BYTE   SYNC
 6245  016273                        KCHAR   SYNC
  (1)  016273  315  117  004         CALL    KCHR    ;GO PRINT THE CHARACTER
  (1)  016276  001                   .BYTE   SYNC
 6246
 6247  016277  021  011  042         LXI     D,ENVMNO ;NOW SEND THE REST
 6248  016302  315  144  004         CALL    KLINE1
 6249  016305  072  013  042         LDA     ENVBUF  ;GRAB FIRST CHAR OF ENVELOPE JUST SENT
 6250  016310  376  077               CPI     QUES    ;IS IT QUESTION MARK??
 6251  016312  312  165  032         JZ      MMERR1  ;IF IT WAS, ABORT ENVELOPE STUFF, RESET APT
 6252
 6253  016315  376  045               CPI     PERCNT  ;IS IT A PER CENT SIGN??
 6254  016317  312  165  032         JZ      MMERR1  ;IF IT WAS, ABORT ENVELOPE STUFF, RESET APT
 6255
 6256  016322  072  261  040  APT.WT: LDA     APTANS  ;NOW WAIT FOR THE APT SYS TO ANSWER(ACK OR NACK)
 6257  016325  247                    ANA     A       ;IF ZERO, GOT NO ANSWER YET
 6258  016326  312  322  034         JZ      APT.WT  ;SO WAIT
```

```
 6260                          ;FINALLY GOT AN ANSWER
 6261  016331  376  116              CPI     'N      ;WAS IT A NACK??
 6262  016333  312  265  034         JZ      TSK2TSK ;IF YES, SEND IT OUT AGAIN
 6263  016336                  DECEX1:
 6264  016336                  DECEX2:
 6265  016336  257                    XRA     A       ;USE FAST WAY TO CLEAR A RAM LOCATION
 6266  016337  062  252  040         STA     MAILFG  ;SAY END OF THIS ENVELOPE
 6267  016342  041  013  042         LXI     H,ENVBUF ;POINT TO THE BUFFER
 6268  016345  042  352  040         SHLD    ENVPNT  ;SAVE THE POINTER TO THE BUFFER
 6269  016350  311                    RET             ;THEN OUT
```

```
 6271                           ;SUBROUTINE TO MOVE A STRING OF CHARACTERS INTO THE TTY INPUT BUFFER,
 6272                           ;KEEPING TRACK OF THE NUMBER OF COMMAS AND OTHER IMPORTANT FEATURES OF THE
 6273                           ;STRING. MUST PASS THE SOURCE OF THE CHARACTERS IN REG B,C, SUBROUTINE WILL
 6274                           ;BOMB REGISTERS D,E AND H,L
 6275  016351  001  253  041   MV.ALL: LXI     B,E.BEG+2 ;POINT TO THE CHARACTER BUFFER TO BE EXECUTED
 6276  016354  315  160  010           CALL    BFRST   ;RESET CMD CHAIN POINTERS
 6277  016357  021  114  041   MV.INP: LXI     D,BUFBG ;DE, WILL POINT TO THE INPUT BUFFER
 6278  016362  041  204  040           LXI     H,EOL   ;AND HL WILL POINT TO THE COMMA/EOL COUNTER
 6279  016365  066  000               MVI     M,0     ;MAKE SURE COUNT BEGINS AT 0
 6280
 6281  016367  012               MV.IN1: LDAX    B       ;GET FIRST CHARACTER FROM WHEREVER IT IS
 6282  016370  022                       STAX    D       ;AND PUT IT INTO THE BUFFER
 6283  016371  003                       INX     B       ;UP POINTER
 6284  016372  023                       INX     D       ;AND THIS ONE TOO
 6285
 6286                           ;NOW CHECK FOR COMMA OR EOL
 6287  016373  376  054                   CPI     COMMA   ;IS IT A COMMA??
 6288  016375  314  005  035           CZ      MV.CNT  ;IF YES, INCREMENT THE COUNT
 6289
 6290                           ;FALL THRU IF WAS A COMMA BEFORE, EOLCH WILL NOT MATCH
 6291  016400  376  377                   CPI     EOLCH   ;IS IT AN END OF LINE?
 6292  016402  302  367  034           JNZ     MV.IN1  ;IF NOT, THERE IS MORE TO DO
 6293
 6294                           ;HERE IF WAS AN EOL.. NOT ONLY DO WE BUMP THE COUNT, WE ALSO GET OUT
 6295  016405  064               MV.CNT: INR     M       ;UP COUNT
 6296  016406  311                       RET             ;AND OUT
```

```
 6298                           .SBTTL  MODE 4 FOR DECIDING TO COLLECT AN ENVELOPE OR CTY OUT
 6299                           ;MODE 4 HANDLER. WATCHES FOR THE FIRST SYNC CHAR, THEN GOES INTO
 6300                           ;A FINITE STATE MACHINE MODE WHERE IT COLLECTS AN ENVELOPE
 6301                           ;WHEN YOU ENTER HERE, REG B HAS A COPY OF THE CHARACTER JUST TYPED
 6302  016407                   MODE4:
 6303  016407  376  001         M4.0:   CPI     SYNC    ;LOOK FOR A SYNC CHARACTER
 6304  016411  302  250  035           JNZ     MMOUT   ;IF NOT, SIMPLE PRINT OF CHARACTER ON CTY
 6305
 6306  016414  041  022  035           LXI     H,M4.1  ;SHIFT ENVELOPER TO NEXT INPUT STATE
 6307  016417  303  126  034           JMP     SETDIS  ;AND SET INTERRUPT HANDLER TO COME HERE WHEN
 6308                                                   ;DONE WITH INTERRUPT
 6309
 6310                           ;STATE 2 OF ENVELOPE EATER,, THIS CODE WILL DISCARD ANY ADDITIONAL SYNCS
 6311                           ; STORE THE MESSAGE NUMBER WHEN IT FINALLY GETS HERE (& FLIC TO STATE 3)
 6312                           ; OR COLLECT THE FIRST CHARACTER OF A CMD SEQUENCE(& FLIC TO STATE 3)
 6313  016422                   M4.1:
 6314  016422  376  001                 CPI     SYNC    ;IS THIS AN ADDITIONAL SYNC CHAR
 6315  016424  310                       RZ              ;IF YES, IGNORE AND PROCEED
 6316                           ;FALL THRU IF NOT A SYNC
 6317  016425  041  033  035           LXI     H,COLLECT ;NOW GO TO NEXT STATE OF THE ENVELOPE COLLECTOR
 6318  016430  042  346  040           SHLD    MODDIS  ;SET UP FOR INTERRUPT HANDLER TO FIND
```

```
 6320                           .SBTTL  ENVELOPE COLLECTOR
 6321                           ;THIS IS WHERE YOU COME ON CHARACTERS THAT ARE PART OF AN ENVELOPE.
 6322                           ;THIS CODE CHECKS FOR 2 KINDS OF TERMINATORS
 6323                           ;1) END OF ENVELOPE
 6324                           ;2) END OF CONTROL SEQUENCE
 6325                           ;OR ELSE MERELY STUFFS THE CHARACTER INTO THE ENVELOPE BUFFER.
 6326                           ;WHEN AN ENTIRE MESSAGE HAS BEEN RECEIVED, THEN WE WILL CALCULATE THE
 6327                           ;CHECKSUMS OR WHATEVER, MAKE WITH THE ACKS, NACKS, AND EXECUTE WHATEVER
 6328                           ;THE STUFF MAY BE
 6329                           ;WHEN YOU ENTER HERE, REG B HAS A COPY OF THE CHARACTER JUST TYPED
 6330  016433  376  044        COLLECT: CPI    DOLLAH  ;TOPS20 CALLS A SPADE A 'DOLLAR'
 6331  016435  312  236  035          JZ     ACTION  ;IF '$', TREAT LIKE AN ALTMODE
 6332
 6333  016440  376  033                CPI    ALT     ;IF ALTMODE, THEN END OF CONTROL SEQUENCE
 6334  016442  312  236  035          JZ     ACTION  ;AND JUMP IF IT WAS ALTMODE. EXECUTE CONTROL CHAR
 6335
 6336  016445  376  015                CPI    CRCHR   ;IF CARRIAGE RETURN, THEN END OF ENVELOPE
 6337  016447  312  102  035          JZ     EXECUT  ;GO EXECUTE THE ENVELOPE IF <CR>
 6338
 6339  016452  376  001                CPI    SYNC    ;ALSO LOOK FOR 'SYNC', WHICH MEANS 'RESYNC'
 6340  016454  312  050  034          JZ     SETM4   ;IF YES, THEN MUST RE-SYNC
 6341                           ;WE MUST BE AWARE OF 'RE-SYNCING', IN CASE THE 'ALT' OR 'CR' WAS GARBLED
 6342                           ;AS IT CAME DOWN THE LINE, AND WAS MISSED BY THE 8080
 6343                           ;RE-SYNCING REQUIRES STARTING AT THE BEGINNING OF MODE4
 6344
 6345
 6346                           ;FALL THRU IF MUST SIMPLY SHOVE THE CHARACTER INTO THE BUFFER
 6347  016457  052  247  041    COL.LP: LHLD    E.BUF   ;GET POINTER TO THE LAST CHARACTER IN THE BUFFER
 6348  016462  043                    INX     H       ;BUMP POINTER TO FIRST FREE
 6349  016463  167                    MOV     M,A     ;AND STACK THE CHARACTER IN THE BUFFER
 6350  016464  042  247  041          SHLD    E.BUF   ;REPLACE THE POINTER
 6351
 6352  016467  041  301  040          LXI     H,E.CNT ;GET CURRENT CHARACTER COUNT
 6353  016472  064                    INR     M       ;UPDATE
 6354  016473  176                    MOV     A,M     ;NOW COPY COUNT TO ACCUM FOR TESTING
 6355  016474  376  134                CPI     ^O134   ;TOO MUCH FOR AN ENVELOPE??
 6356  016476  322  220  035          JNC     NACK.EN ;IF TOO MANY, NACK IT.. MAYBE HE WILL START OVER
 6357  016501  311                    RET             ;ELSE OUT
```

```
 6359                           ;HERE WHEN AN ENVELOPE IS COMPLETE,, WE MUST COMPUTE THE CHECKSUM
 6360                           ;AND COMPARE AGAINST THE CHECK CHARACTER SENT OVER, THEN ACTUALLY EXECUTE
 6361                           ;THE CONTENTS OF THE ENVELOPE
 6362  016502  072  301  040   EXECUT: LDA     E,CNT   ;GET CHAR COUNT SO WE CAN TELL WHEN WE FINISH
 6363  016505  117                   MOV     C,A     ;PUT IT IN 'C'
 6364
 6365  016506  041  252  041          LXI     H,E,BEG+1 ;POINT TO THE CHECKSUM IN THE ENVELOPE BUFFER
 6366  016511  176                   MOV     A,M     ;GET CHECKSUM CHARACTER INTO THE ACCUM
 6367  016512  043                   INX     H       ;UPDATE PAST THE CHECKSUM JUST COLLECTED
 6368  016513  015                   DCR     C       ; AND DOWN THE CHAR COUNT FOR THE THINGS WE JUST
 6369  016514  015                   DCR     C       ;  PICKED OUT OF THE LIST
 6370  016515  015                   DCR     C       ;WE WANT LOOP TO END AT -1, INSTEAD OF 0
 6371
 6372  016516  206           ENV.LP: ADD     M       ;ADD CHARACTERS TO CHECKSUM
 6373  016517  043                   INX     H       ;NEXT CHARACTER
 6374  016520  015                   DCR     C       ;BUT FIRST SEE IF DONE YET
 6375  016521  362  116  035          JP      ENV.LP  ;BACK IF NOT
 6376
 6377                           ;WHEN DONE,CHECK THAT CHECKSUM HAS WORKED OUT TO BE ZERO
 6378  016524  346  077              ANI     ^O77    ;ONLY SIX BITS COUNT
 6379  016526  302  220  035          JNZ     NACK.EN ;IF NOT .EQ. 0, THEN CHECKSUM FAILED AND 'NACK'
```

```
 6381                           ;FALL THRU TO HERE IF OK SO FAR
 6382  016531  066  377               MVI     M,EOLCH ;MARK THE END OF THE ENVELOPE WITH EOL MARKER
 6383
 6384                           ;NOW MUST CHECK THE MESSAGE NUMBER FOR OK-NESS
 6385  016533  041  361  040          LXI     H,LSTMSG ;GET POINTER TO MESSAGE NUMBER
 6386  016536  116                    MOV     C,M     ;SAVE IT IN 'C' FOR A LITTLE WHILE
 6387  016537  072  251  041          LDA     E.BEG   ;GRAB CURRENT MESSAGE NUMBER
 6388
 6389  016542  271                    CMP     C       ;ARE THEY THE SAME??
 6390  016543  312  202  035          JZ      ACK.EN  ;IF YES, DO SIMPLE ACK AND IGNORE MESSAGE
 6391
 6392                           ;IF DIFF, TWAS A GOOD MESSAGE, SAVE NUMBER AND EXECUTE
 6393  016546  167                    MOV     M,A     ;SAVE MESSAGE NUMBER AS THE LAST
 6394
 6395  016547  315  351  034          CALL    MV.ALL  ;AND MOVE THE STUFF TO A BUFFER FOR EXECUTION
 6396  016552  076  041               MVI     A,^O41  ;EVERY COMMAND ENVELOPE EXECUTED RESETS THE ENV NUMBER
 6397  016554  062  011  042          STA     ENVMNO  ; SO RESET THE ENVELOPE MESSAGE NUMBER TO 41
 6398  016557  315  050  034          CALL    SETM4   ;MODE 4 TO GRAB INTERRUPTS CORRECTLY WHILE RUNNING
 6399  016562  315  336  034          CALL    DECEX1  ;BEFORE EXECUTING, CLEAR ALL OLD MESSAGES
 6400  016565  041  176  035          LXI     H,OKDN ;TELL NORMAL ENDS TO RETURN HERE FOR FURTHER ORDERS
 6401  016570  042  350  040          SHLD    NOREND  ;PASS INFO IN THE DEDICATED RAM POSITION
 6402  016573  303  222  002          JMP     DCODE   ;AND BEGIN EXECUTION OF THE STRING READ IN
```

```
 6404  016576  373            OKIN:   EI              ;MUST ALLOW INTERRUPTS HERE
 6405  016577  315  205  034          CALL    DECNET  ;IF YES, MAIL ENVELOPE BEFORE ACK'ING
 6406
 6407                         ;NOW OK TO ACKNOWLEDGE THE COMMAND
 6408  016602  315  050  034  ACK.EN: CALL    SETM4   ;SEND 'ACK' DOWN THE KLINIK LINE
 6409
 6410  016605                 ACK:    KLINE   M.ACK
  (1)  016605  315  137  004          CALL    KLINE   ;PRINT LINE OF CHARS
  (1)  016610  213  035              .ADDR   M.ACK   ;BUFF TO PRINT
 6411  016612  311                    RET             ;DONE WITH THIS
 6412  016613  001            M.ACK:  .BYTE   SYNC    ;SYNC
 6413  016614  001                    .BYTE   SYNC    ;SYNC
 6414  016615  101                    .BYTE   'A      ;ACKNOWLEDGE CHAR
 6415  016616  033                    .BYTE   ALT     ;ALTMODE
 6416  016617  000                    .BYTE   0       ;END OF STRING
 6417  016620  315  050  034  NACK.EN: CALL   SETM4   ;SEND 'NACK' DOWN THE KLINIK LINE
 6418  016623                 NACK:   KLINE   M.NACK
  (1)  016623  315  137  004          CALL    KLINE   ;PRINT LINE OF CHARS
  (1)  016626  231  035              .ADDR   M.NACK  ;BUFF TO PRINT
 6419  016630  311                    RET             ;BACK TO CALLER
 6420
 6421  016631  001            M.NACK: .BYTE   SYNC    ;SYNC
 6422  016632  001                    .BYTE   SYNC    ;SYNC
 6423  016633  116                    .BYTE   'N      ;NEGATIVE ACKNOWLEDGE CHAR
 6424  016634  033                    .BYTE   ALT     ;ALTMODE
 6425  016635  000                    .BYTE   0       ;END OF STRING
 6426
 6427  016636  052  247  041  ACTION: LHLD    E.BUF   ;GET THE TYPE OF CONTROL THIS WAS(ACK OR NACK)
 6428  016641  176                    MOV     A,M     ;PUT IT INTO ACCUM
 6429  016642  062  261  040          STA     APTANS  ;SET IT INTO THE ANSWER WORD
 6430  016645  303  050  034          JMP     SETM4   ;AND NOW RESET INTERRUPT HANDLER AND OUT
```

```
 6432                           ;THIS IS THE CODE DO DO STRAIGHT OUTPUT FROM THE KLINIK LINE TO THE CTY
 6433                           ;AND INCLUDE A SCHEME FOR BUFFERING THE OUTPUT SO THAT A 9600 BAUD
 6434                           ;KLINIK LINE WILL OUTPUT OK TO A 300 BAUD CTY.
 6435                           ;IF INTERRUPTED WHILE PRINTING A CHARACTER, THE CHARACTERS WAITING TO BE
 6436                           ;PRINTED ARE STACKED AT THE 'SYSOUT' POINTER. CHARACTERS THAT ARE REMOVED FROM
 6437                           ;THE WAITING BUFFER ARE REMOVED VIA THE POINTER 'SYSIN'
 6438  016650  247              MMOUT:  ANA     A       ;SEE IF THIS IS A NULL CHARACTER
 6439  016651  310                      RZ              ;IF YES, DONT DO NOTHIN
 6440  016652  052  123  042          LHLD    SYSOUT  ;SEE IF WE ARE BUSY PRINTING
 6441  016655  174                      MOV     A,H     ;GET AN INDICATOR
 6442  016656  247                      ANA     A       ;SET PC FLAGS
 6443  016657  312  320  035          JZ      NOTBUSY ;GO IF NOT BUSY
 6444
 6445                           ;FELL TO HERE IF BUSY PRINTING
 6446  016662  353                      XCHG            ;SAVE THE CURRENT 'SYSOUT' VALUE(IN D,E)
 6447  016663  052  121  042          LHLD    SYSIN   ; NOW SEE IF THIS IS FIRST TIME IN
 6448  016666  174                      MOV     A,H     ;GET THE INDICATOR
 6449  016667  247                      ANA     A       ;SET FLAGS
 6450  016670  302  301  035          JNZ     STCK.Y  ;JUMP IF ALREADY STACKING
 6451
 6452  016673  041  125  042          LXI     H,SYSBUF ;FIRST TIME IN, SO SET INPUT FLAG
 6453  016676  042  121  042          SHLD    SYSIN   ;SET IT
 6454
 6455  016701  041  053  335  STCK.Y: LXI     H,-SYSEND ;NOW SEE IF BUFFER IS FULL
 6456  016704  031                      DAD     D       ;ADD END TO THE CURRENT TO SEE IF BUFF FULL
 6457  016705  174                      MOV     A,H     ;SEE IF ZERO
 6458  016706  265                      ORA     L       ;SEE IF ZERO
 6459  016707  310                      RZ              ;IF .EQ. 0 THROW AWAY STUFF..BUFF IS FULL
 6460
 6461                           ;HERE IF NOT FULL. MUST STACK THIS CHARACTER
 6462  016710  353                      XCHG            ;CURRENT POINTER GOES BACK TO HL REG(PNT TO SYSOUT)
 6463  016711  160                      MOV     M,B     ;CHAR INTO RAM SPACE
 6464  016712  043                      INX     H       ;UP COUNT
 6465  016713  066  000                MVI     M,0     ;GUARANTEE A ZERO BYTE AT THE END OF BUFFER
 6466  016715  303  004  036          JMP     SETOUT  ;PUT THE POINTER BACK WHERE IT GOES
```

```
 6468                           ;HERE IF NOT PRINTING YET.. PRINT FIRST CHARACTER AND PLAN ON SOME MORE
 6469  016720  041  125  042    NOTBUSY: LXI    H,SYSBUF ;THIS IS THE FIRST TIME IN
 6470  016723  042  123  042             SHLD    SYSOUT  ;SET THE FLAG & THE POINTER
 6471
 6472  016726  170                      MOV     A,B     ;GET CHAR BACK TO ACCUM SO CAN PRINT IT
 6473  016727  107             MORE.:   MOV     B,A     ;WHERE EVER U COME FROM, SAVE ACCUM IN B REG
 6474  016730  376  012                  CPI     LFCHR   ;IS THIS A LINE FEED??
 6475  016732  302  344  035             JNZ     MM.PNT  ;IF NO, NOTHING SPECIAL
 6476
 6477  016735  072  260  040             LDA     CNTLQ.ON ;MUST WE ANSWER EVERY <LF> WITH A 'CNTROL-Q'
 6478  016740  247                      ANA     A       ;IF FLAG .EQ., THEN NO, IF YES THEN WRITE IT
 6479  016741  304  123  004             CNZ     KCHRO   ;YES, A LINE FEED, SEND THE SYSTEM A 'CONTROL-Q'
 6480  016744  170             MM.PNT:  MOV     A,B     ;NO MATTER HOW U GOT HERE, CHAR GOES TO ACCUM
 6481  016745  373                      EI              ;INTERRUPTS ON NOW, BEGIN PRINTING
 6482  016746  315  056  004             CALL    PCHR1Z  ;PRINT A CHAR
 6483
 6484                           ;BACK TO HERE WHEN DONE PRINTING
 6485  016751  363                      DI              ;DON'T BOTHER ME FOR A BIT
 6486  016752  052  121  042             LHLD    SYSIN   ;GRAB POINTER OF THINGS WAITING TO BE PRINTED
 6487  016755  174                      MOV     A,H     ;GET FLAG
 6488  016756  247                      ANA     A       ;SET FLAGS
 6489  016757  312  376  035             JZ      DONE.BUF ;IF NOTHIN, ALL DONE
 6490
 6491                           ;HERE WHEN SOMETHING TO DO
 6492  016762  176                      MOV     A,M     ;GRAB A CHARACTER TO PRINT
 6493  016763  247                      ANA     A       ;MUST FIRST CHECK FOR END OF BUFFER
 6494  016764  312  376  035             JZ      DONE.BUF ;IF DONE RESET THE POINTERS AND GET OUT
 6495
 6496  016767  043                      INX     H       ;NEXT POINT
 6497  016770  042  121  042             SHLD    SYSIN   ;SET INTO RAM
 6498  016773  303  327  035             JMP     MORE.   ;DO MORE
 6499
 6500                           ;HERE ON DONE ALL..FALL INTO Z-BUFF CODE
 6501  016776                   DONE.BUF:
 6502  016776  041  000  000    Z.TBUF:  LXI     H,0     ;WE NEED TO CLEAR SOME BUFFERS
 6503  017001  042  121  042             SHLD    SYSIN   ;CLEAR POINTER
 6504  017004  042  123  042    SETOUT:  SHLD    SYSOUT  ; AND CLEAR POINTER
 6505  017007  311                      RET
```

```
 6507                           .SBTTL **** MORE ERROR ESCAPES ****
 6508  017010                   NOACK:  PLINE   NOA     ;"PRINT NO DATA ACK"
  (1)  017010  337                       RST     3       ;PRINT LINE OF CHARS
  (1)  017011  220  037                   .ADDR   NOA     ;BUFF TO PRINT
 6509
 6510  017013  041  001  000            LXI     H,1     ;ERR CODE IS 1
 6511  017016  042  154  040   ERRRTN:  SHLD    ERRCD   ;SET ERROR CODE
 6512  017021                           ENDCMD
  (1)  017021  311                       RET             ;RETURN TO CALLER
 6513
 6514  017022  257              NOARB:  XRA     A       ;CLR ACCUM
 6515  017023  323  210                OUT     BUSCTL  ;***** I/O WRT 210/0 *****
 6516  017025                           PLINE   NBR     ;"?NO BUS RESP"
  (1)  017025  337                       RST     3       ;PRINT LINE OF CHARS
  (1)  017026  234  037                   .ADDR   NBR     ;BUFF TO PRINT
 6517
 6518  017030  041  002  000            LXI     H,2     ;ERR CODE IS 2
 6519  017033  303  016  036            JMP     ERRRTN  ;GO SET ERROR CODE
 6520
 6521  017036  257              NIXOM:  XRA     A       ;CLR ACCUM
 6522  017037  323  210                OUT     BUSCTL  ;***** I/O WRT 210/0 ***** CLR OUT NIXOM BIT AFTER THE REPORT
 6523  017041                           PLINE   NXMMSG  ;"PRINT ?NXM"
  (1)  017041  337                       RST     3       ;PRINT LINE OF CHARS
  (1)  017042  226  037                   .ADDR   NXMMSG  ;BUFF TO PRINT
 6524  017044  041  003  000            LXI     H,3     ;ERROR CODE 3
 6525
 6526  017047  303  016  036            JMP     ERRRTN  ;SET ERROR CODE
```

```
 6528                           .SBTTL  ****DISK TRANSFER CHANNEL COMMAND LIST****
 6529
 6530  017052                   DSKSEQ:
 6531
 6532                           ;A MINOR NOTE: 40000 WEIGHT BIT IS 'VALID'
 6533                           ;            : 100000 WEIGHT BIT IS '36-BIT XFR'
 6534
 6535  017052                           UBA, 763,001    ;ADDRESS OF UBA IS FIRST ELEMENT OF THE LIST
  (1)  017052  001                     .BYTE   001 & 377
  (1)  017053  346                     .BYTE   Q.1
  (1)  017054  073                     .BYTE   Q.2
 6536  017055                           DI, 140,001     ;VALID & '36 BIT XFR' FOR PAGE 1 (1000-1777)
  (1)  017055  001                     .BYTE   001 & 377
  (1)  017056  300                     .BYTE   Q.1
  (1)  017057  000                     .BYTE   Q.2
 6537  017060                           LI, D776,P.10   ;GET DRIVE STATUS & CONTROL REG, SO CAN SET UNIT
  (1)  017060  374                     .BYTE   Q.1
  (1)  017061  010                     .BYTE   P.10 & 377
  (1)  017062  013                     .BYTE   Q.2
 6538  017063                           DI.INDIRECT UNITNM ;NOW SET THE UNIT NUMBER
  (1)  017063  132  040                .ADDR   UNITNM
  (1)  017065  200                     .BYTE   ^O200
 6539  017066                           EI, D776,P.12   ;ADDRESS OF DRIVE STATUS
  (1)  017066  374                     .BYTE   Q.1
  (1)  017067  012                     .BYTE   P.12 & 377
  (1)  017070  023                     .BYTE   Q.2
 6540  017071                           TWAIT 400       ;CHECK THAT THE DRIVE IS PRESENT
  (1)  017071  000  001                .ADDR   400
  (1)  017073  060                     .BYTE   <14*4>
 6541  017074                           WAIT 200        ;CHECK & WAIT FOR READY
  (1)  017074  200  000                .ADDR   200
  (1)  017076  030                     .BYTE   <6*4>
 6542  017077                           LI, D776,P.10   ;ADDR OF DRIVE STATUS REG
  (1)  017077  374                     .BYTE   Q.1
  (1)  017100  010                     .BYTE   P.10 & 377
  (1)  017101  013                     .BYTE   Q.2
 6543  017102                           DI, 0,40        ;ISSUE CONTROLLER CLR
  (1)  017102  040                     .BYTE   40 & 377
  (1)  017103  000                     .BYTE   Q.1
  (1)  017104  000                     .BYTE   Q.2
 6544  017105                           DI.INDIRECT UNITNM ;SET TO UNIT #
  (1)  017105  132  040                .ADDR   UNITNM
  (1)  017107  200                     .BYTE   ^O200
 6545  017110                           LI, D776,P.00   ;ADDR OF CONTROLLER STATUS REG
  (1)  017110  374                     .BYTE   Q.1
  (1)  017111  000                     .BYTE   P.00 & 377
  (1)  017112  013                     .BYTE   Q.2
 6546  017113                           DI, 0,11        ;ISSUE DRIVE CLEAR
  (1)  017113  011                     .BYTE   11 & 377
  (1)  017114  000                     .BYTE   Q.1
  (1)  017115  000                     .BYTE   Q.2
 6547  017116                           DI, 0,21        ;SET 'READ-IN-PRESET'
  (1)  017116  021                     .BYTE   21 & 377
  (1)  017117  000                     .BYTE   Q.1
```

```
  (1)  017120  000                    .BYTE   Q.2
 6548  017121                       LI. D776,P.12   ;GET TO THE DRIVE STATUS REGISTER
  (1)  017121  374                    .BYTE   Q.1
  (1)  017122  012                    .BYTE   P.12 & 377
  (1)  017123  013                    .BYTE   Q.2
 6549  017124                       WAIT 200        ;WAIT FOR IT TO BE READY
  (1)  017124  200  000               .ADDR   200
  (1)  017126  030                    .BYTE   <6*4>
 6550  017127                       TWAIT   100     ;NOW CHECK THAT 'PRE-SET' HAS SET VOLUME VALID
  (1)  017127  100  000               .ADDR   100
  (1)  017131  060                    .BYTE   <14*4>
 6551  017132                       LI. D776,P.06   ;ADDR TRACK/SECTOR REG
  (1)  017132  374                    .BYTE   Q.1
  (1)  017133  006                    .BYTE   P.06 & 377
  (1)  017134  013                    .BYTE   Q.2
 6552  017135                       DI.INDIRECT     BLKNUM ;EMPTY FOR NOW
  (1)  017135  105  040               .ADDR   BLKNUM
  (1)  017137  200                    .BYTE   ^O200
 6553  017140                       LI. D776,P.34   ;ADDR OF CYLINDER REG
  (1)  017140  374                    .BYTE   Q.1
  (1)  017141  034                    .BYTE   P.34 & 377
  (1)  017142  013                    .BYTE   Q.2
 6554  017143                       DI.INDIRECT     BLKADR ;EMPTY FOR NOW
  (1)  017143  112  040               .ADDR   BLKADR
  (1)  017145  200                    .BYTE   ^O200
 6555  017146                QXFR:  LI. D776,P.02   ;ADD OF WORD COUNT REG
  (1)  017146  374                    .BYTE   Q.1
  (1)  017147  002                    .BYTE   P.02 & 377
  (1)  017150  013                    .BYTE   Q.2
 6556  017151                       DI. 176,000     ;512 WORDS IS 1024 18-BIT BYTES(A PAGE)
  (1)  017151  000                    .BYTE   000 & 377
  (1)  017152  374                    .BYTE   Q.1
  (1)  017153  000                    .BYTE   Q.2
 6557  017154                       LI. D776,P.04   ;ADDR OF UNIBUS ADDRESS REG
  (1)  017154  374                    .BYTE   Q.1
  (1)  017155  004                    .BYTE   P.04 & 377
  (1)  017156  013                    .BYTE   Q.2
 6558  017157                       DI. 4,000       ;SET SM10 MEM ADDR TO 1000
  (1)  017157  000                    .BYTE   000 & 377
  (1)  017160  010                    .BYTE   Q.1
  (1)  017161  000                    .BYTE   Q.2
 6559  017162                       LI. D776,P.00   ;BACK TO STATUS REG
  (1)  017162  374                    .BYTE   Q.1
  (1)  017163  000                    .BYTE   P.00 & 377
  (1)  017164  013                    .BYTE   Q.2
 6560  017165                       DI. 0,71        ;ISSUE READ
  (1)  017165  071                    .BYTE   71 & 377
  (1)  017166  000                    .BYTE   Q.1
  (1)  017167  000                    .BYTE   Q.2
 6561  017170                       EI. D776,P.00   ;NOW READ TO CHECK FOR ERRORS IN XFER
  (1)  017170  374                    .BYTE   Q.1
  (1)  017171  000                    .BYTE   P.00 & 377
  (1)  017172  023                    .BYTE   Q.2
 6562  017173                       WAIT 200        ;CHECK FOR READY BIT TRUE..
```

```
  (1)  017173  200  000              ,ADDR   200
  (1)  017175  030                   ,BYTE   <6*4>
 6563  017176                        EI, D776,P,12   ;CHECK DRIVE STATUS REG ITSELF
  (1)  017176  374                   ,BYTE   Q,1
  (1)  017177  012                   ,BYTE   P,12 & 377
  (1)  017200  023                   ,BYTE   Q,2
 6564  017201                        ERRTST  40000   ;TEST 'ERR' BIT
  (1)  017201  000  100              ,ADDR   40000
  (1)  017203  040                   ,BYTE   <10*4>
 6565                         ;IF IT WAS OK, THEN CHECK THE CONTROLLER FOR ERRORS
 6566  017204                        EI, D776,P,00   ;EXAMINE CONTROLLER
  (1)  017204  374                   ,BYTE   Q,1
  (1)  017205  000                   ,BYTE   P,00 & 377
  (1)  017206  023                   ,BYTE   Q,2
 6567  017207                        ERRTST 060000   ;DRIVE RDY, NOW SEE IF ENCOUNTERED ERRORS
  (1)  017207  000  140              ,ADDR   060000
  (1)  017211  040                   ,BYTE   <10*4>
 6568  017212                        ENDLST          ;END OF CHANNEL COMMAND LIST
  (1)  017212  000                   ,BYTE   0
  (1)  017213  000                   ,BYTE   0
  (1)  017214  050                   ,BYTE   <12*4>
```

```
 6570                          .SBTTL  TAPE TRANSFER CHANNEL COMMAND LIST
 6571                          ;DRIVE CONTROLLER REGISTER IS 776440..FOLLOWING COMMANDS APPLY
 6572                          ;       7=REWIND
 6573                          ;       11=DRIVE CLEAR
 6574                          ;       25=ERASE
 6575                          ;       27=WRITE TAPE MARK
 6576                          ;       31=SPACE FORWARD(SKIP A FILE)
 6577                          ;       33=SPACE REVERSE(SKIP A FILE, MOVING TAPE IN REVERSE)
 6578                          ;       51=WRITE CHECK FORWARD
 6579                          ;       57=WRITE CHECK REVERSE
 6580                          ;       61=WRITE FORWARD
 6581                          ;       71=READ FORWARD(GO!)
 6582                          ;       77=READ REVERSE(GO!)
 6583
 6584  017215                  MTASEQ: UBA, 763,001    ;ADDRESS OF UBA PAGING RAM
  (1)  017215  001                      .BYTE   001 & 377
  (1)  017216  346                      .BYTE   Q.1
  (1)  017217  073                      .BYTE   Q.2
 6585  017220                          DI, 40,001      ;SET VALID IN PAGE 1
  (1)  017220  001                      .BYTE   001 & 377
  (1)  017221  100                      .BYTE   Q.1
  (1)  017222  000                      .BYTE   Q.2
 6586  017223                          LI, D772,P.10   ;SET ADDRESS OF DRIVE CONTROL REGISTER
  (1)  017223  364                      .BYTE   Q.1
  (1)  017224  010                      .BYTE   P.10 & 377
  (1)  017225  013                      .BYTE   Q.2
 6587  017226                          DI, 0,40                ;ISSUE CONTROLLER AND SLAVE CLR
  (1)  017226  040                      .BYTE   40 & 377
  (1)  017227  000                      .BYTE   Q.1
  (1)  017230  000                      .BYTE   Q.2
 6588  017231                          DI.INDIRECT TAPEUNIT ;SET TAPE UNIT #
  (1)  017231  137  040                 .ADDR   TAPEUNIT
  (1)  017233  200                      .BYTE   ^O200
 6589  017234                          LI, D772,P.32   ;SLAVE SELECT/FORMAT/DENSITY REG
  (1)  017234  364                      .BYTE   Q.1
  (1)  017235  032                      .BYTE   P.32 & 377
  (1)  017236  013                      .BYTE   Q.2
 6590  017237                          DI.INDIRECT DEN.SLV ;SET SLAVE, FORMAT, DENSITY(TEMP:DENS=1600,F=0,S=0)
  (1)  017237  362  040                 .ADDR   DEN.SLV
  (1)  017241  200                      .BYTE   ^O200
 6591  017242                          EI, D772,P.12   ;READ THE DRIVE STATUS, TO MAKE SURE IT EXISTS
  (1)  017242  364                      .BYTE   Q.1
  (1)  017243  012                      .BYTE   P.12 & 377
  (1)  017244  023                      .BYTE   Q.2
 6592  017245                          TWAIT 400       ;CHECK THE 'DRIVE PRESENT' BIT
  (1)  017245  000  001                 .ADDR   400
  (1)  017247  060                      .BYTE   <14*4>
 6593  017250                          WAIT 200        ;IF WAS PRESENT, WAIT FOR IT TO BE READY
  (1)  017250  200  000                 .ADDR   200
  (1)  017252  030                      .BYTE   <6*4>
 6594  017253                          LI, D772,P.06   ;FRAME COUNT REGISTER
  (1)  017253  364                      .BYTE   Q.1
  (1)  017254  006                      .BYTE   P.06 & 377
  (1)  017255  013                      .BYTE   Q.2
```

```
 6595  017256                          DI, 0,0         ;FRAME COUNT TO 0 IS MAX NUMB OF FRAMES
  (1)  017256  000                     .BYTE   0 & 377
  (1)  017257  000                     .BYTE   Q.1
  (1)  017260  000                     .BYTE   Q.2
 6596  017261                          LI, D772,P.00   ;SET ADDRESS TO RH11 CONTROL REGISTER
  (1)  017261  364                     .BYTE   Q.1
  (1)  017262  000                     .BYTE   P.00 & 377
  (1)  017263  013                     .BYTE   Q.2
 6597  017264                          DI, 0,7         ;ISSUE "REWIND" TO TAPE
  (1)  017264  007                     .BYTE   7 & 377
  (1)  017265  000                     .BYTE   Q.1
  (1)  017266  000                     .BYTE   Q.2
 6598  017267                          EI, D772,P.12   ;READ THE DRIVE STATUS TO CHECK FOR READY
  (1)  017267  364                     .BYTE   Q.1
  (1)  017270  012                     .BYTE   P.12 & 377
  (1)  017271  023                     .BYTE   Q.2
 6599  017272                          WAIT 200        ;WAIT FOR REWIND TO COMPLETE
  (1)  017272  200  000                .ADDR   200
  (1)  017274  030                     .BYTE   <6*4>
 6600  017275                  QTXFR:  LI, D772,P.04   ;CONTROLLER TO MEMORY DEST. REG
  (1)  017275  364                     .BYTE   Q.1
  (1)  017276  004                     .BYTE   P.04 & 377
  (1)  017277  013                     .BYTE   Q.2
 6601  017300                          DI, 4,000       ;SET KS10 START ADDRESS TO 1000
  (1)  017300  000                     .BYTE   000 & 377
  (1)  017301  010                     .BYTE   Q.1
  (1)  017302  000                     .BYTE   Q.2
 6602  017303                          LI, D772,P.02   ;SET ADDRESS TO WORD COUNT REGISTER
  (1)  017303  364                     .BYTE   Q.1
  (1)  017304  002                     .BYTE   P.02 & 377
  (1)  017305  013                     .BYTE   Q.2
 6603  017306                          DI, 176,000     ;1 PAGE OF 512 WORDS IS 1024 18-BIT BYTES
  (1)  017306  000                     .BYTE   000 & 377
  (1)  017307  374                     .BYTE   Q.1
  (1)  017310  000                     .BYTE   Q.2
 6604  017311                          LI, D772,P.06   ;FRAME COUNT REGISTER
  (1)  017311  364                     .BYTE   Q.1
  (1)  017312  006                     .BYTE   P.06 & 377
  (1)  017313  013                     .BYTE   Q.2
 6605  017314                          DI, 0,0         ;FRAME COUNT TO 0 IS MAX NUMB OF FRAMES
  (1)  017314  000                     .BYTE   0 & 377
  (1)  017315  000                     .BYTE   Q.1
  (1)  017316  000                     .BYTE   Q.2
 6606  017317                          LI, D772,P.00   ;SET ADDRESS TO RH11 CONTROL REGISTER
  (1)  017317  364                     .BYTE   Q.1
  (1)  017320  000                     .BYTE   P.00 & 377
  (1)  017321  013                     .BYTE   Q.2
 6607  017322                          DI,INDIRECT SKP.GO ;ISSUE 'XFER CMD'(31=SKIP .OR. 71=RD-IN)
  (1)  017322  144  040                .ADDR   SKP.GO
  (1)  017324  200                     .BYTE   ^O200
 6608  017325                          EI, D772,P.12   ;READ THE DRIVE STATUS TO CHECK FOR READY
  (1)  017325  364                     .BYTE   Q.1
  (1)  017326  012                     .BYTE   P.12 & 377
  (1)  017327  023                     .BYTE   Q.2
```

```
 6609  017330                          WAIT 200        ;WAIT FOR FILE READ TO COMPLETE
  (1)  017330  200  000                ,ADDR   200
  (1)  017332  030                     ,BYTE   <6*4>
 6610  017333                          EI, D772,P,14   ;LOOK AT THE DRIVE ERROR REGISTER
  (1)  017333  364                     ,BYTE   Q,1
  (1)  017334  014                     ,BYTE   P,14 & 377
  (1)  017335  023                     ,BYTE   Q,2
 6611  017336                RETRY,: ERRTST 070300   ;ERRORS WORTH RETRYING??
  (1)  017336  300  160                ,ADDR   070300
  (1)  017340  040                     ,BYTE   <10*4>
 6612  017341                FRMERR: ERRTST 103400   ;SEE IF THIS WAS A CORRECTABLE TYPE ERROR
  (1)  017341  000  207                ,ADDR   103400
  (1)  017343  040                     ,BYTE   <10*4>
 6613  017344                          EI, D772,P,12   ;READ THE DRIVE STATUS TO CHECK FOR ERRORS
  (1)  017344  364                     ,BYTE   Q,1
  (1)  017345  012                     ,BYTE   P,12 & 377
  (1)  017346  023                     ,BYTE   Q,2
 6614  017347                          ERRTST 40000    ;GET ANY DRIVE ERRORS??
  (1)  017347  000  100                ,ADDR   40000
  (1)  017351  040                     ,BYTE   <10*4>
 6615  017352                          LI, D772,P,00   ;NOW ADDRESS TO CONTROLLER STATUS
  (1)  017352  364                     ,BYTE   Q,1
  (1)  017353  000                     ,BYTE   P,00 & 377
  (1)  017354  013                     ,BYTE   Q,2
 6616  017355                          ERRTST 60000    ;CHECK FOR ERRORS THERE,
  (1)  017355  000  140                ,ADDR   60000
  (1)  017357  040                     ,BYTE   <10*4>
 6617  017360                          ENDLST          ;END OF CHANNEL COMMAND LIST
  (1)  017360  000                     ,BYTE   0
  (1)  017361  000                     ,BYTE   0
  (1)  017362  050                     ,BYTE   <12*4>
```

```
 6619                           ;QUICK LITTLE ROUTINE TO RESET THE MAGTAPE AFTER IT SUFFERS AN
 6620                           ;IGNORABLE ERROR
 6621  017363                   MTARST: LI, D772,P,10   ;SET ADDRESS OF DRIVE CONTROL REGISTER
  (1)  017363  364                      ,BYTE   Q,1
  (1)  017364  010                      ,BYTE   P,10 & 377
  (1)  017365  013                      ,BYTE   Q,2
 6622  017366                         DI, 0,40        ;ISSUE CONTROLLER AND SLAVE CLR
  (1)  017366  040                      ,BYTE   40 & 377
  (1)  017367  000                      ,BYTE   Q,1
  (1)  017370  000                      ,BYTE   Q,2
 6623  017371                         DI,INDIRECT TAPEUNIT ;FIX UNIT NUMBER
  (1)  017371  137  040                 ,ADDR   TAPEUNIT
  (1)  017373  200                      ,BYTE   ^D200
 6624  017374                         LI, D772,P,04   ;CONTROLLER TO MEMORY DEST, REG
  (1)  017374  364                      ,BYTE   Q,1
  (1)  017375  004                      ,BYTE   P,04 & 377
  (1)  017376  013                      ,BYTE   Q,2
 6625  017377                         DI, 4,000       ;SET KS10 START ADDRESS TO 1000
  (1)  017377  000                      ,BYTE   000 & 377
  (1)  017400  010                      ,BYTE   Q,1
  (1)  017401  000                      ,BYTE   Q,2
 6626  017402                         LI, D772,P,06   ;FRAME COUNT REGISTER
  (1)  017402  364                      ,BYTE   Q,1
  (1)  017403  006                      ,BYTE   P,06 & 377
  (1)  017404  013                      ,BYTE   Q,2
 6627  017405                         DI, 0,0         ;FRAME COUNT TO 0 IS MAX NUMB OF FRAMES
  (1)  017405  000                      ,BYTE   0 & 377
  (1)  017406  000                      ,BYTE   Q,1
  (1)  017407  000                      ,BYTE   Q,2
 6628  017410                         ENDLST          ;QUICK OUT
  (1)  017410  000                      ,BYTE   0
  (1)  017411  000                      ,BYTE   0
  (1)  017412  050                      ,BYTE   <12*4>
```

```
 6630  017413  077  102  125   .IIF NDF,NOROOM INIER:  .ASCIZ /?BUS\/          ;BUS POLLUTED ON POWER UP
 6631  017421  077  102  106   BV:     .ASCIZ /?BFO/           ;INPUT BUFFER OVERFLOW
 6632  017426  077  111  114   CMDNG:  .ASCII   /?IL/          ;ILLEGAL INSTRUCTION
 6633  017431  015  012  000           .BYTE   ^O15,^O12,0     ;CR-LF
 6634  017434  077  125  111   TTM:    .ASCIZ /?UI\/           ;UNKNOWN INTERRRUPT
 6635  017441  102  125  123   EBHED:  .ASCIZ   /BUS 0-35\/    ;MESSAGE HEADER FOR "EB" CMD
 6636  017453  113  123  061   KSPRMT: .ASCII /KS10>/          ;PROMPT MESSAGE
 6637  017460  377  000           .BYTE   ^O377,0
 6638  017462  040  103  131   MSG10:  .ASCIZ % CYC\SENT/%     ;CYCLE TYPE FOR "DB" COMMAND
 6639  017475  134  122  103   DRCVD:  .ASCIZ %\RCVD/%         ;DATA RECEIVED ON BUS(DB CMD)
 6640  017504  077  101  057   ECVER:  .ASCIZ %?A/B\%          ;A & B COPIES OF CRAM BITS DID NOT MATCH
 6641  017512  120  103  057   PCMSG:  .ASCIZ %PC/%            ;OBVIOUS
 6642  017516  045  110  114   HLTMS:  .ASCIZ .%HLTD/.         ;MESSAGE "HALTED/XXXXXX" WHERE XXXXXX IS DATA
 6643  017525  077  104  116   EXMS:   .ASCIZ /?DNF\/          ;"DID NOT FINISH"-INSTR
 6644  017533  077  102  124   BTFAIL: .ASCIZ /?BT /            ;DEVICE ERROR OR TIMEOUT DURING BOOT OPERATION
 6645  017540  102  124  040   BTMSG1: .ASCIZ /BT SW/           ;MESSAGE SAYS BOOTING, USING BOOT SW
 6646  017546  077  104  116   SMERR:  .ASCIZ  /?DNC\/         ;"DID NOT COMPLETE"-THE HALT
 6647  017554  117  106  106   OFFMSG: .ASCIZ /OFF\/           ;MESSAGE SAYS SIGNAL IS 'OFF'
 6648  017561  077  120  101   PARMSG: .ASCIZ /?PAR ERR /      ;REPORT CLOCK FREEZE DUE TO PAR ERR
 6649  017573  077  115  122   MOSMSG: .ASCIZ /?MRE\/           ;MEMORY REFRESH ERROR
 6650  017601  077  102  103   ERRMSG: .ASCIZ /?BC /           ;TA-DA...,BOOT CHECK
 6651  017606  077  122  125   RN.:    .ASCIZ /?RUNNING\/      ;TRYING TO DO A CMD THAT MAY SCREW UP
 6652  017620  077  116  104   NDA:    .ASCIZ /?NDA\/          ;RECEIVED NO DATA ACKNOWLEDGE ON MEM REQUEST
 6653  017626  077  116  130   NXMMSG: .ASCIZ /?NXM\/          ;REFERRENCED NON EXISTANT MEMORY LOCATION
 6654  017634  077  116  102   NBR:    .ASCIZ /?NBR\/          ;CONSOLE WAS NOT GRANTED BUS ON A REQUEST
 6655  017642  077  122  101   RAG:    .ASCIZ /?RA/            ;CMD REQUIRES ARGUEMENT..U GOTTA TYPE SOMETHING
 6656  017646  077  102  116   BB1:    .ASCIZ /?BN/            ;TYPED A BAD NUMBER(I.E 9 OR X OR # ETC.)
 6657  017652  076  076  125   Q.UBA:  .ASCIZ />>UBA?/         ;QUERY FOR UNIBUS ADAPTER
 6658  017661  076  076  122   Q.RH:   .ASCIZ />>RHBASE?/      ;QUERY FOR RH11 TO USE
 6659  017673  076  076  125   Q.UNIT: .ASCIZ />>UNIT?/        ;QUERY FOR UNIT TO USE
 6660  017703  076  076  124   Q.TCU:  .ASCIZ />>TCU?/         ;QUERY FOR TAPE CONTROLL UNIT
 6661  017712  076  076  104   Q.DEN:  .ASCIZ />>DENS?/        ;QUERY FOR TAPE DENSITY
 6662  017722  076  076  123   Q.SLV:  .ASCIZ />>SLV?/         ;QUERY FOR TAPE SLAVE
 6663  017731  077  113  101   KAMSG:  .ASCIZ /?KA\/           ;KEEP ALIVE FAILED
 6664  017736  077  106  122   FRCMSG: .ASCIZ /?FRC\/          ;HAD A FORCED RELOAD
 6665  017744  077  120  127   PWLEN:  .ASCIZ /?PWL/           ;PASSWORD LENGTH ERROR
 6666  017751  077  116  101   NOACCS: .ASCII /?NA/            ;NOT AVAILABLE (KLINIK LINE THAT IS)
 6667  017754  015  012  000           .BYTE   ^O15,^O12,0     ;CR-LF
 6668  017757  120  127  072   QPW:    .ASCII /PW:/            ;ASK FOR A PASSWORD MESSAGE
 6669  017762  015  012  000           .BYTE   ^O15,^O12,0     ;CR-LF
 6670  017765  102  124  040   AUTOMS: .ASCIZ /BT AUTO/        ;BEGINNING AUTO BOOT SEQUENCE
```

```
                        .SBTTL - SM10 DATA STORAGE PARAMETERS
                        ;NOTE THAT ALL THESE LOCATIONS ARE RAM LOCATIONS
                        ;AND MAY BE USED WITH THE
                        ;ASSEMBLY OF SEPARATE MODULES OF CONSOLE CODE....
      020000                    .=      RAMST
020000 000002           T80DT:  .BLKB   2
020002 000002           C80AD:  .BLKB   2
020004 000002           CRMAD:  .BLKB   2
020006 000002           BYTCNT: .BLKB   2
020010 000002           CRMFN:  .BLKB   2
020012 000005           EMBUF:  .BLKB   5
020017 000005           MEMAD:  .BLKB   5
020024 000005           IOAD:   .BLKB   5
020031 000002           ENEXT:  .BLKB   2
020033 000002           DNEXT:  .BLKB   2
020035 000005           CHRBUF: .BLKB   5
020042 000005           BUSAD:  .BLKB   5
020047 000005           DMDAT:  .BLKB   5
020054 000006           RAMX1:  .BLKB   6
020062 000006           RAMX2:  .BLKB   6
020070 000003           ER.LOC: .BLKB   3       ;PLACE TO EXECUTE 8080 'IN'S' AND 'OUT'S'
020073 000005           TMPBF2: .BLKB   5
020100 000005           TMPB2:  .BLKB   5
020105 000005           BLKNUM: .BLKB   5
020112 000005           BLKADR: .BLKB   5
020117 000001           EXM1:   .BLKB   1
020120 000001           NOPNT:  .BLKB   1
020121 000001           BT.TYPE: .BLKB  1
020122 000001           P8.TMP: .BLKB   1
020123 000001           ERADDR: .BLKB   1
020124 000001           KLNKSW: .BLKB   1
020125 000002           KPWPNT: .BLKB   2
020127 000001           KPWCNT: .BLKB   1
020130 000001           WATCHC: .BLKB   1
020131 000001           CMDS..: .BLKB   1
020132 000005           UNITNM: .BLKB   5
020137 000005           TAPEUNIT: .BLKB 5
020144 000005           SKP.GO: .BLKB   5
020151 000001           BRKON:  .BLKB   1
020152 000002           BRKDT:  .BLKB   2
020154 000002           ERRCD:  .BLKB   2
020156 000001           USRMD:  .BLKB   1
020157 000001           RPEND:  .BLKB   1
020160 000001           RPCNTR: .BLKB   1
020161 000001           BFCNT:  .BLKB   1
020162 000001           STPPD:  .BLKB   1
020163 000001           EIFLAG: .BLKB   1
020164 000001           DIFLAG: .BLKB   1
020165 000001           RNFLG:  .BLKB   1
020166 000001           CHKPAR: .BLKB   1
020167 000001           CHKREF: .BLKB   1
020170 000002           ECSAV:  .BLKB   2
020172 000010           RM100:  .BLKB   8
020202 000001           BTMSK:  .BLKB   1
```

```
 6726  020203 000001                  BTNUM:  .BLKB   1
 6727  020204 000001                  EOL:    .BLKB   1
 6728  020205 000002                  AM.AI:  .BLKB   2
 6729  020207 000002                  RPBUFS: .BLKB   2
 6730  020211 000002                  RPLST:  .BLKB   2
 6731  020213 000001                  RPTON:  .BLKB   1
 6732  020214 000001                  CHRCNT: .BLKB   1
 6733  020215 000002                  BUF.:   .BLKB   2
 6734  020217 000002                  HLSAVE: .BLKB   2
 6735  020221 000002                  CMD..:  .BLKB   2
 6736  020223 000002                  .ARG1:  .BLKB   2
 6737  020225 000002                  CMCNT:  .BLKB   2
 6738  020227 000002                  FIRST:  .BLKB   2
 6739  020231 000002                  CHKHLT: .BLKB   2
 6740  020233 000002                  OCTSV:  .BLKB   2
 6741  020235 000005                  SHRTAD: .BLKB   5
 6742  020242 000005                  RHBASE: .BLKB   5
 6743  020247 000001                  CSLMODE: .BLKB  1
 6744  020250 000001                  KACNTR: .BLKB   1
 6745  020251 000001                  MMFLG:  .BLKB   1
 6746  020252 000001                  MAILFG: .BLKB   1
 6747  020253 000001                  VERCNT: .BLKB   1
 6748  020254 000001                  UBANUM: .BLKB   1
 6749  020255 000001                  GOCODE: .BLKB   1
 6750  020256 000001                  SECRET: .BLKB   1
 6751  020257 000001                  DIECNT: .BLKB   1
 6752  020260 000001                  CNTLQ.ON: .BLKB 1
 6753  020261 000001                  APTANS: .BLKB   1
 6754  020262 000007                  PASSWORD: .BLKB 7
 6755  020271 000006                  KPWBUF: .BLKB   6
 6756  020277 000001                  PWRTRY: .BLKB   1
 6757  020300 000001                  KLLINE.ON: .BLKB 1
 6758  020301 000001                  E.CNT:  .BLKB   1
 6759  020302 000020                  CRMBF:  .BLKB   ^D16
 6760  020322 000020                  CRMTM:  .BLKB   ^D16
 6761  020342 000002                  KATIM1: .BLKB   2
 6762  020344 000002                  .IIF DF,SCECOD,SCEADR:  .BLKB   2       ;LAST FAILING ADDRESS
 6763  020346 000002                  MODDIS: .BLKB   2
 6764  020350 000002                  NOREND: .BLKB   2
 6765  020352 000002                  ENVPNT: .BLKB   2
 6766  020354 000001                  PARBT:  .BLKB   1
 6767  020355 000001                  TRAPEN: .BLKB   1
 6768  020356 000001                  MTAUBA: .BLKB   1
 6769  020357 000001                  DSKUBA: .BLKB   1
 6770  020360 000001                  STATE:  .BLKB   1
 6771  020361 000001                  LSTMSG: .BLKB   1
 6772  020362 000005                  DEN.SLV: .BLKB  5
 6773  020367 000005                  MTBASE: .BLKB   5
 6774  020374 000005                  DSBASE: .BLKB   5
 6775  020401 000031                  RPINI:  .BLKB   ^D25
 6776  020432 000062                  RPTBFI: .BLKB   ^D50
 6777  020514 000132                  BUFBG:  .BLKB   ^D90
 6778  020646 000001                  BUFEN:  .BLKB   1
 6779  020647 000002                  E.BUF:  .BLKB   2
```

```
 6780  020651 000140                E.BEG:  .BLKB   ^O140
 6781  021011 000001                ENVMNO: .BLKB   1
 6782  021012 000001                ENVCHK: .BLKB   1
 6783  021013 000106                ENVBUF: .BLKB   70      ;EB IS LONGEST COMMAND WITH 67. CHARS
 6784  021121 000002                SYSIN:  .BLKB   2
 6785  021123 000002                SYSOUT: .BLKB   2
 6786  021125 000200                SYSBUF: .BLKB   ^O200
 6787  021325 000001                SYSEND: .BLKB   1       ;AND EMPTY BYTE TO HOLD END OF LIST
 6788       001              .IF     DF,SCECOD
 6789  021326 000001                SC.OFF: .BLKB   1       ;RECOVERY ON OR OFF FLAG
 6790  021327 000050                RHSAVE: .BLKB   40.     ;BUFFER FOR SAVING RH STUFF
 6791       000              .ENDC
 6792  021377                FREE:
 6793       000000                   .END            ;END STATEMENT
```

END.M80 CROSS REFERENCE TABLE

```
A     =%000007          532   567   580   630   638   652   655   658   659   662   665   667   668
                        670   673   680   687   691   704   705   706   707   708   717   718   722
                        727   731   732   744   747   749   757   764   775   795   802   815   838
                        840   869   887   928   932   947   950   960   971   998  1013  1035  1048
                       1064  1068  1075  1076  1085  1115  1148  1150  1152  1167  1174  1176  1182
                       1190  1199  1217  1236  1266  1278  1283  1291  1293  1307  1312  1314  1321
                       1328  1330  1338  1339  1340  1342  1365  1381  1419  1440  1455  1461  1640
                       1645  1648  1666  1673  1680  1687  1689  1698  1706  1721  1722  1735  1745
                       1759  1795  1799  1803  1807  1811  1839  1841  1849  1851  1857  1897  1923
                       1929  1932  1937  1941  1945  1952  1956  1962  1963  1967  1977  1979  1982
                       1985  1986  1993  1996  2003  2007  2012  2018  2023  2045  2048  2054  2072
                       2084  2102  2105  2112  2114  2123  2127  2130  2133  2139  2147  2152  2156
                       2163  2168  2225  2229  2247  2268  2293  2299  2312  2315  2322  2326  2331
                       2337  2342  2351  2363  2369  2418  2421  2438  2441  2492  2497  2499  2505
                       2509  2514  2518  2522  2539  2543  2587  2592  2594  2597  2600  2606  2611
                       2615  2619  2654  2659  2662  2676  2681  2684  2701  2710  2754  2756  2757
                       2769  2777  2779  2791  2809  2821  2833  2845  2865  2870  2900  2904  2922
                       2923  2927  2930  2936  2945  2973  2983  2998  3006  3008  3015  3017  3044
                       3046  3049  3052  3062  3064  3072  3074  3078  3080  3083  3090  3092  3094
                       3101  3105  3117  3125  3127  3131  3134  3149  3152  3157  3172  3185  3188
                       3211  3217  3227  3256  3262  3264  3280  3282  3290  3307  3360  3375  3377
                       3391  3462  3485  3492  3523  3526  3541  3554  3555  3556  3572  3586  3596
                       3611  3612  3630  3659  3672  3696  3710  3718  3730  3737  3751  3754  3762
                       3763  3767  3768  3772  3776  3777  3782  3785  3791  3805  3806  3811  3896
                       3906  3931  3953  3984  3999  4002  4010  4012  4021  4028  4037  4049  4066
                       4069  4074  4079  4084  4093  4184  4238  4243  4251  4253  4255  4259  4261
                       4270  4271  4273  4275  4277  4287  4295  4302  4304  4312  4330  4390  4400
                       4407  4411  4430  4445  4454  4455  4457  4464  4467  4520  4568  4570  4625
                       4630  4661  4674  4681  4697  4702  4722  4723  4724  4738  4767  4781  4822
                       4831  4842  4866  4881  4914  4933  4951  4953  4954  4956  4966  4972  4985
                       4991  5006  5018  5023  5028  5038  5109  5134  5149  5167  5169  5200  5205
                       5230  5231  5251  5256  5258  5317  5424  5432  5433  5450  5454  5456  5458
                       5479  5494  5496  5498  5499  5501  5508  5510  5512  5534  5537  5540  5544
                       5551  5555  5570  5571  5587  5589  5603  5612  5613  5627  5628  5639  5654
                       5658  5675  5676  5678  5697  5707  5733  5735  5736  5754  5762  5796  5804
                       5831  5832  5894  5898  6021  6024  6039  6057  6059  6073  6102  6131  6137
                       6146  6151  6157  6165  6174  6177  6187  6198  6199  6204  6211  6222  6226
                       6232  6234  6257  6265  6349  6354  6363  6366  6393  6396  6428  6438  6441
                       6442  6448  6449  6457  6472  6473  6478  6480  6487  6488  6492  6493  6514
                       6521
ACK       016605       1972  6410#
ACK.EN    016602       6390  6408#
ACTION    016636       6331  6334  6427#
ADATP     015576       2503  2595  3708  3729  4772  5993#
ADATT     015571       2415  2432  5988#
ALT     = 000033         94# 6333  6415  6424
AM.AI     020205       2501  2546  6728#
APTANS    020261       6243  6256  6429  6753#
APT.WT    016322       6256# 6258
ARBRES= 000020           60# 2424  2445  2525  2622
ARG     = 000000       5073# 5078# 5080# 5084# 6535# 6536# 6537# 6539# 6542# 6543# 6545# 6546# 6547#
                       6548# 6551# 6553# 6555# 6556# 6557# 6558# 6559# 6560# 6561# 6563# 6566# 6584#
                       6585# 6586# 6587# 6589# 6591# 6594# 6595# 6596# 6597# 6598# 6600# 6601# 6602#
                       6603# 6604# 6605# 6606# 6608# 6610# 6613# 6615# 6621# 6622# 6624# 6625# 6626#
```

```
                        6627#
ARGBEG  014643          5538    5541#
ARGBG1  014644          5535    5542#
ARGQQ   014753          5595#   5606
ARG1  = 000000          5073#   5078#   5080#   5084#   6535#   6536#   6537#   6539#   6542#   6543#   6545#   6546#   6547#
                        6548#   6551#   6553#   6555#   6556#   6557#   6558#   6559#   6560#   6561#   6563#   6566#   6584#
                        6585#   6586#   6587#   6589#   6591#   6594#   6595#   6596#   6597#   6598#   6600#   6601#   6602#
                        6603#   6604#   6605#   6606#   6608#   6610#   6613#   6615#   6621#   6622#   6624#   6625#   6626#
                        6627#
ARG16.  014244          2752    2766    2788    2831    2863    2898    2921    2943    3025    3035    3402    3483    3608
                        3760    3997    4340    5284#
ARG96   014627          3351    5534#
ARROW = 000076           112#
AUTOMS  017765           910    6670#
A.BTER  015201          4026    4188    4214    4409    5720#
A.CRMO  013226          4878    4925#
A.CRMZ  013240          4930#
A0003 = 000113           49#    2004    2124    2323    2506    2598    3086    6011
A0411 = 000111           48#    3085    6007
A1219 = 000107           47#    1998    2116    2317    3084    6003
A16PK   014752          5593#   5609
A2027 = 000105           46#    1997    2115    2316    3047    3081    4254    5999
A256    000224          710#    719     724
A2835 = 000103           45#    1995    2113    2314    3045    3079    4252    5995
B     =%000000           607     651     673     680     706     713     746     750     753     760     766     769     774
                         950     952    1076    1099    1266    1656    1666    1668    1673    1680    1689    1698    1706
                        1722    1799    1803    1807    1819    1886    1929    1938    1945    1954    1956    1959    2048
                        2054    2084    2102    2105    2147    2225    2235    2244    2250    2261    2359    2366    2380
                        2391    2499    2505    2594    2597    2870    2876    2904    3126    3140    3144    3150    3151
                        3153    3154    3170    3190    3313    3316    3378    3379    3390    3598    3657    3668    3690
                        3700    4359    4360    4369    4441    4449    4452    4475    4476    4515    4535    4550    4575
                        4577    4615    4622    4678    4679    4701    4707    4709    4712    4713    4726    4728    4732
                        4736    4746    4748    4749    4751    4756    4758    4762    4780    4786    4789    4791    4793
                        4802    4803    4804    4808    4809    4825    4826    4840    4847    4956    4964    4979    4982
                        4986    4990    5038    5042    5147    5159    5222    5237    5239    5294    5308    5325    5333
                        5336    5341    5350    5364    5370    5374    5381    5411    5418    5426    5429    5434    5452
                        5482    5483    5519    5526    5549    5592    5604    5608    5619    5635    5639    5647    5751
                        5759    5771    5785    5821    5825    5835    5838    5854    5857    5876    6102    6121    6143
                        6203    6204    6221    6232    6275    6281    6283    6463    6472    6473    6480
BBLOOP  011603          4284    4301#
BB1     017646          5705    6656#
BCA.ER  013371          4832    4990#
BCB.ER  013376          4988    4993#
BCCMD   012760          1570    4815#
BCC.ER  013356          4980    4984#
BC.ALP  013004          4826#   4844    4857
BC.BER  013352          4968    4981#
BC.BLP  013065          4864#   4868
BC.BL1  013101          4872#   4883
BC.CER  013342          4901    4910    4978#
BC.CLP  013153          4897#   4917
BC.DB2  013055          4853    4858#
BC.2ND  013062          4848    4863#
BC.3RD  013134          4887#
```

END.M80 CROSS REFERENCE TABLE

```
BELL  = 000007            89#   2198
BFCNT   020161          1810    1814    1848    1852    2192    6716#
BFOVR   002001          1355#   1813
BFRST   004160           697     924    1758    1908    2187#   3937    6276
BIT15 = 100000            66#   1484    1486    1488    1490    1493    1495    1497    1501    1504    1508    1510    1512
                        1515    1526    1574    1608
BLKADR  020112          4176    4204    6554    6696#
BLKNUM  020105          4178    4185    4206
617 84A DISCONNECTED 00:01:38:50 4095 337

@
```