```
                                5  **      HBUG - HEATH/WINTEK TERMINAL DEBUGGER.
                                6  *
                                7  *         J. G. L., 10/01/76, FOR *WINTEK* CORPORATION
                                8  *
                                9  *       COPYRIGHT 10/76, WINTEK CORPORATION,
                               10  *                        LAFAYETTE, INDIANA.
                               11  *
                               12  *          G. C.,       78.10
                               13  *                       79/12   --.05.--
                               14  *          W. Z.,       80/08   --.06.--
                               15  *
                               17  **      ASSEMBLY CONFIGURATION.
                               18
                               20
                               22  **      MACHINE INSTRUCTIONS.
                               23
000.303                        24  MI.JMP  EQU     11000011B       JMP
000.072                        25  MI.LDA  EQU     00111010B       LDA
000.327                        26  MI.BKP  EQU     11010111B       RST 2 (BREAKPOINT)
                               27
                               28  **      CHANNEL USED FOR LOAD/DUMP
                               29
000.005                        30  CN.LD   EQU     5               CHANNEL 5
                               31
                               32
000.000                        33          XTEXT   ASCII

                               35X **      ASCII CHARACTER EQUIVALENCES.
                               36X
000.015                        37X CR      EQU     13              CARRIAGE RETURN
000.012                        38X LF      EQU     10              LINE FEED
000.200                        39X NULL    EQU     200Q            PAD CHARACTER
000.000                        40X NUL2    EQU     0
000.007                        41X BELL    EQU     7               BELL CHARACTER
000.177                        42X RUBOUT  EQU     177Q
000.010                        43X BKSP    EQU     10Q             CTL-H
000.026                        44X C.SYN   EQU     26Q             SYNC
000.002                        45X C.STX   EQU     2               STX
000.047                        46X QUOTE   EQU     47Q
000.011                        47X TAB     EQU     11Q
000.033                        48X ESC     EQU     33Q
000.012                        49X NL      EQU     12Q             NEW LINE (HDOS SYSTEMS)
000.212                        50X ENL     EQU     NL+200Q         NL + END-OF-LINE-FLAG
000.014                        51X FF      EQU     14Q             FORM FEED
000.001                        52X CTLA    EQU     01Q             CTL-A
```

```
000.002                 53X CTLB    EQU     02Q             CTL-B
000.003                 54X CTLC    EQU     03Q             CTL-C
000.004                 55X CTLD    EQU     04Q             CTL-D
000.017                 56X CTLO    EQU     17Q             CTL-O
000.020                 57X CTLP    EQU     20Q             CTL-P
000.021                 58X CTLQ    EQU     21Q             CTL-Q
000.023                 59X CTLS    EQU     23Q             CTL-S
000.032                 60X CTLZ    EQU     32Q             CTL-Z
000.000                 61          XTEXT   HOSDEF

                        63X **      HOSDEF - DEFINE HOS PARAMETER.
                        64X *
                        65X
                        66X
000.040                 67X VERS    EQU     2*16+0          VERSION 2.0
                        68X
000.377                 69X SYSCALL EQU     377Q            SYSCALL INSTRUCTION
                        70X
                        71X
000.000                 72X         ORG     0
                        73X
                        74X *       RESIDENT FUNCTIONS
                        75X
000.000                 76X .EXIT   DS      1               EXIT (MUST BE FIRST)
000.001                 77X .SCIN   DS      1               SCIN
000.002                 78X .SCOUT  DS      1               SCOUT
000.003                 79X .PRINT  DS      1               PRINT
000.004                 80X .READ   DS      1               READ
000.005                 81X .WRITE  DS      1               WRITE
000.006                 82X .CONSL  DS      1               SET/CLEAR CONSOLE OPTIONS
000.007                 83X .CLRCO  DS      1               CLEAR CONSOLE BUFFER
000.010                 84X .LOADO  DS      1               LOAD AN OVERLAY
000.011                 85X .VERS   DS      1               RETURN HDOS VERSION NUMBER
000.012                 86X .SYSRES DS      1               PRECEDING FUNCTIONS ARE RESIDENT
                        87X
                        88X
                        89X *       *HDOSOVLO.SYS*  FUNCTIONS
                        90X
000.040                 91X         ORG     40A
                        92X
000.040                 93X .LINK   DS      1               LINK  (MUST BE FIRST)
000.041                 94X .CTLC   DS      1               CTL-C
000.042                 95X .OPENR  DS      1               OPENR
000.043                 96X .OPENW  DS      1               OPENW
000.044                 97X .OPENU  DS      1               OPENU
000.045                 98X .OPENC  DS      1               OPENC
000.046                 99X .CLOSE  DS      1               CLOSE
000.047                100X .POSIT  DS      1               POSITION
000.050                101X .DELET  DS      1               DELETE
000.051                102X .RENAM  DS      1               RENAME
000.052                103X .SETTP  DS      1               SETTOP
000.053                104X .DECODE DS      1               NAME DECODE
000.054                105X .NAME   DS      1               GET FILE NAME FROM CHANNEL
000.055                106X .CLEAR  DS      1               CLEAR CHAN
```

```
000.056                 107X .CLEARA DS      1          CLEAR ALL CHANS
000.057                 108X .ERROR  DS      1          LOOKUP ERROR
000.060                 109X .CHFLG  DS      1          CHANGE FLAGS
000.061                 110X .DISMT  DS      1          FLAG SYSTEM DISK DISMOUNTED
000.062                 111X .LOADD  DS      1          LOAD DEVICE DRIVER
000.063                 112X .OPEN   DS      1          Parametrized Open
                        113X
                        114X
                        115X *       *H0OSOVL1.SYS* FUNCTIONS
                        116X
000.200                 117X         ORG     200Q
                        118X
000.200                 119X .MOUNT  DS      1          MOUNT (MOST BE FIRST)
000.201                 120X .DMOUN  DS      1          DISMOUNT
000.202                 121X .MONMS  DS      1          MOUNT/NO MESSAGE
000.203                 122X .DMNMS  DS      1          DISMOUNT/NO MESSAGE
000.204                 123X .RESET  DS      1          RESET = DISMOUNT/MOUNT OF UNIT
000.205                 124X .CLEAN  DS      1          Clean device
000.206                 125X .DAD    DS      1          Dismount All Disks          /80.08.sc/
000.207                 126          XTEXT   MTR
```

```
                        129X **     MTR - PAM/8 EQUIVALENCES.
                        130X *
                        131X *      THIS DECK CONTAINS SYMBOLIC DEFINITIONS USED TO
                        132X *      MAKE USE OF THE PAM/8 CODE AND CONTROL BYTES.


                        134X **     IO PORTS
                        135X
    000.360             136X IP.PAD  EQU   360Q          PAD INPUT PORT
    000.360             137X OP.CTL  EQU   360Q          CONTROL OUTPUT PORT
    000.360             138X OP.DIG  EQU   360Q          DIGIT SELECT OUTPUT PORT
    000.361             139X OP.SEG  EQU   361Q          SEGMENT SELECT OUTPUT PORT
    000.362             140X IP.CON  EQU   362Q          H-88/H-89/HA-8-8 Configuration  /80.07.sc/
    000.362             141X OP2.CTL EQU   362Q          H-88/H-89/HA-8-8 Control Port   /80.07.sc/


                        143X **     FRONT PANEL CONTROL BITS.                          /80.07.sc/
                        144X *
                        145X *      CB.*  set in OP.CTL
                        146X *      CB2.* set in OP2.CTL
                        147X *
                        148X
    000.020             149X CB.SSI  EQU   00010000B     SINGLE STEP INTERRUPT
    000.040             150X CB.MTL  EQU   00100000B     MONITOR LIGHT
    000.100             151X CB.CLI  EQU   01000000B     CLOCK INTERRUPT ENABLE
    000.200             152X CB.SPK  EQU   10000000B     SPEAKER ENABLE
                        153X
    000.001             154X CB2.SSI EQU   00000001B     Single Step Interrupt
    000.002             155X CB2.CLI EQU   00000010B     Clock Interrupt Enable
    000.040             156X CB2.ORG EQU   00100000B     ORG.0 Select
    000.100             157X CB2.SID EQU   01000000B     Side 1 Select


                        159X **     Secondary Control Bits
                        160X


                        162X **     MONITOR MODE FLAGS.
                        163X
    000.000             164X DM.MR   EQU   0             MEMORY READ
    000.001             165X DM.MW   EQU   1             MEMORY WRITE
    000.002             166X DM.RR   EQU   2             REGISTER READ
    000.003             167X DM.RW   EQU   3             REGISTER WRITE
```

```
                        169X **     USER OPTION BITS.
                        170X *
                        171X *      THESE BITS ARE SET IN CELL .MFLAG.
                        172X
000.200                 173X UO.HLT EQU     10000000B       DISABLE HALT PROCESSING
000.100                 174X UO.NFR EQU     CB.CLI          NO REFRESH OF FRONT PANEL
000.002                 175X UO.DDU EQU     00000010B       DISABLE DISPLAY UPDATE
000.001                 176X UO.CLK EQU     00000001B       ALLOW PRIVATE INTERRUPT PROCESSING


                        178X **     MONITOR IDENTIFICATION FLAGS
                        179X *
                        180X *      THESE BYTES IDENTIFY THE ROM MONITOR.
                        181X *      THEY ARE THE VARIOUS VALUES OF LOCATION .IDENT
                        182X
000.021                 183X M.PAM8 EQU     021Q            'LXI' INSTRUCTION AT 000.000 IN PAM-8
000.303                 184X M.FOX  EQU     303Q            'JMP' INSTRUCTION AT 000.000 IN FOX ROM


                        186X **     Configuration Flags                                 /80:07:sc7
                        187X *
                        188X *      These bits are read in IP;CON:
                        189X *
                        190X
000.003                 191X CN.174M EQU    00000011B       Port 174Q Device-Type Mask
000.014                 192X CN.170M EQU    00001100B       Port 170Q Device-Type Mask
000.020                 193X CN.PRI  EQU    00010000B       Primary/Secondary:  1=>primary == 170Q
000.040                 194X CN.MEM  EQU    00100000B       Memory Test/Normal Switch:  0=>Test; 1=>Normal
000.100                 195X CN.BAU  EQU    01000000B       Baud Rate:  0=>9600; 1=>19,200
000.200                 196X CN.ABO  EQU    10000000B       Auto-Boot:  1=>Auto-Boot
                        197X
000.000                 198X CND.H17 EQU    00B             H-17 Disk;              Valid only in CN.174M
000.000                 199X CND.NDI EQU    00B             No Device Installed;  Valid only in CN.170M
000.001                 200X CND.H47 EQU    01B             H-47 Disk


                        202X **     ROUTINE ENTRY POINTS.
                        203X *
                        204X
000.000                 205X .IDENT  EQU    0000A           IDENTIFICATION LOCATION
000.053                 206X .DLY    EQU    0053A           DELAY
001.267                 207X .LOAD   EQU    1267A           TAPE LOAD
001.374                 208X .DUMP   EQU    1374A           TAPE DUMP
002.136                 209X .ALARM  EQU    2136A           ALARM ROUTINE
002.140                 210X .HORN   EQU    2140A           HORN
002.172                 211X .CTC    EQU    2172A           CHECK TAPE CHECKSUM
002.205                 212X .TPERR  EQU    2205A           TAPE ERROR ROUTINE
002.264                 213X .PCHL   EQU    2264A           PCHL INSTRUCTION
002.265                 214X .SRS    EQU    2265A           SCAN RECORD START
002.325                 215X .RNP    EQU    2325A           READ NEXT PAIR
002.331                 216X .RNB    EQU    2331A           READ NEXT BYTE
```

```
002,347                 217X .CRC     EQU     2347A           CRC-16 CALCULATOR
003,017                 218X .WNP     EQU     3017A           WRITE NEXT PAIR
003,024                 219X .WNB     EQU     3024A           WRITE NEXT BYTE
003,122                 220X .DOD     EQU     3122A           DECODE FOR OCTAL DISPLAY
003,260                 221X .RCK     EQU     3260A           READ CONSOLE KEYSET
003,356                 222X .DODA    EQU     3356A           SEGMENT CODE TABLE


                        224X **       RAM CELLS USED BY H8MTR.
                        225X *
                        226X
040,000                 227X .START   EQU     40000A          START DUMP ADDRESS
040,002                 228X .IOWRK   EQU     40002A          IN OR OUT INSTRUCTION
040,005                 229X .REGI    EQU     40005A          DISPLAYED REGISTER INDEX
040,006                 230X .DSPROT  EQU     40006A          PERIOD FLAG BYTE
040,007                 231X .DSPMOD  EQU     40007A          DISPLAY MODE
040,010                 232X .MFLAG   EQU     40010A          USER OPTION BYTE
040,011                 233X .CTLFLG  EQU     40011A          PANEL CONTROL BYTE
040,013                 234X .ALEDS   EQU     40013A          ABUSS LEDS
040,021                 235X .DLEDS   EQU     40021A          DBUSS LEDS
040,024                 236X .ABUSS   EQU     40024A          ABUSS REGISTER
040,027                 237X .CRCSUM  EQU     40027A          CRCSUM WORD
040,031                 238X .TPERRX  EQU     40031A          TAPE ERROR EXIT VECTOR
040,033                 239X .TICCNT  EQU     40033A          CLOCK TICK COUNTER
040,035                 240X .REGPTR  EQU     40035A          REGISTER POINTER
040,037                 241X .UIVEC   EQU     40037A          USER INTERRUPT VECTORS
040,064                 242X .NMIRET  EQU     40064A          H88/H89 NMI Return Address      /80.07.sc/
040,066                 243X .CTL2FL  EQU     40066A          OP2.CTL Control Byte            /80.07.sc/
000,207                 244           XTEXT   HOSEQU


                        246X **       HDOS SYSTEM EQUIVALENCES.
                        247X *
                        248X
024,000                 249X S.GRT0   EQU     24000A          SYSTEM AREA FOR  GRT0
025,000                 250X S.GRT1   EQU     25000A          SYSTEM AREA FOR  GRT1
026,000                 251X S.GRT2   EQU     26000A          SYSTEM AREA FOR  GRT2
                        252X
030,000                 253X ROMBOOT  EQU     30000A          ROM BOOT ENTRY
                        254X
040,100                 255X          ORG     40100A          FREE SPACE FROM PAM-8
                        256X
040,100                 257X          DS      8               JUMP TO SYSTEM EXIT
040,110                 258X D.CON    DS      16              DISK CONSTANTS
040,130                 259X SYDD     EQU     *               SYSTEM DISK ENTRY POINT
040,130                 260X D.VEC    DS      24*3            SYSTEM ROM ENTRY VECTORS
040,240                 261X D.RAM    DS      31              SYSTEM ROM WORK AREA
040,277                 262X S.VAL    DS      36              SYSTEM VALUES
040,343                 263X S.INT    DS      115             SYSTEM INTERNAL WORK AREAS
041,126                 264X          DS      16
041,146                 265X S.SOVR   DS      2               STACK OVERFLOW WARNING
041,150                 266X          DS      42200A-*        SYSTEM STACK
001,032                 267X STACKL   EQU     *-S.SOVR        STACK SIZE
```

```
                    268X
042.200             269X STACK   EQU     *               LWA+1 SYSTEM STACK
042.200             270X USERFWA EQU     *               USER FWA
042.200             271          XTEXT   ESVAL


                    273X **      S.VAL - SYSTEM VALUE DEFINTIONS.
                    274X *
                    275X *       THESE VALUES ARE SET AND MAINTAINED BY THE SYSTEM.
                    276X *
                    277X *       THE DECK HOSEQU MUST BE MODIFIED WHEN THIS IS MODIFIED.
                    278X
                    279X
040.277             280X         ORG     S.VAL
                    281X
040.277             282X S.DATE  DS      9               SYSTEM DATE (IN ASCII)
040.310             283X S.DATC  DS      2               CODED DATE
040.312             284X S.TIME  DS      4               TIME FROM MIDNIGHT (IN TICS)
040.316             285X S.HIMEM DS      2               HARDWARE HIGH MEMORY ADRESS+1
                    286X
040.320             287X S.SYSM  DS      2               FWA RESIDENT SYSTEM
                    288X
040.322             289X S.USRM  DS      2               LWA USER MEMORY
                    290X
040.324             291X S.OMAX  DS      2               MAX OVERLAY SIZE FOR SYSTEM
                    292X
                    293X
                    294X **      THE FOLLOWING FIVE CELLS SHOULD BE MODIFIED/READ ONLY VIA THE .CONSL SYSCALL
                    295X
000.200             296X CSL.ECH EQU     10000000B       SUPPRESS ECHO
000.004             297X CSL.RAW EQU     00000100B       Raw Mode I/O                    /80.09.sc/
000.002             298X CSL.WRP EQU     00000010B       WRAP LINES AT WIDTH
000.001             299X CSL.CHR EQU     00000001B       OPERATE IN CHARACTER MODE
                    300X
000.000             301X I.CSLMD EQU     0               S.CSLMD IS FIRST BYTE
040.326             302X S.CSLMD DS      1               CONSOLE MODE
                    303X
000.200             304X CTP.BKS EQU     10000000B       TERMINAL PROCESSES BACKSPACES
000.100             305X CTP.FF  EQU     01000000B       Terminal Processes Form-Feed    /80.09.sc/
000.040             306X CTP.MLI EQU     00100000B       MAP LOWER CASE TO UPPER ON INPUT
000.020             307X CTP.MLO EQU     00010000B       MAP LOWER CASE TO UPPER ON OUTPUT
000.010             308X CTP.2SB EQU     00001000B       TERMINAL NEEDS TWO STOP BITS
000.002             309X CTP.BKM EQU     00000010B       MAP BKSP (UPON INPUT) TO RUBOUT
000.001             310X CTP.TAB EQU     00000001B       TERMINAL SUPPORTS TAB CHARACTERS
                    311X
000.001             312X I.CONTY EQU     1               S.CONTY IS 2ND BYTE
000.000             313X         ERRNZ   *-S.CSLMD-I.CONTY
040.327             314X S.CONTY DS      1               CONSOLE TYPE FLAGS
000.002             315X I.CUSOR EQU     2               S.CUSOR IS 3RD BYTE
000.000             316X         ERRNZ   *-S.CSLMD-I.CUSOR
040.330             317X S.CUSOR DS      1               CURRENT CURSOR POSITION
000.003             318X I.CONWI EQU     3               S.CONWI IS 4TH BYTE
000.000             319X         ERRNZ   *-S.CSLMD-I.CONWI
040.331             320X S.CONWI DS      1               CONSOLE WIDTH
```

```
                    321X
000.001             322X CO.FLG  EQU     00000001B       CTL-O FLAG
000.200             323X CS.FLG  EQU     10000000B       CTL-S FLAG
                    324X
000.004             325X I.CONFL EQU     4               S.CONFL IS 5TH BYTE
000.000             326X         ERRNZ   *-S,CSLMD-I,CONFL
040.332             327X S.CONFL DS      1               CONSOLE FLAGS
                    328X
040.333             329X S.CAADR DS      2               ADDRESS FOR ABORT PROCESSING (>256 IF VALID)
040.335             330X S,CCTAB DS      6               ADDR FOR CTL-A, CTL-B, CTL-C PROCESSING
040.343             331          XTEXT   ECDEF


                    333X **      ERROR CODE DEFINITIONS.
                    334X
000.000             335X         ORG     0
000.000             336X         DS      1               NO ERROR #0
000.001             337X EC.EOF  DS      1               END OF FILE
000.002             338X EC.EOM  DS      1               END OF MEDIA
000.003             339X EC.ILC  DS      1               ILLEGAL SYSCALL CODE
000.004             340X EC.CNA  DS      1               CHANNEL NOT AVAILABLE
000.005             341X EC.DNS  DS      1               DEVICE NOT SUITABLE
000.006             342X EC.IDN  DS      1               ILLEGAL DEVICE NAME
000.007             343X EC.IFN  DS      1               ILLEGAL FILE NAME
000.010             344X EC.NRD  DS      1               NO ROOM FOR DEVICE DRIVER
000.011             345X EC.FNO  DS      1               CHANNEL NOT OPEN
000.012             346X EC.ILR  DS      1               ILLEGAL REQUEST
000.013             347X EC.FUC  DS      1               FILE USAGE CONFLICT
000.014             348X EC.FNF  DS      1               FILE NAME NOT FOUND
000.015             349X EC.UND  DS      1               UNKNOWN DEVICE
000.016             350X EC.ICN  DS      1               ILLEGAL CHANNEL NUMBER
000.017             351X EC.DIF  DS      1               DIRECTORY FULL
000.020             352X EC.IFC  DS      1               ILLEGAL FILE CONTENTS
000.021             353X EC.NEM  DS      1               NOT ENOUGH MEMORY
000.022             354X EC.RF   DS      1               READ FAILURE
000.023             355X EC.WF   DS      1               WRITE FAILURE
000.024             356X EC.WPV  DS      1               WRITE PROTECTION VIOLATION
000.025             357X EC.WP   DS      1               DISK WRITE PROTECTED
000.026             358X EC.FAP  DS      1               FILE ALREADY PRESENT
000.027             359X EC.DDA  DS      1               DEVICE DRIVER ABORT
000.030             360X EC.FL   DS      1               FILE LOCKED
000.031             361X EC.FAO  DS      1               FILE ALREADY OPEN
000.032             362X EC.IS   DS      1               ILLEGAL SWITCH
000.033             363X EC.UUN  DS      1               UNKNOWN UNIT NUMBER
000.034             364X EC.FNR  DS      1               FILE NAME REQUIRED
000.035             365X EC.DIW  DS      1               DEVICE IS NOT WRITABLE (OR WRITE LOCKED)
000.036             366X EC.UNA  DS      1               UNIT NOT AVAILABLE
000.037             367X EC.ILV  DS      1               ILLEGAL VALUE
000.040             368X EC.ILO  DS      1               ILLEGAL OPTION
000.041             369X EC.VPM  DS      1               VOLUME PRESENTLY MOUNTED ON DEVICE
000.042             370X EC.NVM  DS      1               NO VOLUME PRESENTLY MOUNTED
000.043             371X EC.FOD  DS      1               FILE OPEN ON DEVICE
000.044             372X EC.NPM  DS      1               NO PROVISIONS MADE FOR REMOUNTING MORE DISKS
000.045             373X EC.DNI  DS      1               DISK NOT INITIALIZED
000.046             374X EC.DNR  DS      1               DISK IS NOT READABLE
```

```
000.047                 375X EC.DSC  DS      1               DISK STRUCTURE IS CORRUPT
000.050                 376X EC.NCV  DS      1               NOT CORRECT VERSION OF HDOS
000.051                 377X EC.NOS  DS      1               NO OPERATING SYSTEM MOUNTED
000.052                 378X EC.IOI  DS      1               ILLEGAL OVERLAY INDEX
000.053                 379X EC.OTL  DS      1               OVERLAY TO LARGE
000.054                 380          XTEXT   FBDEF


                        382X **      FILE BLOCK DEFINITIONS.
                        383X
000.000                 384X         ORG     0
000.000                 385X FB.CHA  DS      1               CHANNEL NUMBER
000.001                 386X FB.FLG  DS      1               FLAGS
000.002                 387X FB.FWA  DS      2               BUFFER FWA
000.004                 388X FB.PTR  DS      2               BUFFER POINTER
000.006                 389X FB.LIM  DS      2               LIMIT OF DATA IN BUFFER (READ OPERATIONS)
000.010                 390X FB.LWA  DS      2               LWA OF BUFFER
000.012                 391X FB.NAM  DS      4+8+4+1         NAME OF FILE
000.021                 392X FB.NAML EQU     *-FB.NAM
000.033                 393X FBENL   EQU     *               ENTRY LENGTH
000.033                 394          XTEXT   FILDEF


                        396X **      FILDEF - FILE TYPE DEFINITIONS.
                        397X *
                        398X *       DB      377Q,FT.XXX
                        399X
                        400X
000.000                 401X FT.ABS  EQU     0               ABSOLUTE BINARY
000.001                 402X FT.PIC  EQU     1               POSITION INDEPENDANT CODE
000.002                 403X FT.REL  EQU     2               RELOCATABLE CODE
000.003                 404X FT.BAC  EQU     3               COMPILED BASIC CODE
000.033                 405          XTEXT   U8251
```

```
                        408X **       8251 USART BIT DEFINITIONS.
                        409X *
                        410X
                        411X **       PORT ADDRESSES
                        412X
000.000                 413X UDR      EQU     0               DATA REGISTER IS EVEN
000.001                 414X USR      EQU     1               STATUS REGISTER IS NEXT
                        415X
000.372                 416X SC.UART  EQU     372Q            CONSOLE USART ADDRESS (IFF 8251)
                        417X
                        418X
                        419X **       MODE INSTRUCTION CONTROL BITS.
                        420X
000.100                 421X UMI.1B   EQU     01000000B       1 STOP BIT
000.200                 422X UMI.HB   EQU     10000000B       1 1/2 STOP BITS
000.300                 423X UMI.2B   EQU     11000000B       2 STOP BITS
000.040                 424X UMI.PE   EQU     00100000B       EVEN PARITY
000.020                 425X UMI.PA   EQU     00010000B       USE PARITY
000.000                 426X UMI.L5   EQU     00000000B       5 BIT CHARACTERS
000.004                 427X UMI.L6   EQU     00000100B       6 BIT CHARACTERS
000.010                 428X UMI.L7   EQU     00001000B       7 BIT CHARACTERS
000.014                 429X UMI.L8   EQU     00001100B       8 BIT CHARACTERS
000.001                 430X UMI.1X   EQU     00000001B       CLOCK X 1
000.002                 431X UMI.16X  EQU     00000010B       CLOCK X 16
000.003                 432X UMI.64X  EQU     00000011B       CLOCK X 64
                        433X
                        434X **       COMMAND INSTRUCTION BITS.
                        435X
000.100                 436X UCI.IR   EQU     01000000B       INTERNAL RESET
000.040                 437X UCI.RO   EQU     00100000B       READER-ON CONTROL FLAG
000.020                 438X UCI.ER   EQU     00010000B       ERROR RESET
000.004                 439X UCI.RE   EQU     00000100B       RECEIVE ENABLE
000.002                 440X UCI.IE   EQU     00000010B       ENABLE INTERRUPTS FLAG
000.001                 441X UCI.TE   EQU     00000001B       TRANSMIT ENABLE
                        442X
                        443X **       STATUS READ COMMAND BITS.
                        444X
000.100                 445X USR.BD   EQU     01000000B       Break Detect                    /80.08.sc/
000.040                 446X USR.FE   EQU     00100000B       FRAMING ERROR
000.020                 447X USR.OE   EQU     00010000B       OVERRUN ERROR
000.010                 448X USR.PE   EQU     00001000B       PARITY ERROR
000.004                 449X USR.TXE  EQU     00000100B       TRANSMITTER EMPTY
000.002                 450X USR.RXR  EQU     00000010B       RECEIVER READY
000.001                 451X USR.TXR  EQU     00000001B       TRANSMITTER READY
000.033                 452           XTEXT   ABSDEF



                        454X **       ABS FORMAT EQUIVALENCES.
                        455X
000.000                 456X          ORG     0
                        457X
000.000                 458X ABS.ID   DS      1               377Q = BINARY FILE FLAG
000.001                 459X          DS      1               FILE TYPE (FT.ABS)
000.002                 460X ABS.LDA  DS      2               LOAD ADDRESS
000.004                 461X ABS.LEN  DS      2               LENGTH OF ENTIRE RECORD
```

```
000.006                 462X ABS.ENT DS      2              ENTRY POINT
                        463X
000.010                 464X ABS.COD DS      0              CODE STARTS HERE
000.010                 465          XTEXT   PICDEF

                        467X **      PIC FORMAT EQUIVALENCES.
                        468X
000.000                 469X         ORG     0
                        470X
000.000                 471X PIC.ID  DS      1              377Q = BINARY FILE FLAG
000.001                 472X         DS      1              FILE TYPE (FT.PIC)
000.002                 473X PIC.LEN DS      2              LENGTH OF ENTIRE RECORD
000.004                 474X PIC.PTR DS      2              INDEX OF START OF PIC TABLE
                        475X
000.006                 476X PIC.COD DS      0              CODE STARTS HERE
000.006                 477          XTEXT   DIRDEF

                        479X **      DIRECTORY ENTRY FORMAT.
                        480X
000.000                 481X         ORG     0
                        482X
                        483X
000.377                 484X DF.EMP  EQU     377Q           FLAGS ENTRY EMPTY
000.376                 485X DF.CLR  EQU     376Q           FLAGS ENTRY EMPTY, REST OF DIR ALSO CLEAR
                        486X
000.000                 487X DIR.NAM DS      8              NAME
000.010                 488X DIR.EXT DS      3              EXTENSION
000.013                 489X DIR.PRO DS      1              PROJECT
000.014                 490X DIR.VER DS      1              VERSION
000.015                 491X DIRIDL  EQU     *              FILE IDENTIFICATION LENGTH
                        492X
000.015                 493X DIR.CLU DS      1              CLUSTER FACTOR
000.016                 494X DIR.FLG DS      1              FLAGS
000.017                 495X         DS      1              RESERVED
000.020                 496X DIR.FGN DS      1              FIRST GROUP NUMBER
000.021                 497X DIR.LGN DS      1              LAST GROUP NUMBER
000.022                 498X DIR.LSI DS      1              LAST SECTOR INDEX (IN LAST GROUP)
000.023                 499X DIR.CRD DS      2              CREATION DATE
000.025                 500X DIR.ALD DS      2              LAST ALTERATION DATE
                        501X
000.027                 502X DIRELEN EQU     *              DIRECTORY ENTRY LENGTH
000.027                 503          XTEXT   IOCDEF
```

```
                        505X **      I/O CHANNEL DEFINITIONS.
                        506X
000.000                 507X         ORG     0
                        508X
000.000                 509X IOC.LNK DS      2          ADDRESS OF NEXT CHANNEL, =0 IF LAST
000.002                 510X IOC.DDA DS      2          THREAD JUMP TO DEVICE DRIVER (VIA DEV TABLE)
                        511X
000.004                 512X IOC.FLG DS      1          FILE TYPE FLAGS
000.001                 513X FT.DD   EQU     00000001B  =1 IF DIRECTORY DEVICE
000.002                 514X FT.OR   EQU     00000010B  =1 IF OPEN FOR READ
000.004                 515X FT.OW   EQU     00000100B  =1 IF OPEN FOR WRITE
000.010                 516X FT.OU   EQU     00001000B  =1 IF OPEN FOR UPDATE
000.020                 517X FT.OC   EQU     00010000B  =1 IF OPEN FOR CHARACTER MODE   /80.02.GC/
000.003                 518X IOC.SQL EQU     *-IOC.DDA  LENGTH OF INFO FOR SEQUENTIAL FILE (FROM IOC)
                        519X
000.005                 520X IOC.GRT DS      2          ADDRESS OF GROUP RESERVATION TABLE
000.007                 521X IOC.SPG DS      1          SECTORS PER GROUP, THIS DEVICE
000.010                 522X IOC.CGN DS      1          CURRENT GROUP NUMBER
000.011                 523X IOC.CSI DS      1          CURRENT SECTOR INDEX (IN CURRENT GROUP)
000.012                 524X IOC.LGN DS      1          LAST GROUP NUMBER
000.013                 525X IOC.LSI DS      1          LAST SECTOR INDEX (IN LAST GROUP)
000.010                 526X IOC.DRL EQU     *-IOC.FLG  LENGTH OF INFO NORMALLY COPIED BACK TO
                        527X *                          THE CHANNEL TABLE
000.014                 528X IOC.DTA DS      2          DEVICE TABLE ADDRESS FOR THIS DEVICE
000.016                 529X IOC.DES DS      2          SECTOR NUMBER OF DIRECTORY ENTRY
000.020                 530X IOC.DEV DS      2          DEVICE CODE
000.022                 531X IOC.UNI DS      1          UNIT NUMBER (0-9)
000.021                 532X IOC.DIL EQU     *-IOC.DDA  LENGTH OF INFO FOR DIRECTORY FILE (FROM IOC)
                        533X
000.023                 534X IOC.DIR DS      DIRELEN    DIRECTORY ENTRY
                        535X
000.052                 536X IOCELEN EQU     *          IOC ENTRY LENGTH
                        537X
000.001                 538X IOCCTD  EQU     1          INDEX OF USER CHANNEL #0 IN CHANTAB (FIRST = 0)


042.170                 540          ORG     USERFWA-ABS.COD
042.170  377.000      541          DB      377Q,FT.ABS
042.172  200 042      542          DW      USERFWA    LOAD
042.174  201 015      543          DW      MEML-USERFWA  SIZE
042.176  353 057      544          DW      PRS        ENTRY
                        545
```

```
                            549  **      CMD - COMMAND COMPLETION PROCESSOR.
                            550  *
                            551  *       (H,L) = COMMAND STRING ADDRESS
                            552  *       (B,C) = CONTROL CARD ADDRESS
                            553
                            554
042.200                     555  CCP     EQU     *               ENTRY
                            556
042.200  076 072            557          MVI     A,MI.LDA
042.202  062 217 044        558          STA     FICA            READ CHARACTERS FROM BUFFER
042.205  041 312 044        559  CMD1    LXI     H,LINE
042.210  042 020 045        560          SHLD    LINPTR
                            561
                            562  *       INPUT 1 CHARACTER
                            563
042.213  315 143 053        564  CMD2    CALL    $INCHA          READ ONE CHARACTER
042.216  315 113 053        565          CALL    $MCU            MAP TO UPPER CASE
042.221  376 004            566          CPI     CTLD
042.223  312 001 046        567          JE      EXIT            IS EXIT
                            568
                            569  *       ADD TEMPORARILY TO LINE
                            570
042.226  052 020 045        571  CMD3    LHLD    LINPTR
042.231  167                572          MOV     M,A             RTORE IN LINE
042.232  043                573          INX     H
042.233  257                574          XRA     A
042.234  167                575          MOV     M,A             FOLLOW WITH 00
                            576
                            577  *       CLEAR NXTCHA, PATCNT
                            578
042.235  041 000 377        579          LXI     H,377000A
042.240  042 306 044        580          SHLD    NXTCHA
                            581
042.243  041 232 056        582          LXI     H,CMDTAB
042.246  042 310 044        583          SHLD    CMDADR
                            584
                            585  *       CHECK AGAINS NEXT COMMAND DESCRIPTION.
                            586
042.251  041 022 045        587  CMD4    LXI     H,CMD.8A
042.254  006 057            588          MVI     B,CMD.TL-CMD.8A (B) = BYTE COUNT
042.256  315 212 031        589          CALL    $ZERO           ZERO TABLES
042.261  041 307 044        590          LXI     H,PATCNT
042.264  064                591          INR     M
042.265  043                592          INX     H
042.266  136                593          MOV     E,M
042.267  043                594          INX     H
042.270  126                595          MOV     D,M             (D,E) = ADDRESS OF LAST COMMAND
042.271  315 275 044        596          CALL    SRC             SCAN FOR NEXT COMMAND
042.274  162                597          MOV     M,D
042.275  053                598          DCX     H
042.276  163                599          MOV     M,E             REPLACE CMDADR
042.277  001 312 044        600          LXI     B,LINE          (BC) = ADDRESS OF INPUT CHARACTER
042.302  032                601          LDAX    D
042.303  247                602          ANA     A
042.304  302 325 042        603          JNZ     CMD5            HAVE COMMAND ELEMENT
                            604
```

```
                                605  *       NO MORE COMMANDS. HAVE:
                                606  *
                                607  *       1) NO MATCHES, OR
                                608  *       2) A UNIQUE NEXT CHARACTER
                                609
042.307  072 306 044           610          LDA     NXTCHA
042.312  247                   611          ANA     A
042.313  302 226 042           612          JNZ     CMD3            (A) = AUTO GENERATED CHARACTER
042.316  315 060 054           613          CALL    $TYPCH
042.321  007                   614          DB      7               BELL
042.322  303 213 042           615          JMP     CMD2            READ FROM CONSOLE
                                616
                                617  *       CHECK NEXT TABLE ELEMENT FOR MATCH
                                618
042.325  012                   619  CMD5    LDAX    B               (A) = NEXT LINE CHARACTER
042.326  247                   620          ANA     A
042.327  302 376 042           621          JNZ     CMD7            IF SOME
                                622
                                623  *       NO MORE TEXT. SEE IF CAN ANTICIPATE NEXT CHARACTER
                                624
042.332  032                   625          LDAX    D               (A) = COMMAND ELEMENT
042.333  376 300               626          CPI     0C0H
042.335  312 376 042           627          JE      CMD7            PROCESS STRING RETURNS
042.340  315 061 044  CMD6     628          CALL    AEC             ACCEPT ENTERED COMMAND
042.343  376 012               629          CPI     NL
042.345  312 213 042           630          JE      CMD2            CANNOT COMPLETE CARRIAGE-RETURN
042.350  247                   631          ANA     A
042.351  310                   632          RZ                      EXIT IF ENTIRE COMMAND MATCHED
042.352  372 213 042           633          JM      CMD2            CANNOT COMPLETE
042.355  041 306 044           634          LXI     H,NXTCHA
                                635
                                636  *       SEE IF THIS IS THE FIRST COMPLETION CHARACTER,
                                637  *       OR IF IT IS THE SAME CHARACTER AS PREVIOUSLY FOUND
                                638
                                639
042.360  276                   640          CMP     M
042.361  312 251 042           641          JE      CMD4            SAME AS PREVIOUS, CAN COMPLETE
042.364  127                   642          MOV     D,A
042.365  206                   643          ADD     M
042.366  167                   644          MOV     M,A
042.367  272                   645          CMP     D               SEE IF NXTCHA WAS 0
042.370  312 251 042           646          JE      CMD4            CAN COMPLETE
042.373  303 213 042           647          JMP     CMD2            CANNOT COMPLETE
                                648
                                649  *       HAVE PATTERN AND TEXT. SEE IF MATCH.
                                650
042.376  325                   651  CMD7    PUSH    D
042.377  041 000 000           652          LXI     H,0
043.002  071                   653          DAD     SP
043.003  042 304 044           654          SHLD    STKPTR          SAVE STACKPOINTER
043.006  041 053 043           655          LXI     H,CMD.NG
043.011  345                   656          PUSH    H               SET 'CMD.NG' AS RETURN ADDRESS
043.012  032                   657          LDAX    D
043.013  147                   658          MOV     H,A             (H) = NEXT REQUIRED CHARACTER
043.014  007                   659          RLC                     (A) = PATTERN ELEMENT
043.015  332 027 043           660          JC      CMD8            IS COMPLEX ELEMENT
```

```
043.020  012            661          LDAX    B               (A) = NEXT TEXT ELEMENT
043.021  003            662          INX     B               ASSUME MATCH
043.022  274            663          CMP     H
043.023  300            664          RNE                     TO CMD.NG IF BAD
043.024  303 045 043    665          JMP     CMD.OK          GOOD
                        666
                        667  *       HAVE COMPLEX PATTERN ELEMENT
                        668
043.027  007            669  CMD8    RLC
043.030  007            670          RLC
043.031  007            671          RLC
043.032  346 007        672          ANI     7
043.034  315 076 031    673          CALL    $TBRA           BRANCH TO PROCESSOR
                        674
                        675  **      SPECIAL PATTERN ELEMENT TABLE.
                        676
043.037  036            677          DB      CMD.8-*         ENCLOSURE
043.040  106            678          DB      CMD.9-*         STRING CALL
043.041  132            679          DB      CMD.A-*         OCTAL ADDRESS
043.042  150            680          DB      CMD.B-*         FILE NAME
043.043  233            681          DB      CMD.C-*         STRING RETURN
043.044  241            682          DB      CMD.D-*         ADDRESS LIST


                        684  **      COMPLEX ROUTINES RETURN TO THESE THREE POINTS:
                        685  **
                        686
                        687
                        688  **      CMD.OK - NORMAL EXIT
                        689
043.045  023            690  CMD.OK  INX     D
043.046  341            691  CMD.OK. POP     H
043.047  341            692          POP     H
043.050  303 325 042    693          JMP     CMD5
                        694
                        695
                        696  **      CMD.NG - MATCH NO GOOD.
                        697
043.053  052 304 044    698  CMD.NG  LHLD    STKPTR
043.056  371            699          SPHL
043.057  321            700          POP     D
043.060  303 251 042    701          JMP     CMD4            TRY NEXT COMMAND
                        702
                        703
                        704  **      CMD.RA - RAN OUT OF TEXT WHILE MATCHING A COMPLEX
                        705  *       ELEMENT.
                        706  *
                        707  *       (A) = NEXT ELEMENT NEEDED
                        708
043.063  052 304 044    709  CMD.RA  LHLD    STKPTR
043.066  371            710          SPHL
043.067  341            711          POP     H
043.070  076 200        712          MVI     A,200Q          DONT ALLOW ANY COMPLETION
043.072  303 340 042    713          JMP     CMD6
```

```
                                717  **      CMD8 - PROCESS OPTION STRINGS.
                                718  *
                                719  *       1000                    8-CODE
                                720  *            NNN                TARGET INDEX
                                721  *               F               FLAG
                                722  *
                                723  *                F = 0, MAY MATCH ONE
                                724  *                F = 1, MUST MATCH ONE
                                725
                                726
043.075                         727  CMD.8   EQU     *
043.075  012                    728          LDAX    B               (A) = TEXT CHARACTER
043.076  147                    729          MOV     H,A             (H) = TEXT CHARACTER
043.077  032                    730          LDAX    D
043.100  157                    731          MOV     L,A             (L) = 8X FLAG
043.101  023                    732  CMD.81  INX     D
043.102  032                    733          LDAX    D               (A) = NEXT PATTERN CHARACTER
043.103  274                    734          CMP     H
043.104  312 121 043            735          JE      CMD.82          IF GOT A MATCH
043.107  007                    736          RLC
043.110  322 101 043            737          JNC     CMD.81          NOT FINISHED YET
                                738
                                739  *       NO MATCH
                                740
043.113  175                    741          MOV     A,L             (A) = 8X CODE
043.114  017                    742          RRC
043.115  330                    743          RC                      REQUIRE MATCH - EXIT TO CMD.NG
043.116  303 045 043            744          JMP     CMD.OK          ACCEPT
                                745
                                746  *       HAVE MATCH
                                747
043.121  175                    748  CMD.82  MOV     A,L             (A) = 8X CODE
043.122  017                    749          RRC
043.123  346 007                750          ANI     7
043.125  306 022                751          ADI     #CMD.8A
043.127  157                    752          MOV     L,A
043.130  174                    753          MOV     A,H             (A) = TEXT CHARACTER
043.131  046 045                754          MVI     H,CMD.8A/256
043.133  167                    755          MOV     M,A
043.134  003                    756          INX     B
                                757
                                758  *       SKIP REMAINDER OF OPTIONS
                                759
043.135  023                    760  CMD.83  INX     D
043.136  032                    761          LDAX    D               CHECK TEXT PATTERN CHARACTER
043.137  007                    762          RLC
043.140  322 135 043            763          JNC     CMD.83          IF NOT TERMINATOR
043.143  303 045 043            764          JMP     CMD.OK          EXIT FOUND
```

```
                                766  **      CMD.9 - STRING CALL
                                767  *
                                768  *       1001                    9 CODE
                                769  *          NNNN   STRING NUMBER
                                770
                                771
043.146  032                   772  CMD.9    LDAX    D               (A) = 9X MODE
043.147  353                   773           XCHG
043.150  042 026 045           774           SHLD    CMD.9A          SAVE RETURN ADDRESS
043.153  021 020 057           775           LXI     D,CMDEXS        POINT TO EXTENSION STRING
043.156  346 017               776           ANI     17Q
043.160  157                   777           MOV     L,A
043.161  315 275 044           778  CMD.91   CALL    SRC             SKIP REMAINDER OF COMMAND STRING
043.164  055                   779           DCR     L
043.165  302 161 043           780           JNZ     CMD.91          IF MORE
043.170  303 046 043           781           JMP     CMD.OK.         DONE


                                783  **      CMD.A - OCTAL ADDRESS.
                                784  *
                                785  *       NO DEFAULTING IS ALLOWED.
                                786  *       THE ADDRESS MAY BE FOLLOWED BY A MODIFIER
                                787  *        AAAAAA(NNN)
                                788  *
                                789  *       1010                    A CODE
                                790  *           NN  VALUE INDEX
                                791  *              F                =1 IF NO DEFAULT ALLOWED
                                792  *               F               =1 IF NO /LEN ALLOWED
                                793  *
                                794
                                795
043.173  032                   796  CMD.A    LDAX    D               (A) = FLAG
043.174  346 014               797           ANI     14Q
043.176  041 030 045           798           LXI     H,CMD.AA
043.201  315 072 030           799           CALL    $DADA           (HL) = ADDRESS OF STORE AREA
043.204  315 112 044           800           CALL    DAS             DECODE ADDRESS SPECIFICATION
043.207  303 045 043           801           JMP     CMD.OK          IS OK


                                803  **      CMD.B - FILE NAME
                                804  *
                                805  *       VALID HDOS FILE NAME
                                806  *
                                807  *       1011                    B CODE
                                808  *            0000               NO SPECIFICATION
                                809
                                810
043.212  315 250 043           811  CMD.B    CALL    CMD.B5          EXAMINE NEXT CHARACTER
043.215  330                   812           RC                      NOT GOOD 1ST CHARACTER
043.216  003                   813           INX     B               ADVANCE POINTER
043.217  041 222 057           814           LXI     H,CMD.BA        (HL) = WORK AREA
043.222  167                   815           MOV     M,A             STORE 1ST CHARACTER
```

```
043.223  043            816          INX     H
043.224  315 250 043     817  CMD.B1  CALL    CMD.B5          GET NEXT CHARACTER
043.227  332 243 043     818          JC      CMD.B2          NOT PART OF FILE NAME
043.232  003             819          INX     B               ADVANCE POINTER
043.233  167             820          MOV     M,A             STORE
043.234  043             821          INX     H
043.235  076 243         822          MVI     A,#CMD.BA+FB.NAML
043.237  275             823          CMP     L
043.240  302 224 043     824          JNE     CMD.B1          NOT JUST LONG ENOUGH
                         825
                         826  *       NAME GATHERED.
                         827
043.243  066 000         828  CMD.B2  MVI     M,0             FLAG END OF NAME
043.245  303 045 043     829          JMP     CMD.OK          EXIT

                         831  **      CMD.B5 - EXAMINE NEXT CHARACTER FOR VALIDITY
                         832  *
                         833  *       ENTRY   (BC) = CHARACTER ADDRESS
                         834  *       EXIT    'C' CLEAR IF CHARACTER VALID (0-9, A-Z, : OR .)
                         835  *               'C' SET IF CHARACTER INVALID
                         836  *       USES    A,F
                         837
                         838
043.250  012             839  CMD.B5  LDAX    B
043.251  376 056         840          CPI     '.'
043.253  330             841          RC                      TOO SMALL
043.254  310             842          RE                      IS .
043.255  376 072         843          CPI     ':'
043.257  310             844          RE                      IS :
043.260  376 060         845          CPI     '0'
043.262  330             846          RC                      NOT DIGIT
043.263  376 072         847          CPI     '9'+1
043.265  077             848          CMC
043.266  320             849          RNC                     IS DIGIT
043.267  376 101         850          CPI     'A'
043.271  330             851          RC                      NOT ALPHA
043.272  376 133         852          CPI     'Z'+1
043.274  077             853          CMC
043.275  311             854          RET                     RETURN WITH VERDICT

                         856  **      CMD.C - STRING RETURN
                         857  *
                         858  *       1100                    C FLAG
                         859  *          0000
                         860
                         861
043.276  052 026 045     862  CMD.C   LHLD    CMD.9A
043.301  353             863          XCHG
043.302  303 045 043     864          JMP     CMD.OK          EXIT
```

```
                                866  **      CMD.D - ADDRESS LIST
                                867  *
                                868  *       ADDR[(CNT)][,...,ADDR[(CNT)]]
                                869  *
                                870  *       NONE MAY BE NULL.
                                871
                                872
043.305  041 040 045           873  CMD.D   LXI     H,CMD.DA
043.310  325                   874  CMD.D1  PUSH    D
043.311  021 352 043           875          LXI     D,CMD.DB        POINT TO FLAG CHARACTER
043.314  315 112 044           876          CALL    DAS
                                877
                                878  *       WAS OK, SEE IF MORE TEXT FOLLOWS.
                                879  *
                                880  *       IF ',', TAKE IT AND PROCESS NEXT ADDRESS
                                881  *       IF NL, EXIT WITH MATCH
                                882  *       IF NULL, REQUIRE A ','
                                883  *       ELSE ERROR
                                884
043.317  321                   885          POP     D
043.320  043                   886          INX     H
043.321  076 100               887          MVI     A,#CMD.DA2
043.323  275                   888          CMP     L               'Z' SET IF ENOUGH VALUES READ
043.324  012                   889          LDAX    B
043.325  003                   890          INX     B
043.326  312 336 043           891          JE      CMD.D2          IF ALREADY READ ENOUGH VALUES
043.331  376 054               892          CPI     ','
043.333  312 310 043           893          JE      CMD.D1          DECODE NEXT ADDRESS
043.336  066 003               894  CMD.D2  MVI     M,3             SET DEFAULT FLAG FOR LAST+1 VALUE
043.340  376 012               895          CPI     NL
043.342  312 045 043           896          JE      CMD.OK          COMMAND COMPLETE, ACCEPT
043.345  247                   897          ANA     A
043.346  300                   898          RNZ                     IS NOT NULL? ILLEGAL
043.347  303 063 043           899          JMP     CMD.RA          RUN OUT
                                900
043.352  242                   901  CMD.DB  DB      0A2H
```

```
                                905  **      ACN - ACCUMULATE NUMBER.
                                906  *
                                907  *       ACN ACCUMULATES A N-DIGIT NUMBER
                                908  *
                                909  *       ENTRY   (B,C) = TEXT ADDRESS
                                910  *               (A) = NUMBER OF DIGITS
                                911  *               (D) = BASE
                                912  *       EXIT    (D,E) = VALUE
                                913  *               'Z' FLAG SET OF 0 DIGITS
                                914  *               (A) = NZ IF OVERFLOW
                                915
                                916
 043.353  345                   917  ACN     PUSH    H               SAVE (H,L)
 043.354  365                   918          PUSH    PSW
 043.355  041 000 000           919          LXI     H,0             (H,L) = ACCUMULATOR
 043.360  134                   920          MOV     E,H             (E) = OVERFLOW FLAG
 043.361  365                   921  ACN1    PUSH    PSW
 043.362  315 217 044           922          CALL    FIC
 043.365  326 060               923          SUI     '0'
 043.367  332 027 044           924          JC      ACN2            NOT DIGIT
 043.372  272                   925          CMP     D
 043.373  322 027 044           926          JNC     ACN2            TOO LARGE
 043.376  365                   927          PUSH    PSW             SAVE DIGIT VALUE
 043.377  325                   928          PUSH    D               SAVE BASE AND OVERFLOW FLAG
 044.000  172                   929          MOV     A,D             (A) = BASE
 044.001  353                   930          XCHG                    (DE) = ACCUMULATOR
 044.002  315 007 031           931          CALL    $MU86           (HL) = ACCUMULATOR*BASE
 044.005  321                   932          POP     D               RESTORE (DE)
 044.006  203                   933          ADD     E               ACCUMULATE OVERFLOWS
 044.007  137                   934          MOV     E,A             (E) = OVERFLOW INDICATOR
 044.010  361                   935          POP     PSW
 044.011  315 072 030           936          CALL    $DADA           (HL) = ACCUMULATOR*BASE+DIGIT
 044.014  173                   937          MOV     A,E
 044.015  316 000               938          ACI     0
 044.017  137                   939          MOV     E,A             ACCUMULATE OVERFLOWS
 044.020  361                   940          POP     PSW             (A) = COUNT
 044.021  075                   941          DCR     A
 044.022  302 361 043           942          JNZ     ACN1            IF MORE TO GO
 044.025  365                   943          PUSH    PSW
 044.026  003                   944          INX     B
                                945
                                946  *       GOT ALL DIGITS
                                947
 044.027  013                   948  ACN2    DCX     B
                                949
                                950  *       IF BASE = 8, SHIFT TOP HALF RIGHT TO MAKE UP
                                951  *       FOR DIGIT 2, WHICH CONTAINS ONLY 2 DIGITS.
                                952
 044.030  076 010               953          MVI     A,8
 044.032  272                   954          CMP     D
 044.033  302 051 044           955          JNE     ACN3            NOT OCTAL
 044.036  173                   956          MOV     A,E             (A) = OVERFLOW
 044.037  037                   957          RAR
 044.040  137                   958          MOV     E,A             (E) = BITS 1-7 OF OVERFLOW
 044.041  174                   959          MOV     A,H
 044.042  037                   960          RAR
```

```
044.043  147           961          MOV     H,A
044.044  076 000       962          MVI     A,0
044.046  213           963          ADC     E               ADD OVERFLOW FROM SHIFT
044.047  213           964          ADC     E
044.050  137           965          MOV     E,A
044.051  361           966  ACN3    POP     PSW             (A) = ORIGINAL DIGIT COUNT
044.052  127           967          MOV     D,A             (D) = COUNT
044.053  361           968          POP     PSW
044.054  272           969          CMP     D
044.055  173           970          MOV     A,E             (A) = CARRY FLAG
044.056  353           971          XCHG                    (DE) = RESULT
044.057  341           972          POP     H
044.060  311           973          RET                     RETURN


                       975  **      AEC - ACCEPT ECHOED CHARACTER.
                       976  *
                       977  *       AEC ACCEPTS AND ECHOS THE ENTERED CHARACTER.
                       978
                       979
044.061  305           980  AEC     PUSH    PSW
044.062  052 020 045   981          LHLD    LINPTR
044.065  176           982          MOV     A,M
044.066  247           983          ANA     A
044.067  312 110 044   984          JZ      AEC1            IF ALREADY TYPED
044.072  315 064 054   985          CALL    $TYPC.          TYPE IT
044.075  043           986          INX     H
044.076  066 000       987          MVI     M,0
044.100  042 020 045   988          SHLD    LINPTR
044.103  376 012       989          CPI     NL
000.000                990          ERRNZ   LF-NL           TWO CHARACTER MATCH
                       991  *       MVI     A,LF            ASSUME CR
044.105  314 064 054   992          CE      $TYPC.          IF CR, ECHO CRLF
044.110  361           993  AEC1    POP     PSW
044.111  311           994          RET                     EXIT


                       996  **      DAS - DECODE ADDRESS SPECIFICATION.
                       997  *
                       998  *       ENTRY   ((HL)) = VALUE BLOCK
                       999  *               ((DE)) = PATTERN CODE
                      1000  *       EXIT    TO CMD.NG IF BAD
                      1001  *               RETURNS IF OK
                      1002
                      1003
044.112  325           1004 DAS     PUSH    D
044.113  032           1005         LDAX    D
044.114  365           1006         PUSH    PSW             SAVE CODE
044.115  076 006       1007         MVI     A,6             (A) = MAX DIGITS
044.117  026 010       1008         MVI     D,8             (D) = BASE
044.121  315 353 043   1009         CALL    ACN             ACCUMULATE NUMBER
044.124  066 000       1010         MVI     M,0
```

```
044.126  302 151 044  1011          JNZ     DAS1            NOT DEFAULTED
044.131  361         1012          POP     PSW             (A) = OPTION FLAG
044.132  365         1013          PUSH    PSW
044.133  017         1014          RRC
044.134  017         1015          RRC
044.135  332 053 043  1016          JC      CMD.NG          DEFAULT NOT ALLOWED
                     1017
                     1018  *       HAVE NON-NUMERIC, IS EITHER DEFAULT (NULL) OR #
                     1019
044.140  064         1020          INR     M               ASSUME NULL
044.141  012         1021          LDAX    B
044.142  326 043     1022          SUI     '#'
044.144  302 155 044  1023          JNE     DAS2            NOT #, IS NULL
044.147  003         1024          INX     B
044.150  064         1025          INR     M
044.151  247         1026  DAS1    ANA     A               CHECK CARRY
044.152  302 053 043  1027          JNZ     CMD.NG          OVERFLOW
044.155  043         1028  DAS2    INX     H
044.156  163         1029          MOV     M,E
044.157  043         1030          INX     H
044.160  162         1031          MOV     M,D
044.161  043         1032          INX     H               (HL) = ADDRESS OF COUNT FIELD
044.162  066 001     1033          MVI     M,1             ASSUME 1
044.164  361         1034          POP     PSW
044.165  321         1035          POP     D
044.166  017         1036          RRC
044.167  330         1037          RC                      IF COUNT NOT ALLOWED
                     1038
                     1039  *       SEE IF /CNT FOLLOWS
                     1040
044.170  012         1041          LDAX    B
044.171  376 057     1042          CPI     '/'
044.173  300         1043          RNE                     IF NONE
044.174  325         1044          PUSH    D
044.175  003         1045          INX     B
044.176  076 003     1046          MVI     A,3
044.200  026 012     1047          MVI     D,10
044.202  315 353 043  1048          CALL    ACN             ACCUMULATE DECIMAL NUMBER
044.205  312 053 043  1049          JZ      CMD.NG          IF NONE
044.210  262         1050          ORA     D
044.211  302 053 043  1051          JNZ     CMD.NG          IF OVERFLOW
044.214  163         1052          MOV     M,E             SAVE VALUE
044.215  321         1053          POP     D
044.216  311         1054          RET                     IS OK ELEMENT


                     1056  **      FIC - FETCH INPUT CHARACTER.
                     1057  *
                     1058  *       FIC IS CALLED TO GET THE NEXT INPUT CHARACTER.
                     1059  *
                     1060  *       ENTRY   (B,C) = INPUT POINTER
                     1061
                     1062
044.217              1063  FIC     EQU     *
```

```
044.217                    1064  FICA    EQU     *               TOGGLE FLAG
044.217  303 235 044       1065          JMP     FIC2            NO-OP'ED IF TO READ FROM MEMORY
044.222  012               1066          LDAX    B
044.223  247               1067          ANA     A
044.224  312 063 043       1068          JZ      CMD.RA          IF NONE
044.227  003               1069          INX     B
044.230  311               1070          RET
                           1071
                           1072  *       READ FROM TERMINAL
                           1073
044.231  315 060 054       1074  FIC1    CALL    $TYPCH          REFUSE ENTRY
044.234  007               1075          DB      7               BELL
044.235  315 124 053       1076  FIC2    CALL    $RCHAR          INPUT A CHARACTER
044.240  376 004           1077          CPI     CTLD
044.242  312 001 046       1078          JE      EXIT            CTL-D
044.245  062 207 057       1079  FIC2.5  STA     $LSTIN
044.250  376 012           1080          CPI     NL
044.252  310               1081          RE                      ACCEPT WITH NO ECHO
044.253  376 040           1082          CPI     ' '
044.255  312 064 054       1083          JE      $TYPC.          ACCEPT WITH ECHO
044.260  376 060           1084          CPI     '0'
044.262  332 231 044       1085          JC      FIC1            NOT DIGIT
044.265  376 072           1086          CPI     '9'+1
044.267  332 064 054       1087          JC      $TYPC.          ACCEPT DIGIT WITH ECHO
044.272  303 231 044       1088          JMP     FIC1            REFUSE


                           1090  **      SRC - SKIP REMAINDER OF COMMAND PATTERN.
                           1091  *
                           1092  *       SRC SCANS A STRING UNTIL A BYTE IS FOUND.
                           1093  *
                           1094  *       ENTRY   (D,E) = STRING ADDRESS
                           1095  *       EXIT    (D,E) UPDATED
                           1096
                           1097
044.275  032               1098  SRC     LDAX    D
044.276  247               1099          ANA     A
044.277  023               1100          INX     D
044.300  302 275 044       1101          JNZ     SRC             MORE TO GO
044.303  311               1102          RET
```

```
044.304  000 000  1105  STKPTR  DW    0          STACK POINTER
044.306  000      1106  NXTCHA  DB    0          NEXT CHAR
044.307  000      1107  PATCNT  DB    0          INDEX OF CURRENT PATTERN
044.310  000.000  1108  CMDADR  DW    0          ADDRESS OF CURRENT COMMAND DESCRIPTOR
044.312           1109  LINE    DS    70
044.312           1110  FNRA    EQU   LINE       FNR WORK AREA
045.020           1111  LINPTR  DS    2          LINE POINTER
                  1112
045.022           1113  CMD.8A  DS    4          4 KEY VALUES
                  1114
045.026           1115  CMD.9A  DS    2          RETURN ADDRESS
                  1116
                  1117  **      ADDRESS BLOCK FORMAT.
                  1118  *
                  1119  *       EACH ADDRESS BLOCK CONSISTS OF 4 BYTES:
                  1120  *
                  1121  *       0 - FLAG BITS.
                  1122  *       1-2 - ADDRESS VALUE (IF EXPLICIT)
                  1123  *       3 - LENGTH MODIFIER
                  1124
045.030           1125  CMD.AA  DS    2*4        TWO ADDRESSES
                  1126
045.040           1127  CMD.DA  DS    4*8        8 ADDRESSES
045.100           1128  CMD.DA2 DS    1          HOLDS END OF STRING FLAG IF 8 ENTRYS
                  1129
045.101           1130  CMD.TL  DS    0          END OF TABLE
```

```
045.101                  1134  HBUG    EQU     *               MAIN ENTRY POINT
                         1135
045.101                  1136  START   EQU     *
                         1137
045.101  061 200 042     1138          LXI     SP,STACK        SET STACK VALUE
045.104  315 054 031     1139          CALL    $SAVALL         SAVE ENTRY REGISTERS
045.107  315 136 031     1140          CALL    $TYPTX
045.112  012 012 110     1141          DB      NL,NL,'HDOS DBUG # 102.06.00.'                  /WCZ080780/
045.142  040 040 040     1142  ISSUEA  DB      '   ',NL,ENL
045.150  076 001         1143          MVI     A,'A'-'@'
045.152  041 332 045     1144          LXI     H,INTRPT
045.155  377 041         1145          DB      SYSCALL,.CTLC
045.157  315 052 053     1146          CALL    SDC             SET UP DEBUG CONSOLE            /79.12.GC/
                         1147
                         1148  *       PRESET REGISTERS ON STACK
                         1149
045.162  361             1150          POP     PSW             RESTORE ENTRY REGISTERS
045.163  301             1151          POP     B
045.164  321             1152          POP     D
045.165  041 200 042     1153          LXI     H,USERFWA
045.170  343             1154          XTHL                    SET HBUG AS P-REG VALUE
045.171  345             1155  HBUG1   PUSH    H               SAVE H
045.172  325             1156          PUSH    D
045.173  305             1157          PUSH    B
045.174  365             1158          PUSH    PSW
045.175  041 012 000     1159          LXI     H,10
045.200  071             1160          DAD     SP
045.201  345             1161          PUSH    H               SAVE SP
045.202  041 000 000     1162          LXI     H,0
045.205  071             1163          DAD     SP
045.206  042 226 045     1164          SHLD    REGPTR          SAVE REGISTER POINTER


                         1166  **      TBGX - TERMINAL DEBUGGER EXIT.
                         1167  *
                         1168  *       COMMAND PROCESSORS RETURN HERE.
                         1169  *
                         1170
                         1171
045.211                  1172  RESTART EQU     *
045.211  076 005         1173          MVI     A,CN.LD
045.213  377 055         1174          DB      SYSCALL,.CLEAR          CLEAR I/O CHANNEL
                         1175
                         1176  *       CLEAR LOAD/DUMP CHANNEL
                         1177
045.215  076 005         1178          MVI     A,CN.LD
045.217  377 055         1179          DB      SYSCALL,.CLEAR  CLEAR CHANNEL
045.221  257             1180          XRA     A
045.222  062 211 057     1181          STA     MEMFB+FB.FLG    CLEAR OPEN/CLOSE FLAGS
                         1182
045.225                  1183  TBGX    EQU     *
045.225  041 000 000     1184          LXI     SP,0            (SP) = REGPTR
045.226                  1185  REGPTR  EQU     *-2             FWA OF REGISTERS ON STACK
                         1186  *       CALL    SDC             SET DEBUGGER CONSOLE ENVIRONMENT /79.12.GC/
045.230  315 246 052     1187          CALL    RBM             REMOVE BREAKPOINTS FROM MEMORY
```

```
045.233  072 330 040  1188           LDA     S.CUSOR
045.236  247          1189           ANA     A
045.237  304 135 053  1190           CNZ     $CRLF           IF LF NEEDED
045.242  315 136 031  1191           CALL    $TYPTX          TYPE PROMPT
045.245  072 102 272  1192           DB      ':B',':'+200Q
                      1193
                      1194  *        GET ANOTHER COMMAND.
                      1195
045.250  315 200 042  1196           CALL    CCP             CALL COMMAND COMPLETION PROCESSOR
045.253  072 307 044  1197           LDA     PATCNT          (A) = COMMAND INDEX
045.256  041 225 045  1198           LXI     H,TBGX
045.261  345          1199           PUSH    H               SET RETURN ADDRESS
045.262  041 030 045  1200           LXI     H,CMD.AA
045.265  315 061 031  1201           CALL    $TJMP           BRANCH THROUGH TABLE
                      1202
045.270  045 046      1203  HBUGA    DW      TB.DVS          DISPLAY VALUES, SINGLE ADDRESS
045.272  045 046      1204           DW      TB.DVP          DISPLAY VALUES, PAIR ADDRESS
045.274  063 046      1205           DW      TB.CMS          CHANGE MEMORY, SINGLE ADDRESS
045.276  063 046      1206           DW      TB.CMP          CHANGE MEMORY, PAIR ADDRESS
000.004               1207  TB.DARI  EQU     *-HBUGA/2               TB.DAR   INDEX
045.300  072 046      1208           DW      TB.DAR          DISPLAY ALL REGISTERS
045.302  113 046      1209           DW      TB.DSR          DISPLAY SINGLE REGISTER
045.304  121 046      1210           DW      TB.CSR          CHANGE SINGLE REGISTER
045.306  145 046      1211           DW      TB.EXE          EXEC COMMAND
045.310  155 046      1212           DW      TB.STP          STEP COMMAND
045.312  231 046      1213           DW      TB.SBL          SET BREAKPOINT LIST COMMAND
045.314  234 046      1214           DW      TB.DBL          DISPLAY BREAKPOINT LIST
045.316  314 046      1215           DW      TB.CBL          CLEAR BREAKPOINT LIST
045.320  346 046      1216           DW      TB.CAB          CLEAR ALL BREAKPOINTS
045.322  216 047      1217           DW      TB.DMP          DUMP
045.324  006 050      1218           DW      TB.LOA          LOAD
045.326  150 050      1219           DW      TB.LOA.         LOAD PIC
045.330  356 046      1220           DW      TB.GO           GO


                      1222  **       INTRPT - CTL-C INTERRUPT PROCESSING.
                      1223  *
                      1224  *        DECIDE IF WE WERE IN HBUG MODE OR IN USER MODE.
                      1225  *        IF HBUG MODE, JUST POP THROUGH.
                      1226
                      1227
045.332  315 136 031  1228  INTRPT   CALL    $TYPTX
045.335  136 301      1229           DB      '^','A'+200Q
045.337  076 000      1230           MVI     A,0             (A) = USER MODE FLAG
045.340               1231  USERMD   EQU     *-1
045.341  247          1232           ANA     A
045.342  312 225 045  1233           JZ      TBGX            IS JUST IN HBUG
045.345  257          1234           XRA     A
045.346  062 340 045  1235           STA     USERMD          SET DBUG MODE
045.351  315 052 053  1236           CALL    SDC             SET UP DEBUG CONSOLE            /79.12.GC/
045.354  361          1237           POP     PSW             DISCARD HDOS RETURN ADDRESS
045.355  361          1238           POP     PSW             (PSW) = USER PSW VALUES
045.356  345          1239           PUSH    H               RE-SAVE USER REGISTERS
045.357  325          1240           PUSH    D               RE-SAVE USER REGISTERS
```

```
045.360  305          1241          PUSH    B
045.361  365          1242          PUSH    PSW
045.362  041 012 000  1243          LXI     H,10
045.365  071          1244          DAD     SP
045.366  345          1245          PUSH    H               SAVE SP VALUE ON STACK
045.367  041 000 000  1246          LXI     H,0
045.372  071          1247          DAD     SP
045.373  042 226 045  1248          SHLD    REGPTR          SET NEW REGISTER POINTER
045.376  303 123 047  1249          JMP     REX             TREAT AS BREAKPOINT


                      1251  **      EXIT - PROCESS CTL-@ (END OF FILE ON CONSOLE INPUT)
                      1252  *
                      1253  *       IF HE IS SURE, EXIT  TO O/S
                      1254
                      1255
046.001  315 136 031  1256  EXIT    CALL    $TYPTX
046.004  136 104 012  1257          DB      '^D',NL,BELL,'Are You SURE?',' '+2000
046.026  315 124 053  1258          CALL    $RCHAR
046.031  315 113 053  1259          CALL    $MCU
046.034  376 131      1260          CPI     'Y'
046.036  302 225 045  1261          JNE     TRGX            SAVED AT THE BRINK OF DEATH!
046.041  076 001      1262  EXITI   MVI     A,1             FLAG ABORT EXIT                 /79.12.BC/
046.043  377 000      1263          DB      SYSCALL,.EXIT
```

```
                          1267  **      TB.DVS - DISPLAY VALUE. SINGLE ADDRESS SPECIFIED.
                          1268  *
                          1269  *       ADDR[(LEN)][OPT]
                          1270
                          1271
     046.045              1272  TB.DVS  EQU     *


                          1274  **      TB.DVP - DISPLAY VALUE, PAIRED ADDRESS SPECIFIED.
                          1275  *
                          1276  *       ADDR-ADDR[OPT]
                          1277
                          1278
     046.045              1279  TB.DVP  EQU     *
     046.045  037        1280          RAR                     (A) = COMMAND INDEX
     046.046  315 170 052 1281          CALL    RAS             RESOLVE ADDRESS SPECIFICATION
     046.051  315 373 051 1282  DVP2    CALL    DVB             DISPLAY VALUE WITH BLANK
     046.054  315 305 051 1283          CALL    CUB             SEE IF DONE             /80.02.GC/
     046.057  330        1284          RC                      DONE                    /80.02.GC/
     046.060  303 051 046 1285          JMP     DVP2                                    /80.02.GC/
```

```
                                1289  **      TB.CMS - CHANGE MEMORY, SINGLE ADDRESS SPECIFIED.
                                1290  *
                                1291  *       ADDR(LEN)]=COPT]VALUES
                                1292
                                1293
        046.063                 1294  TB.CMS  EQU     *


                                1296  **      TB.CMP - CHANGE MEMORY ADDRESS PAIR.
                                1297  *
                                1298  *       ADDR-ADDR=COPT]VALUELIST
                                1299
                                1300
        046.063                 1301  TB.CMP  EQU     *
        046.063  037           1302          RAR                     (A) = COMMAND INDEX
        046.064  315 170 052   1303          CALL    RAS             RESOLVE ADDRESS SPECIFICATION
        046.067  303 371 050   1304          JMP     ANV             ACCEPT NEW VALUES
```

```
                                1307  **      TB.DAR - DISPLAY ALL REGISTERS.
                                1308  *
                                1309  *       A=XXX, B=XXX, C=XXX, ... ,  ETC.
                                1310
                                1311
046.072  021 111 057            1312  TB.DAR  LXI     D,DARA
046.075  006 013                1313          MVI     B,DARAL         (B) = ENTRY COUNT
046.077  315 135 053            1314          CALL    $CRLF           NEW LINE
046.102  315 353 051            1315  TB.DAR1 CALL    DRV             DISPLAY REGISTER VALUE
046.105  005                    1316          DCR     B
046.106  023                    1317          INX     D
046.107  302 102 046            1318          JNZ     TB.DAR1
046.112  311                    1319          RET                     EXIT


                                1321  **      TB.DSR - DISPLAY SINGLE REGISTER
                                1322  *
                                1323
                                1324
046.113  315 337 051            1325  TB.DSR  CALL    DRI             DETERMINE REGISTER INDEX
046.116  303 357 051            1326          JMP     DRV.            DISPLAY REGISTER VALUE


                                1328  **      TB.CSR - CHANGE SINGLE REGISTER
                                1329  *
                                1330
                                1331
046.121                         1332  TB.CSR  EQU     *
046.121  315 337 051            1333          CALL    DRI             DETERMINE REGISTER INDEX
046.124  315 322 051            1334          CALL    DRA             DETERMINE REGISTER ADDRESS
046.127  124                    1335          MOV     D,H
046.130  135                    1336          MOV     E,L
046.131  362 135 046            1337          JP      CSR1            IF SINGLE
046.134  023                    1338          INX     D
046.135  346 200                1339  CSR1    ANI     200Q
046.137  062 023 045            1340          STA     CMD.BA+1
046.142  303 371 050            1341          JMP     ANV             ACCEPT NEW VALUE AND EXIT
```

```
                                1345  **      TB.EXE - PROCESS EXEC COMMAND.
                                1346  *
                                1347  *       EXEC ADDR-ADDR(CNT)],...,ADDR(CNT)]
                                1348
                                1349
        046.145                 1350  TB.EXE  EQU     *
        046.145  345           1351          PUSH    H               SAVE START ADDRESS POINTER
        046.146  315 326 052   1352          CALL    SBL             SET BREAKPOINT LIST
        046.151  341           1353          POP     H               (HL) = ADDRESS OF START BLOCK
        046.152  303 356 046   1354          JMP     TB.GO           PROCESS AS *GO*
```

```
                                1358  **      TB.STP - PROCESS SINGLE STEP COMMAND.
                                1359  *
                                1360  *       STEP                    SINGLE STEP AT *P*
                                1361  *       STEP (CNT)              STEP CNT TIMES FROM *P*
                                1362  *       STEP ADDR               STEP ONCE AT *ADDR*
                                1363  *       STEP ADDR(CNT)          STEP CNT TIMES FROM *ADDR*
                                1364
                                1365
046.155                         1366  TB.STP  EQU     *
046.155  315 074 053            1367          CALL    SSA             SET STARTING ADDRESS
046.160  072 033 045            1368          LDA     CMD.AA+3        (A) = COUNT
046.163  062 174 046            1369  STP1    STA     STPA            SAVE
046.166  041 175 046            1370          LXI     H,STPRTN
046.171  303 160 047            1371          JMP     BKP2            PROCESS AS BKPT
                                1372
046.174  000                   1373  STPA    DB      0
                                1374
                                1375  **      SINGLE STEP RETURNS HERE
                                1376
046.175  257                   1377  STPRTN  XRA     A
046.176  062 340 045            1378          STA     USERMD
046.201  315 052 053            1379          CALL    SDC             SET UP DEBUG CONSOLE            /79.12.GC/
046.204  041 000 000            1380          LXI     H,0
046.207  071                   1381          DAD     SP              (HL) = REGPTR VALUE
046.210  042 226 045            1382          SHLD    REGPTR
046.213  315 301 052            1383          CALL    RFD             RESTORE FRONT PANEL DISPLAY
046.216  072 174 046            1384          LDA     STPA
046.221  373                   1385          EI
046.222  075                   1386          DCR     A
046.223  302 143 046            1387          JNZ     STP1
046.226  303 123 047            1388          JMP     REX             RETURN FROM EXECUTION
```

```
                        1392  **      TB.SBL - SET BREAKPOINT LIST.
                        1393  *
                        1394  *       BKPT    A1,...,AN
                        1395
                        1396
      046.231           1397  TB.SBL  EQU     *
      046.231 303 326 052 1398          JMP     SBL             SET BREAKPOINT LIST
```

```
                                1402  **      TB.DBL - DISPLAY BREAKPOINT LIST.
                                1403  *
                                1404  *       TYPE OUT LIST OF ALL BREAKPOINTS, WITH THEIR REPEAT COUNTS.
                                1405  *
                                1406  *       ADDR/RPT
                                1407
                                1408
046.234                         1409  TB.DBL  EQU     *
046.234  041 140 057           1410          LXI     H,BKPTAB
046.237  006 010               1411          MVI     B,BKPTBL
                                1412
                                1413  *       TYPE NON-NULL ENTRYS
                                1414
046.241  353                  1415  DBL1    XCHG
046.242  041 000 106           1416          LXI     H,'F'*256       FULL WORD OCTAL
046.245  042 022 045           1417          SHLD    CMD.8A          SET OPTION
046.250  353                  1418          XCHG
046.251  176                  1419          MOV     A,M
046.252  043                  1420          INX     H
046.253  266                  1421          ORA     M
046.254  312 304 046           1422          JZ      DBL2            IF NULL
046.257  053                  1423          DCX     H
046.260  315 040 052           1424          CALL    FVD             FORMAT VALUE FOR DISPLAY
046.263  315 060 054           1425          CALL    $TYPCH
046.266  057                  1426          DB      '/'
046.267  353                  1427          XCHG
046.270  041 104 000           1428          LXI     H,'D'
046.273  042 022 045           1429          SHLD    CMD.8A          SET DECIMAL BYTE
046.276  353                  1430          XCHG
046.277  315 373 051           1431          CALL    DVB             DISPLAY VALUE WITH BLANK
046.302  053                  1432          DCX     H
046.303  053                  1433          DCX     H
                                1434
                                1435  *       ENTRY PROCESSED, CHECK NEXT.
                                1436
046.304  043                  1437  DBL2    INX     H
046.305  043                  1438          INX     H
046.306  043                  1439          INX     H
046.307  005                  1440          DCR     B
046.310  302 241 046           1441          JNZ     DBL1
046.313  311                  1442          RET                     DONE, EXIT
```

```
                         1446  **      TB.CBL - CLEAR BREAKPOINT LIST.
                         1447  *
                         1448  *       CLEAR A1,...,AN
                         1449
                         1450
046.314                  1451  TB.CBL  EQU     *
046.314  056 040         1452          MVI     L,$CMD.DA
                         1453
                         1454  *       EXAMINE NEXT ADDRESS SUPPLIED.
                         1455
046.316  176           1456  CBL1    MOV     A,M
046.317  017           1457          RRC
046.320  330           1458          RC                      END OF LIST
046.321  043           1459          INX     H
                         1460
                         1461  *       FIND SPECIFIED BREAKPOINT
                         1462
046.322  116           1463          MOV     C,M
046.323  043           1464          INX     H
046.324  106           1465          MOV     B,M             (BC) = SPECIFEID ADDRESS
046.325  315 003 052   1466          CALL    FBT             FIND BREAKPOINT IN TABLE
046.330  302 341 046   1467          JNE     CBL3            IF NOT FOUND
                         1468
                         1469  *       FOUND IT. (DE) = ADDRESS
                         1470
046.333  257           1471  CBL2    XRA     A
046.334  022           1472          STAX    D
046.335  023           1473          INX     D
046.336  022           1474          STAX    D
046.337  023           1475          INX     D
046.340  022           1476          STAX    D
                         1477
                         1478  *       LOOK AT NEXT ADDRESS
                         1479
046.341  043           1480  CBL3    INX     H
046.342  043           1481          INX     H
046.343  303 316 046   1482          JMP     CBL1
```

```
                                1486  **      TB.CAB - CLEAR ALL BREAKPOINTS.
                                1487  *
                                1488  *       CLEAR ALL
                                1489
                                1490
046.346 041 140 057             1491  TB.CAB  LXI     H,BKPTAB
046.351 006 040                 1492          MVI     B,BKPTBL*4      (D) = LENGTH
046.353 303 212 031             1493          JMP     $ZERO           ZERO MEMORY
```

```
                                1497  **      TB.GO - PROCESSS *GO* COMMAND.
                                1498  *
                                1499
                                1500
046.356                         1501  TB.GO   EQU     *
046.356  315 074 053           1502          CALL    SSA             SET START ADDRESS
046.361  315 015 053           1503  GOO     CALL    SBM             SET BREAKPOINTS IN MEMORY
046.364  041 045 047           1504          LXI     H,.BKP.
046.367  042 043 040           1505  GO2     SHLD    :UIVEC+4
046.372  076 303               1506          MVI     A,MI.JMP
046.374  062 042 040           1507          STA     :UIVEC+3                SETUP VECTOR
046.377  052 226 045           1508  GO      LHLD    REGPTR
047.002  371                   1509          SPHL                    RESET STACK
047.003  363                   1510          DI
047.004  041 340 045           1511          LXI     H,USERMD
047.007  064                   1512          INR     M               SET USER MODE
047.010  341                   1513          POP     H               (HL) = STACKPOINTER VALUE
047.011  042 033 047           1514          SHLD    GOA             SAVE FOR STACK
047.014  315 310 052           1515          CALL    RUC             RESTORE USER CONSOLE INVIRONMENT
047.017  361                   1516          POP     PSW
047.020  301                   1517          POP     B
047.021  321                   1518          POP     D
047.022  341                   1519          POP     H
047.023  042 037 047           1520          SHLD    GOB             SAVE (HL) FOR LATER PICKUP
047.026  341                   1521          POP     H               (HL) = RETURN ADDRESS
047.027  042 043 047           1522          SHLD    GOC             SET RETURN ADDRESS
047.032  041 000 000           1523          LXI     H,0             (HL) = STACKPOINTER
047.033                         1524  GOA     EQU     *-2
047.035  371                   1525          SPHL                    SET STACK
047.036  041 000 000           1526          LXI     H,0             (HL) = (HL)
047.037                         1527  GOB     EQU     *-2
047.041  373                   1528          EI
047.042  303 000 000           1529          JMP     0
047.043                         1530  GOC     EQU     *-2             ADDRESS OF ENTRY TO USER PROGRAM
                                1531
                                1532
                                1533  **      CONTROL IS PASSED HERE WHEN BREAKPOINT IS HIT.
                                1534
047.045                         1535  .BKP.   EQU     *
047.045  257                   1536          XRA     A
047.046  062 340 045           1537          STA     USERMD          CLEAR USER MODE
047.051  315 052 053           1538          CALL    SDC             SET UP DEBUG CONSOLE            /79.12.GC/
047.054  041 000 000           1539          LXI     H,0
047.057  071                   1540          DAD     SP
047.060  042 226 045           1541          SHLD    REGPTR          SAVE REGISTER POINTER
047.063  315 246 052           1542          CALL    RBM             REMOVE BREAKPOINTS FROM MEMORY
047.066  041 012 000           1543          LXI     H,10
047.071  071                   1544          DAD     SP
047.072  116                   1545          MOV     C,M
047.073  043                   1546          INX     H
047.074  106                   1547          MOV     B,M
047.075  013                   1548          DCX     B               (BC) = ADDRESS OF INSTRUCTION HIT
047.076  160                   1549          MOV     M,B             STORE DECREMENTED PC
047.077  053                   1550          DCX     H
047.100  161                   1551          MOV     M,C
047.101  315 003 052           1552          CALL    FBT             FIND BREAKPOINT
```

```
047.104  302 123 047  1553           JNZ     REX             IF NOT FOUND
047.107  023          1554           INX     D
047.110  023          1555           INX     D
047.111  353          1556           XCHG
047.112  065          1557           DCR     M
047.113  302 155 047  1558           JNZ     BKP1            IF MORE ITTERATIONS BEFORE ACKNOWLEDING
                      1559
                      1560  *        BREAKPOINT COUNT EXHAUSTED, ACKNOWLEDGE.
                      1561
047.116  257          1562           XRA     A
047.117  053          1563           DCX     H
047.120  167          1564           MOV     M,A
047.121  053          1565           DCX     H
047.122  167          1566           MOV     M,A             CLEAR TEBLE ENTRY


                      1568  **       REX - RETURN FROM EXECUTION
                      1569  *
                      1570  *        PRINT -P=NNNNNN-
                      1571
047.123               1572  REX      EQU     *
047.123  315 136 031  1573           CALL    $TYPTX
047.126  055 120 275  1574           DB      '-P','='+200Q
047.131  041 000 106  1575           LXI     H,'F'*256
047.134  042 022 045  1576           SHLD    CMD.8A          DOUBLE OCTAL VALUE
047.137  315 317 051  1577           CALL    DRA.            DETERMINE REGISTER ADDRESS
047.142  315 040 052  1578           CALL    FVD             FORMAT VALUE
047.145  315 136 031  1579           CALL    $TYPTX
047.150  055 212      1580           DB      '-',ENL
047.152  303 225 045  1581           JMP     TBGX            ENTER CONTROL LOOP
                      1582
                      1583  *        MORE HITS ON THIS BREAKPOINT
                      1584
047.155  041 201 047  1585  BKP1     LXI     H,GO3
047.160  363          1586  BKP2     DI
047.161  072 011 040  1587           LDA     .CTLFLG
047.164  062 302 052  1588           STA     RFDA            SAVE FOR *RFD*
047.167  346 257      1589           ANI     377Q-CB.SSI-CB.CLI  ENABLE STEP, CLEAR CLOCK
047.171  062 011 040  1590           STA     .CTLFLG
047.174  323 360      1591           OUT     OP.CTL
047.176  303 367 046  1592           JMP     GO2             SINGLE STEP OVER SITE OF BREAKPOINT
                      1593
                      1594  **       RETURN FROM SINGLE STEPPING OVER BREAKPOINTED INSTRUCTION
                      1595
047.201  041 000 000  1596  GO3      LXI     H,0
047.204  071          1597           DAD     SP
047.205  042 226 045  1598           SHLD    REGPTR
047.210  315 301 052  1599           CALL    RFD             RESTORE FRONT PANEL DISPLAY
047.213  303 361 046  1600           JMP     GOO
```

```
                                1604  ***     TB.DMP - PROCESS *DUMP* COMMAND.
                                1605  *
                                1606  *       DUMP FNAME ADDR1-ADDR2
                                1607  *
                                1608  *       DUMP IN ABS FORMAT.
                                1609
                                1610
047.216                         1611  TB.DMP  EQU     *
                                1612
                                1613  *       COMPUTE DUMP FWA
                                1614
047.216  072 030 045           1615          LDA     CMD.AA
047.221  037                    1616          RAR
047.222  332 233 047           1617          JC      DMP0            DEFAULT FWA
047.225  052 031 045           1618          LHLD    CMD.AA+1
047.230  042 245 057           1619          SHLD    BFILHDR+ABS.LDA SET FWA
                                1620
                                1621  *       COMPUTE LEN
                                1622
047.233  072 034 045           1623  DMP0    LDA     CMD.AA+4
047.236  037                    1624          RAR
047.237  332 312 047           1625          JC      DMP2            LWA DEFAULTS
047.242  052 035 045           1626          LHLD    CMD.AA+5
047.245  353                    1627          XCHG
047.246  052 245 057           1628          LHLD    BFILHDR+ABS.LDA
047.251  053                    1629          DCX     H                               /78.10.GC/
047.252  173                    1630          MOV     A,E
047.253  225                    1631          SUB     L
047.254  157                    1632          MOV     L,A
047.255  172                    1633          MOV     A,D
047.256  234                    1634          SBB     H
047.257  147                    1635          MOV     H,A             (HL) = COUNT
047.260  332 271 047           1636          JC      DMP1            LWA < FWA
047.263  042 247 057           1637          SHLD    BFILHDR+ABS.LEN SET LENGTH
047.266  303 312 047           1638          JMP     DMP2            OPEN FILE
                                1639
                                1640  *       LWA < FWA
                                1641
047.271  315 136 031           1642  DMP1    CALL    $TYPTX
047.274  007 114 127           1643          DB      BELL,'LWA < FWA',ENL
047.307  303 225 045           1644          JMP     TBGX            EXIT
                                1645
                                1646  *       OPEN DUMP FILE
                                1647
047.312  021 000 050           1648  DMP2    LXI     D,DMPA          USE 'SYOABS' AS DEFAULTS
047.315  041 210 057           1649          LXI     H,MEMFB
047.320  315 076 054           1650          CALL    $FOPEW
                                1651
                                1652  *       WRITE HEADER INFO
                                1653
047.323  315 317 051           1654          CALL    DRA.            LOCATE PC
047.326  315 211 030           1655          CALL    $HLIHL          (HL) = (PC)
047.331  042 251 057           1656          SHLD    BFILHDR+ABS.ENT SET ENTRY
047.334  041 377 000           1657          LXI     H,FT.ABS*256+377Q
047.337  042 243 057           1658          SHLD    BFILHDR         SET BINARY ABS HEADER
047.342  001 010 000           1659          LXI     B,ABS.COD
```

```
047.345  021 243 057  1660          LXI     D,BFILHDR
047.350  041 210 057  1661          LXI     H,MEMFB
047.353  315 000 055  1662          CALL    $FWRIB          WRITE HEADER BYTES TO FILE
047.356  052 247 057  1663          LHLD    BFILHDR+ABS.LEN
047.361  104          1664          MOV     B,H
047.362  115          1665          MOV     C,L             (BC) = COUNT
047.363  052 245 057  1666          LHLD    BFILHDR+ABS.LDA
047.366  353          1667          XCHG                    (DE) = ADDRESS
047.367  041 210 057  1668          LXI     H,MEMFB
047.372  315 000 055  1669          CALL    $FWRIB          WRITE BINARY
047.375  303 266 055  1670          JMP     $FCLO           CLOSE FILE
                      1671
050.000  123 131 060  1672  DMPA    DB      'SYOABS'                DEFAULTS FOR DUMP
```

```
                                1676  ***     TB.LOAD - PROCESS *LOAD* COMMAND.
                                1677  *
                                1678  *       LOAD FNAME
                                1679  *
                                1680  *       LOAD ABS FILE INTO MEMORY.
                                1681
                                1682
        050.006                 1683  TB.LOA  EQU     *
        050.006  021 142 050    1684          LXI     D,LOAA          DEFAULT TO 'SYOABS'
        050.011  041 210 057    1685          LXI     H,MEMFB
        050.014  315 067 054    1686          CALL    $FOPER          OPEN FOR READ
        050.017  001 010 000    1687          LXI     B,ABS.COD
        050.022  021 243 057    1688          LXI     D,BFILHDR
        050.025  315 227 054    1689          CALL    $FREAB          READ HEADER
        050.030  332 107 050    1690          JC      LOA2            PREMATURE EOF
        050.033  052 243 057    1691          LHLD    BFILHDR
        050.036  054            1692          INR     L
        050.037  302 107 050    1693          JNZ     LOA2            NOT BINARY FILE
        000.000                 1694          ERRNZ   FT.ABS
        050.042  174            1695          MOV     A,H
        050.043  247            1696          ANA     A
        050.044  302 107 050    1697          JNZ     LOA2            NOT BINARY FILE
        050.047  052 251 057    1698          LHLD    BFILHDR+ABS.ENT (HL) = ENTRY POINT
        050.052  345            1699          PUSH    H
        050.053  315 317 051    1700          CALL    DRA.            (HL) = ADDRESS OF USER PC
        050.056  321            1701          POP     D               (DE) = NEW PC
        050.057  163            1702          MOV     M,E
        050.060  043            1703          INX     H
        050.061  162            1704          MOV     M,D
                                1705
                                1706  *       SETUP LOAD FWA AND COUNT
                                1707
        050.062  052 247 057    1708          LHLD    BFILHDR+ABS.LEN
        050.065  104            1709          MOV     B,H
        050.066  115            1710          MOV     C,L             (BC) = COUNT
        050.067  052 245 057    1711          LHLD    BFILHDR+ABS.LDA
        050.072  124            1712          MOV     D,H
        050.073  135            1713          MOV     E,L             (DE) = FWA
        050.074  011            1714          DAD     B               (HL) = LWA+1
        050.075  345            1715          PUSH    H               SAVE FOR LATER
        050.076  315 216 051    1716          CALL    CLR             CHECK LOAD RANGE
        050.101  315 227 054    1717          CALL    $FREAB          READ DATA
        050.104  322 314 050    1718          JNC     LOA.2           CLOSE AND END, IF NO ERRORS
                                1719
                                1720  *       FILE FORMAT ERROR
                                1721
        050.107  315 136 031    1722  LOA2    CALL    $TYPTX
        050.112  007 106 117    1723          DB      BELL,'FORMAT ERROR IN FIL','E'+2000
        050.137  303 225 045    1724          JMP     TBGX            EXIT
                                1725
        050.142  123 131 060    1726  LOAA    DB      'SYOABS'        DEFAULT LOAD
```

```
                                1728  ***     TB.LOA. - PROCESS *LOAD PIC* COMMAND.
                                1729  *
                                1730  *       LOAD PIC FNAME ADDR
                                1731  *
                                1732  *       LOAD PIC FILE INTO MEMORY AT LOCATION
                                1733
                                1734
050.150                         1735  TB.LOA. EQU     *
050.150  021 363 050           1736          LXI     D,LOAB          DEFAULTS OF 'SYOPIC'
050.153  041 210 057           1737          LXI     H,MEMFB
050.156  315 067 054           1738          CALL    $FOPER          OPEN FILE
050.161  001 006 000           1739          LXI     B,PIC.COD
050.164  021 243 057           1740          LXI     D,BFILHDR
050.167  315 227 054           1741          CALL    $FREAB          READ HEADER
050.172  332 107 050           1742          JC      LOA2            PREMATURE EOF
050.175  052 243 057           1743          LHLD    BFILHDR
050.200  054                   1744          INR     L
050.201  302 107 050           1745          JNZ     LOA2            NOT BINARY
000.000                         1746          ERRNZ   FT.PIC-1
050.204  045                   1747          DCR     H
050.205  302 107 050           1748          JNZ     LOA2            NOT PIC
                                1749
                                1750  *       LOAD CODE BEFORE RELOCATION
                                1751
050.210  052 247 057           1752          LHLD    BFILHDR+PIC.PTR
050.213  001 372 377           1753          LXI     B,-PIC.COD
050.216  011                   1754          DAD     B               (HL) = BYTES TO READ
050.217  104                   1755          MOV     B,H
050.220  115                   1756          MOV     C,L
050.221  052 031 045           1757          LHLD    CMD.AA+1
050.224  353                   1758          XCHG                    (DE) = LOAD ADDRESS
050.225  315 216 051           1759          CALL    CLR             CHECK LOAD RANGE
050.230  315 227 054           1760          CALL    $FREAB          READ BYTES
050.233  332 107 050           1761          JC      LOA2            FORMAT ERROR
                                1762
                                1763  *       RELOCATE CODE
                                1764
050.236  325                   1765          PUSH    D               SAVE NEXT FREE ADDRESS
050.237  052 031 045           1766          LHLD    CMD.AA+1        (HL) = LOAD ADDRESS
050.242  001 372 377           1767          LXI     B,-PIC.COD
050.245  011                   1768          DAD     B               (HL) = RELOCATION FACTOR
050.246  104                   1769          MOV     B,H
050.247  115                   1770          MOV     C,L
050.250  041 210 057           1771  LOA.1   LXI     H,MEMFB
050.253  305                   1772          PUSH    B               SAVE RELOCATION FACTOR
050.254  001 002 000           1773          LXI     B,2
050.257  021 312 044           1774          LXI     D,LINE
050.262  315 227 054           1775          CALL    $FREAB          READ RELOCATION BYTES
050.265  301                   1776          POP     B               RESTORE RELOCATION FACTOR
050.266  332 107 050           1777          JC      LOA2            FORMAT ERROR
050.271  052 312 044           1778          LHLD    LINE            (HL) = REL ADDRESS OF WORD TO RELOCATE
050.274  174                   1779          MOV     A,H
050.275  265                   1780          ORA     L
050.276  312 314 050           1781          JZ      LOA.2           ALL DONE
050.301  011                   1782          DAD     B               (HL) = ABS ADDRESS OF WORD TO RELOCATE
050.302  176                   1783          MOV     A,M
```

```
050.303  201           1784           ADD     C
050.304  167           1785           MOV     M,A
050.305  043           1786           INX     H
050.306  176           1787           MOV     A,M
050.307  210           1788           ADC     B
050.310  167           1789           MOV     M,A             RELOCATE WORD
050.311  303 250 050   1790           JMP     LOA.1
                       1791
                       1792  *        ALL DONE, PRINT NEXT FREE ADDRESS
                       1793
050.314  041 210 057   1794  LOA.2    LXI     H,MEMFB
050.317  315 266 055   1795           CALL    $FCLO           CLOSE  INPUT FILE
050.322  041 000 106   1796           LXI     H,'F'*256
050.325  042 022 045   1797           SHLD    CMD.8A          FORMAT DOUBLE OCTAL VALUE
050.330  041 000 000   1798           LXI     H,0
050.333  071           1799           DAD     SP              (HL) = ADDRESS OF VALUE
050.334  315 136 031   1800           CALL    $TYPTX
050.337  114 127 101   1801           DB      'LWA+1 =',' '+200Q
050.347  315 040 052   1802           CALL    FVD             FORMAT VALUE FOR DISPLAY
050.352  341           1803           POP     H
                       1804
                       1805  *        RE-INITIALIZE THE DEFAULT CONSOLE DEFINITION BYTES      /79.12.GC/
                       1806
050.353  257           1807           XRA     A                                              /79.12.GC/
050.354  062 107 057   1808           STA     CSLMD                                          /79.12.GC/
050.357  062 110 057   1809           STA     CONFL                                          /79.12.GC/
                       1810
050.362  311           1811           RET
                       1812
050.363  123 131 060   1813  LOAB     DB      'SYOPIC'        DEFAULTS FOR PIC LOAD
```

```
                                1817  **      ANV - ACCEPT NEW VALUE.
                                1818  *
                                1819  *       ANV IS CALLED TO ACCEPT A NEW SINGLE OR DOUBLE BYTE VALUE.
                                1820  *       THE OLD VALUE IS TYPED OUT, FOLLOWED BY A '/', AND THEN
                                1821  *       A NEW VALUE MAY BE NETERED.
                                1822  *
                                1823  *       IF MODE IS OCTAL OR DECIMAL, A BLANK TERMINATES THE
                                1824  *       CURRENT VALUE, A /CR/ TERMINATES THE CURRENT VALUE AND
                                1825  *       THE OPERATION. A NULL VALUE CAUSES THAT BYTE TO REMAIN
                                1826  *       UNCHANGED.
                                1827  *
                                1828  *       IN ASCII MODE, AN /ESC/ TERMINATES ENTRY.
                                1829  *
                                1830  *       ENTRY   (HL) = START ADDRESS
                                1831  *               (DE) = LIMIT ADDRESS
                                1832
                                1833
050,371                         1834  ANV     EQU     *
050.371  076 303                1835          MVI     A,MI.JMP
050.373  062 217 044           1836          STA     FICA            SET FLAG TO READ FROM ITY
                                1837
                                1838  *       TYPE OUT 'OLD VALUE'
                                1839
050,376  325                   1840  ANV1    PUSH    D               SAVE (DE)
050.377  345                   1841          PUSH    H
051,000  315 040 052           1842          CALL    FVD             FORMAT VALUE FOR DISPLAY
051.003  341                   1843          POP     H
051,004  315 060 054           1844          CALL    $TYPCH
051.007  057                   1845          DB      '/'
051,010  072 022 045           1846          LDA     CMD,8A          (A) = DISPLAY OPTION
051.013  026 012               1847          MVI     D,10
051,015  376 104               1848          CPI     'D'
051.017  312 031 051           1849          JE      ANV2            IF DECIMAL
051,022  026 010               1850          MVI     D,8             ASSUME OCTAL (NOT SPECIFIED)
051.024  376 101               1851          CPI     'A'
051,026  312 077 051           1852          JE      ANV4            IS ASCII
                                1853
                                1854  *       ACCUMULATE A DIGIT VALUE
                                1855
051,031  076 120               1856  ANV2    MVI     A,80            (A) = DIGIT COUNT
051.033  315 353 043           1857          CALL    ACN             ACCUMULATE NUMBER
051,036  072 023 045           1858          LDA     CMD,8A+1        (A) /,0 IF FOLLWORD
051.041  312 072 051           1859          JZ      ANV5            IS NULL ENTRY
                                1860
                                1861  *       STORE ENTRY
                                1862
051.044  163                   1863          MOV     M,E             STORE
051,045  043                   1864          INX     H
051.046  247                   1865          ANA     A
051,047  312 054 051           1866  ANV3    JZ      ANV4            IF SINGLE BYTE
051.052  162                   1867          MOV     M,D
051,053  043                   1868          INX     H
                                1869
                                1870  *       ACCEPTED VALUE, IF HE TYPED / /, CONTINUE
                                1871  *       IF IS A CARRIAGE RETURN, STOP.
                                1872
```

```
051.054  321           1873  ANV4    POP     D
051.055  072 207 057   1874          LDA     $LSTIN
051.060  376 040       1875          CPI     ' '
051.062  300           1876          RNE                     STOP IF NOT ' '
051.063  315 305 051   1877  ANV4$5  CALL    CUB             CHECK TO SEE IF DONE            /80.02.GC/
051.066  330           1878          RC                      IF DONE
051.067  303 376 050   1879          JMP     ANV1            MORE DATA
                       1880
                       1881  *       NULL ENTRY
                       1882
051.072  043           1883  ANV5    INX     H
051.073  106           1884          MOV     B,M
051.074  303 047 051   1885          JMP     ANV3            ADJUST MEMORY POINTER
                       1886
                       1887
                       1888  **      IS ASCII VALUE
                       1889
051.077  315 143 053   1890  ANV6    CALL    $INCHA
051.102  376 004       1891          CPI     CTLD
051.104  312 001 046   1892          JE      EXIT            CTL-D
051.107  315 064 054   1893          CALL    $TYPC.          ECHO
051.112  321           1894          POP     D
051.113  376 033       1895          CPI     ESC
051.115  310           1896          RE                      EXIT IF BREAK
051.116  167           1897          MOV     M,A
051.117  043           1898          INX     H
051.120  303 063 051   1899          JMP     ANV4$5


                       1901  **      CEA - COMPUTE EFFECTIVE ADDRESS.
                       1902  *
                       1903  *       ENTRY   (HL) = ADDRESS BLOCK
                       1904  *       EXIT    (HL) = EFFECTIVE ADDRESS
                       1905
                       1906
051.123  176           1907  CEA     MOV     A,M             (A) = FLAGS
051.124  017           1908          RRC
051.125  332 142 051   1909          JC      CEA1            IS BOTTOM VALUE
051.130  017           1910          RRC
051.131  332 146 051   1911          JC      CEA2            IS TOP VALUE
                       1912
                       1913  *       HAVE SPECIFIED ADDRESS.
                       1914
051.134  043           1915          INX     H
051.135  176           1916          MOV     A,M
051.136  043           1917          INX     H
051.137  146           1918          MOV     H,M
051.140  157           1919          MOV     L,A
051.141  311           1920          RET
                       1921
                       1922  *       HAVE BOTTOM (LAST+1) VALUE
                       1923
051.142  052 105 057   1924  CEA1    LHLD    BOTVAL
051.145  311           1925          RET
```

```
                                1926
                                1927  *       HAVE TOP (FIRTST) VALUE
                                1928
051,146  052 103 057            1929  CEA2    LHLD    TOPVAL
051.151  311                    1930          RET


                                1932  **      CLL - CHECK LINE LENGTHS.
                                1933  *
                                1934  *       CLL IS CALLED TO CHECK IF THE CURRENT LINE IS TOO LONG TO
                                1935  *       CONTINUE
                                1936  *
                                1937  *       USES    A,F
                                1938
                                1939
051,152                         1940  CLL     EQU     *
051.152  072 330 040            1941          LDA     S.CUSOR
051,155  306 010                1942          ADI     8               SEE IF WILL RUN OVER
051.157  305                    1943          PUSH    B
051,160  107                    1944          MOV     B,A             (B) = CURRENT COLUMN NUMBER
051.161  072 331 040            1945          LDA     S.CONWI
051,164  270                    1946          CMP     B
051.165  301                    1947          POP     B
051,166  320                    1948          RNC                     NOT AT END
051.167  315 135 053            1949          CALL    $CRLF           NEW LINE
051,172  072 307 044            1950          LDA     PATCNT          DONT PRINT ADDRESS FOR CB,DAR
051.175  376 004                1951          CPI     TB.DARI
051,177  310                    1952          RE                      SKIP IT
051.200  174                    1953          MOV     A,H
051,201  315 032 054            1954          CALL    $TOD            TYPE OCTAL DIGIT
051.204  175                    1955          MOV     A,L
051,205  315 032 054            1956          CALL    $TOD            TYPE OCTAL DIGITS
051.210  315 136 031            1957          CALL    $TYPTX
051,213  040 240                1958          DB      ' ',' '+200Q
051.215  311                    1959          RET


                                1961  **      CLR - CHECK LOAD RANGE.
                                1962  *
                                1963  *       CLR IS CALLED BEFORE A MEMORY LOAD IS PERFORMED.  IT REQUESTS
                                1964  *       SUFFICIENT MEMORY FROM HDOS, AND MAKES SURE THAT THE PROGRAM WILL
                                1965  *       NOT LOAD OVER DBUG.
                                1966  *
                                1967  *       ENTRY   (BC) = TOTAL LENGTH OF LOAD
                                1968  *               (DE) = LOAD FWA
                                1969  *       EXIT    TO CALLER IF OK
                                1970  *                (HL) = #MEMFB
                                1971  *               TO APPROPRIATE ERROR HANDLER (AND THUS TO TRGX) IF ERROR
                                1972  *       USES    A,F,H,L
                                1973
                                1974
051,216  305                    1975  CLR     PUSH    B
```

```
051.217  325            1976          PUSH    D               SAVE REGISTERS
051.220  353            1977          XCHG
051.221  011            1978          DAD     B               (HL) = NEW LWA
051.222  377 052        1979          DB      SYSCALL,,SETTP
051.224  041 210 057    1980          LXI     H,MEMFB         POINT TO FILE IF ERROR
051.227  332 373 055    1981          JC      $FERROR         MEMORY OVERFLOW
051.232  321            1982          POP     D
051.233  301            1983          POP     B               RESTORE REGISTERS
051.234  041 025 317    1984          LXI     H,-RMEML
051.237  031            1985          DAD     D               SEE IF OVERLAYING DBUG
051.240  041 210 057    1986          LXI     H,MEMFB
051.243  330            1987          RC                      NOT OVERLAYING DBUG
051.244  315 136 031    1988          CALL    $TYPTX
051.247  007 101 164    1989          DB      BELL,'Attempt to Load Over DBUG',ENL
051.302  303 211 045    1990          JMP     RESTART         RESET FILES, ENTER COMMAND MODE


                        1992  **      CUB     - CHECK UPPER BOUND                     /80.02.GC/
                        1993  *
                        1994  *       CUB check bounds to see if enough have been processed.
                        1995  *
                        1996  *
                        1997  *       ENTRY:  HL      = NEXT BYTE
                        1998  *               DE      = LAST BYTE
                        1999  *
                        2000  *       EXIT:   PSW     = 'C' SET IF DONE
                        2001  *
                        2002  *       USES:   PSW
                        2003  *
                        2004
051.305  173            2005  CUB     MOV     A,E
051.306  225            2006          SUB     L
051.307  172            2007          MOV     A,D
051.310  234            2008          SBB     H
051.311  330            2009          RC                      DONE
                        2010
051.312  174            2011          MOV     A,H
051.313  265            2012          ORA     L
051.314  300            2013          RNZ                     NEXT ONE IS NOT ZERO
                        2014
051.315  067            2015          STC                     FLAG IT DONE FOR NO WRAP THROUGH THE TOP
051.316  311            2016          RET


                        2018  **      DRA - DETERMINE REGISTER ADDRESS.
                        2019  *
                        2020  *       ENTRY   (DE) = ADDRESS OF *DARA* ENTRY
                        2021  *       EXIT    (HL) = ADDRESS OF VALUE IN MEMORY
                        2022  *                'M' SET IF DOUBLE BYTE VALUE
                        2023  *       USES    A,F,D,E,H,L
                        2024
                        2025
```

```
051.317  021 131 057  2026  DRA.   LXI   D,DARAP
051.322  023          2027  DRA    INX   D
051.323  032          2028         LDAX  D              (A) = CODE
051.324  346 177      2029         ANI   177Q
051.326  052 226 045  2030         LHLD  REGPTR
051.331  315 072 030  2031         CALL  $DADA
051.334  032          2032         LDAX  D              (A) = CODE
051.335  247          2033         ANA   A              SET CODE
051.336  311          2034         RET

                      2036  **     DRI - DETERMINE REGISTER INDEX
                      2037  *
                      2038  *      ENTRY  CMD.8A+1 = REGISTER CODE
                      2039  *      EXIT   (BC) = ADDRESS OF ENTRY IN *PARA*
                      2040  *      USES   A,B,C,D,F
                      2041
                      2042
051.337  072 024 045  2043  DRI    LDA   CMD.8A+2
051.342  041 111 057  2044         LXI   H,DARA
051.345  315 277 053  2045         CALL  $TBLS          TABLE LOOKUP AND RETURN
051.350  053          2046         DCX   H
051.351  353          2047         XCHG
051.352  311          2048         RET

                      2050  **     DRV - DISPLAY REGISTER VALUE.
                      2051  *
                      2052  *      DRV DISPLAYS A REGISTER AS
                      2053  *
                      2054  *      R=XXX    IF 8 BIT, OR
                      2055  *      R=XXXXX  IF 16 BIT
                      2056  *
                      2057  *      THE DISPLAY FORMAT OPTIONS MUST BE SET IN CMD.8A
                      2058  *
                      2059  *      ENTRY  (BC) = POINTER TO DARA ENTRY
                      2060
                      2061
051.353               2062  DRV    EQU   *
051.353  032          2063         LDAX  D
051.354  315 064 054  2064         CALL  $TYPC.         TYPE REGISTER NAME
051.357  315 060 054  2065  DRV.   CALL  $TYPCH
051.362  075          2066         DB    '='
051.363  315 322 051  2067         CALL  DRA            DETERMINE ADDRESS
051.366  346 200      2068         ANI   200Q
051.370  062 023 045  2069         STA   CMD.8A+1       SET NON-ZERO IF DOUBLE
```

```
                                2071  **      DVB - DISPLAY VALUE WITH BLANK.
                                2072  *
                                2073  *       DVB CALLS 'FVD', AND THEN FOLLOWS WITH A BLANK.
                                2074
                                2075
051.373  315 040 052           2076  DVB     CALL    FVD
051.376  076 040               2077          MVI     A,' '
052.000  303 064 054           2078          JMP     $TYPC.          TYPE BLANK



                                2080  **      FBT - FIND BREAKPOINT IN TABLE.
                                2081  *
                                2082  *       ENTRY   (BC) = ADDRESS
                                2083  *       EXIT    (DE) = BKPT TABLE ADDRESS
                                2084  *               'Z' SET IF FOUND
                                2085  *       USES    A,F
                                2086
                                2087
052.003  021 140 057           2088  FBT     LXI     D,BKPTAB
052.006  345                   2089          PUSH    H
052.007  046 010               2090          MVI     H,BKPTBL
                                2091
052.011  032                   2092  FBT1    LDAX    D
052.012  251                   2093          XRA     C
052.013  302 025 052           2094          JNZ     FBT2            IF NO MATCH
052.016  023                   2095          INX     D
052.017  032                   2096          LDAX    D
052.020  033                   2097          DCX     D
052.021  250                   2098          XRA     B
052.022  312 036 052           2099          JZ      FBT3            BOTH MATCH: FOUND IT
                                2100
                                2101  *       CHECK NEXT ENTRY
                                2102
052.025  023                   2103  FBT2    INX     D
052.026  023                   2104          INX     D
052.027  023                   2105          INX     D
052.030  023                   2106          INX     D
052.031  045                   2107          DCR     H
052.032  302 011 052           2108          JNZ     FBT1            IF MORE TO GO
052.035  262                   2109          ORA     D               CLEAR 'Z', NOT FOUND
052.036  341                   2110  FBT3    POP     H
052.037  311                   2111          RET                     EXIT



                                2113  **      FVD - FORMAT VALUE FOR DISPLAY.
                                2114  *
                                2115  *       FVD FORMATS THE SPECIFIED BYTE (OR DOUBLE-BYTE) AS SPECIFIED,
                                2116  *       AND ADDS IT TO THE LINE BEING BUILT.
                                2117  *
                                2118  *       IF NO FORMAT IS SPECIFIED, *OCTAL BYTE* IS USED.
                                2119  *
                                2120  *       IF A LINE IS LARGE ENOUGH ALREADY, IT IS TYPED AND
```

```
                                2121  *       A NEW LINE IS STARTED.
                                2122  *
                                2123  *       ENTRY   (HL) = ADDRESS OF VALUE
                                2124  *       EXIT    (HL) ADVANCED
                                2125
                                2126
        052.040                 2127  FVD     EQU     *
        052.040 315 152 051     2128          CALL    CLL             CHECK LINE LENGTH
                                2129
                                2130  *       OUTPUT LEADING BLANK
                                2131
        052.043 325             2132          PUSH    D               SAVE (DE)
        052.044 345             2133          PUSH    H               SAVE (HL)
        052.045 072 022 045     2134          LDA     CMD.8A
        052.050 041 121 052     2135          LXI     H,FVDA
        052.053 247             2136          ANA     A                                       /78.10.GC/
        052.054 312 066 052     2137          JZ      FVD0.1                                  /78.10.GC/
        052.057 315 277 053     2138          CALL    *TBLS           FIND IN TABLE
        052.062 126             2139          MOV     D,M             (D) = PROCESSOR INDEX
        052.063 303 067 052     2140          JMP     FVD0.2                                  /78.10.GC/
                                2141
        052.066 127             2142  FVD0.1  MOV     D,A                                     /78.10.GC/
                                2143
        052.067 041 117 052     2144  FVD0.2  LXI     H,FVD1                                  /78.10.GC/
        052.072 343             2145          XTHL                    SET RETURN ADDRESS, RESTORE (HL)
        052.073 072 023 045     2146          LDA     CMD.8A+1        (A) = SINGLE/DOBLE FLAG
        052.076 247             2147          ANA     A               'Z' SET IF SINGLE BYTE
        052.077 365             2148          PUSH    PSW
        052.100 172             2149          MOV     A,D             (A) = FORMAT INDEX
        052.101 126             2150          MOV     D,M             (D) = 1ST VALUE
        052.102 043             2151          INX     H
        052.103 312 111 052     2152          JZ      FVD0            IF ONLY ONE BYTE
        052.106 132             2153          MOV     E,D             (E) = 2ND VALUE
        052.107 126             2154          MOV     D,M
        052.110 043             2155          INX     H
        052.111 315 076 031     2156  FVD0    CALL    *TBRA           BRANCH TO PROCESSOR
                                2157
        052.114 012             2158          DB      FVD,8-*         OCTAL
        052.115 023             2159          DB      FVD,D-*         DECIMAL
        052.116 040             2160          DB      FVD,A-*         ASCII
                                2161
                                2162
        052.117 321             2163  FVD1    POP     D               RESTORE (DE)
        052.120 311             2164          RET
                                2165
                                2166
        052.121 104 001         2167  FVDA    DB      'D',1           DECIMAL
        052.123 101 002         2168          DB      'A',2           ASCII
        052.125 000             2169          DB      0               OCTAL
```

```
                                2171  **      FVD.Q - TYPE OCTAL VALUE.
                                2172
052.126  172                   2173  FVD.Q   MOV     A,D
052.127  315 032 054           2174          CALL    $TOD            TYPE OCTAL DIGITS
052.132  361                   2175          POP     PSW
052.133  310                   2176          RZ                      IF ONLY 1 BYTE
052.134  173                   2177          MOV     A,E
052.135  303 032 054           2178          JMP     $TOD            TYPE OCTAL DIGITS


                                2180  **      FVD.D - TYPE DECIMAL VALUE.
                                2181
052.140  361                   2182  FVD.D   POP     PSW
052.141  076 005               2183          MVI     A,5             ASSUME 5 DIGITS
052.143  302 153 052           2184          JNZ     FVD.D1
052.146  132                   2185          MOV     E,D
052.147  026 000               2186          MVI     D,0
052.151  076 003               2187          MVI     A,3             3 DIGITS
052.153  303 332 053           2188  FVD.D1  JMP     $TDD            TYPE DECIMAL DIGITS


                                2190  **      FVD.A - TYPE ASCII VALUE
                                2191
052.156  172                   2192  FVD.A   MOV     A,D
052.157  315 013 054           2193          CALL    $TPA            TYPE PRINTING ASCII
052.162  361                   2194          POP     PSW
052.163  310                   2195          RZ                      EXIT IF SINGLE
052.164  173                   2196          MOV     A,E
052.165  303 013 054           2197          JMP     $TPA            TYPE PRINTING ASCII


                                2199  **      RAS - RESOLVE ADDRESS SPECIFICATION.
                                2200  *
                                2201  *       ENTRY   (HL) = CMD.AA
                                2202  *               (A) = ODD IF ADDRESS PAIR SPECIFIED
                                2203  *       EXIT    (DE) = LWA
                                2204  *               (HL) = FWA
                                2205
                                2206
052.170                         2207  RAS     EQU     *
052.170  365                   2208          PUSH    PSW             SAVE (A)
052.171  315 123 051           2209          CALL    CEA             COMPUTE EFFECTIVE ADDRESS
052.174  353                   2210          XCHG                    (DE) = FWA
052.175  041 034 045           2211          LXI     H,CMD.AA+4
052.200  361                   2212          POP     PSW
052.201  037                   2213          RAR
052.202  322 223 052           2214          JNC     RAS1            IF DOUBLE ADDRESS SPECIFICATION
                                2215
                                2216  *       ADDR-ADDR
                                2217
```

```
052.205  315 123 051  2218          CALL    CEA             COMPUTE EFFECTIVE ADDRESS
052.210  353         2219          XCHG                    (HL) = FWA, (DE) = LWA
052.211  173         2220          MOV     A,E
052.212  225         2221          SUB     L               COMPARE TWO ADDRESSES
052.213  172         2222          MOV     A,D
052.214  234         2223          SBB     H
052.215  332 271 047  2224          JC      DMP1            FIRST > LAST
052.220  303 233 052  2225          JMP     RAS2
                      2226
                      2227  *       ADDRL/CNTJ
                      2228
052.223  053         2229  RAS1    DCX     H
052.224  176         2230          MOV     A,M             (A) = (CMD.AA+3)
052.225  075         2231          DCR     A
052.226  157         2232          MOV     L,A
052.227  046 000      2233          MVI     H,0             (HL) = LENGTH SPECIFIED (0 IF NONE)
052.231  031         2234          DAD     D               (HL) = LWA
052.232  353         2235          XCHG
                      2236
052.233  042 103 057  2237  RAS2    SHLD    TOPVAL
052.236  353         2238          XCHG
052.237  043         2239          INX     H
052.240  042 105 057  2240          SHLD    BOTVAL
052.243  053         2241          DCX     H
052.244  353         2242          XCHG
052.245  311         2243          RET


                      2245  **      RBM - REMOVE BREAKPOINT FROM MEMORY.
                      2246  *
                      2247  *       RBM REMOVES SET BREAKPOINTS FROM MEMORY, BY RESTOREING THE
                      2248  *       ORIGINAL VALUES.
                      2249
                      2250
052.246  001 140 057  2251  RBM     LXI     B,BKPTAB
052.251  026 011      2252          MVI     D,BKPTBL+1
052.253  072 206 057  2253          LDA     BKPFLG
052.256  247         2254          ANA     A
052.257  310         2255          RZ                      NO BREAKPOINTS SET
052.260  363         2256          DI                      NO CTL-B WHILE SETTING BREAKPOINTS
                      2257
052.261  012         2258  RBM1    LDAX    B
052.262  157         2259          MOV     L,A
052.263  003         2260          INX     B
052.264  012         2261          LDAX    B
052.265  147         2262          MOV     H,A             (HL) = ADDRESS OF BKPT
052.266  003         2263          INX     B
052.267  003         2264          INX     B
                      2265
                      2266  *       RESTORE ORIGINAL VALUE
                      2267
052.270  012         2268          LDAX    B               (A) = VALUE
052.271  167         2269          MOV     M,A             SET IN MEMORY
052.272  003         2270          INX     B
```

```
052.273  025              2271           DCR     D
052.274  302 261 052      2272           JNZ     RBM1            IF MORE IN TABLE
052.277  373              2273           EI                      RESTORE INTERRUPTS
052.300  311              2274           RET


                          2276   **      RFD - RESTORE FRONT PANEL DISPLAY.
                          2277   *
                          2278   *       RFD IS CALLED TO RESTORE THE .CTLFLG OPTIONS STORED IN
                          2279   *       RFDA.
                          2280   *
                          2281   *       ENTRY   *RFDA* = CTLFLG VALUE
                          2282   *       EXIT    .CTLFLG RESTORED
                          2283   *       USES    A
                          2284
                          2285
052.301  076 000          2286   RFD     MVI     A,0
052.302                   2287   RFDA    EQU     *-1
052.303  062 011 040      2288           STA     .CTLFLG
052.306  323 360          2289           OUT     OP.CTL


                          2291   **      RUC - RESTORE USER CONSOLE INVIRONMENT.
                          2292   *
                          2293   *       RUC RESTORES THE USER CONSOLE FLAGS.
                          2294   *
                          2295   *       ENTRY   NONE
                          2296   *       EXIT    NONE
                          2297   *       USES    A,F
                          2298
                          2299
052.310  072 107 057      2300   RUC     LDA     CSLMD
052.313  062 326 040      2301           STA     S.CSLMD         STORE USER CONSOLE MODE
052.316  072 110 057      2302           LDA     CONFL
052.321  062 332 040      2303           STA     S.CONFL         STORE CONSOLE FLAGS
052.324  311              2304           RET
052.325  311              2305           RET


                          2307   **      SBL - SET BREAKPOINT LIST.
                          2308   *
                          2309   *       SBL IS CALLED TO SET A LIST OF BREAKPOINTS INTO THE TABLE.
                          2310   *
                          2311   *       ENTRY   (CMD.DA) = BREAKPOINTS
                          2312   *       EXIT    SET IN TABLE
                          2313
                          2314
052.326                   2315   SBL     EQU     *               CALLED AS SUBROUTINE
052.326  041 040 045      2316           LXI     H,CMD.DA
                          2317
```

```
                                2318  *       EXAMINE NEXT BREAKPOINT
                                2319
052.331  176                   2320  SBL1    MOV     A,M             (A) = OPTION
052.332  017                   2321          RRC
052.333  330                   2322          RC                      IF END OF LIST
052.334  043                   2323          INX     H
                                2324
                                2325  *       FIND BREAKPOINT ALREADY IN LIST, OR EMPTY SPOT
                                2326
052.335  116                   2327          MOV     C,M
052.336  043                   2328          INX     H
052.337  106                   2329          MOV     B,M             (BC) = ADDRESS
052.340  315 003 052           2330          CALL    FBT             FIND BREAKPOINT IN TABLE
052.343  312 361 052           2331          JE      SBL2            IF FOUND
052.346  305                   2332          PUSH    B
052.347  001 000 000           2333          LXI     B,0
052.352  315 003 052           2334          CALL    FBT             FIND EMPTY SPOT
052.355  302 376 052           2335          JNE     SBL3            NO SPACE
052.360  301                   2336          POP     B
                                2337
                                2338  *       HAVE SPOT. STORE VALUE
                                2339
052.361  353                   2340  SBL2    XCHG
052.362  161                   2341          MOV     M,C             SET VALUE IN TAL
052.363  043                   2342          INX     H
052.364  160                   2343          MOV     M,B
052.365  023                   2344          INX     D               (DE) = ADDRESS OF REPEAT COUNT
052.366  032                   2345          LDAX    D
052.367  043                   2346          INX     H
052.370  167                   2347          MOV     M,A             SET REPEAT COUNT
052.371  353                   2348          XCHG
052.372  043                   2349          INX     H
052.373  303 331 052           2350          JMP     SBL1            PROCESS NEXT BREAKPOINT
                                2351
                                2352  *       OUT OF SPACE
                                2353
052.376  315 136 031           2354  SBL3    CALL    $TYPTX
053.001  007 116 117           2355          DB      BELL,'NO ROOM',ENL
053.012  303 225 045           2356          JMP     TBGX


                                2358  **      SBM - SET BREAKPOINT IN MEMORY.
                                2359  *
                                2360  *       SBM SETS THE BREAKPOINT INSTRUCTIONS IN MEMORY PREPARATORY
                                2361  *       TO EXECUTION.
                                2362
                                2363
053.015  001 140 057           2364  SBM     LXI     B,BKPTAB
053.020  026 011               2365          MVI     D,BKPTBL+1
053.022  072 206 057           2366          LDA     BKPFLG
053.025  247                   2367          ANA     A
053.026  300                   2368          RNZ                     ALREADY IN MEMORY
053.027  363                   2369          DI                      NO INTERRUPTS WHILE SETTING
                                2370
```

```
053.030  012            2371  SBM1    LDAX    B
053.031  157            2372          MOV     L,A
053.032  003            2373          INX     B
053.033  012            2374          LDAX    B
053.034  147            2375          MOV     H,A
053.035  003            2376          INX     B
053.036  003            2377          INX     B
                        2378
                        2379  *       SET IT
                        2380
053.037  176            2381          MOV     A,M             (A) = INSTRUCTION TO BE SAVED
053.040  002            2382          STAX    B
053.041  066 327        2383          MVI     M,MI.BKP        SET BREAKPOINT
053.043  003            2384  SBM2    INX     B
053.044  025            2385          DCR     D
053.045  302 030 053    2386          JNZ     SBM1            IF MORE TO CHECK
053.050  373            2387          EI                      RESTORE INTERRUPTS
053.051  311            2388          RET                     EXIT


                        2390  **      SDC - SET DEBUGGER CONSOLE INVIRONMENT.
                        2391  *
                        2392  *       SDC SAVES THE USER'S CONSOLE CONTROL FLAGS, AND INSTUTITES
                        2393  *       HBUG'S
                        2394  *
                        2395  *       ENTRY   NONE
                        2396  *       EXIT    NONE
                        2397  *       USES    A,F,H,L
                        2398
                        2399
053.052  041 326 040    2400  SDC     LXI     H,S.CSLMD
053.055  176            2401          MOV     A,M
053.056  062 107 057    2402          STA     CSLMD           CLEAR CONSOLE MODE
053.061  066 201        2403          MVI     M,CSL.ECH+CSL.CHR       SET NO ECHO, CHAR MODE
053.063  056 332        2404          MVI     L,#S.CONFL
000.040                 2405  .       SET     S.CSLMD/256
000.000                 2406          ERRNZ   S.CONFL/256-.   MUST BE IN SAME PAGE
053.065  176            2407          MOV     A,M
053.066  062 110 057    2408          STA     CONFL           SAVE USER CONSOLE FLAGS
053.071  066 000        2409          MVI     M,0             CLEAR FLAGS
053.073  311            2410          RET


                        2412  **      SSA - SET STARTING ADDRESS.
                        2413  *
                        2414  *       SSA SETS AN ENTERED VALUE INTO THE USER PROGRAM PC REGISTER.
                        2415  *
                        2416  *       ENTRY   (HL) = ADDRES S OF VALUE BLOCK
                        2417  *       EXIT    ADDRESS SET.
                        2418
                        2419
053.074  176            2420  SSA     MOV     A,M             (A) = DEFAULT OPTION
```

```
053.075  017            2421        RRC
053.076  330            2422        RC              IF DEFAULT
053.077  315 123 051    2423        CALL    CEA     COMPUTE EFFECTIVE ADDRESS
053.102  104            2424        MOV     B,H
053.103  115            2425        MOV     C,L
053.104  315 317 051    2426        CALL    DRA.    DETERMINE ADDRESS
053.107  161            2427        MOV     M,C
053.110  043            2428        INX     H
053.111  160            2429        MOV     M,B
053.112  311            2430        RET             EXIT
```

```
053.113                2433          XTEXT   MOVE


                       2435X **      $MOVE - MOVE DATA
                       2436X *
                       2437X *       $MOVE MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
                       2438X *       IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
                       2439X *       FIRST TO LAST.
                       2440X *
                       2441X *       IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM
                       2442X *       LAST TO FIRST.
                       2443X *
                       2444X *       THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.
                       2445X *
                       2446X *       ENTRY   (BC) = COUNT
                       2447X *               (DE) = FROM
                       2448X *               (HL) = TO
                       2449X *       EXIT    MOVED
                       2450X *               (DE) = ADDRESS OF NEXT FROM BYTE
                       2451X *               (HL) = ADDRESS OF NEXT *TO* BYTE
                       2452X *               'C' CLEAR
                       2453X *       USES    ALL
                       2454X
                       2455X
030.252                2456X $MOVE   EQU     30252A          IN H17 ROM
053.113                2457          XTEXT   SAVALL


                       2459X **      $RSTALL - RESTORE ALL REGISTERS.
                       2460X *
                       2461X *       $RSTALL RESTORES ALL THE REGISTERS OFF THE STACK, AND
                       2462X *       RETURNS TO THE PREVIOUS CALLER.
                       2463X *
                       2464X *       ENTRY   (SP) = PSW
                       2465X *               (SP+2) = BC
                       2466X *               (SP+4) = DE
                       2467X *               (SP+6) = HL
                       2468X *               (SP+8) = RET
                       2469X *       EXIT    TO *RET*, REGISTERS RESTORED
                       2470X *       USES    ALL
                       2471X
                       2472X
031.047                2473X $RSTALL EQU     31047A          IN H17 ROM
```

```
                                2475X **      $SAVALL - SAVE ALL REGISTERS ON STACK.
                                2476X *
                                2477X *       $SAVALL SAVES ALL THE REGISTERS ON THE STACK.
                                2478X *
                                2479X *       ENTRY   NONE
                                2480X *       EXIT    (SP) = PSW
                                2481X *               (SP+2) = BC
                                2482X *               (SP+4) = DE
                                2483X *               (SP+6) = HL
                                2484X *       USES    H,L
                                2485X
                                2486X
      031.054                   2487X $SAVALL EQU     31054A          IN H17 ROM
      053.113                   2488          XTEXT   MCU


                                2490X **      MCU - MAP LOWER CASE TO UPPER CASE.
                                2491X *
                                2492X *       MCU MAPS A LOWER CASE ALPHABETIC TO UPPER
                                2493X *       CASE.
                                2494X *
                                2495X *       ENTRY   (A) = CHARACTER
                                2496X *       EXIT    (A) = CHARACTER RESULT
                                2497X *       USES    A,F
                                2498X
                                2499X
      053.113  376 141          2500X $MCU    CPI     'a'
      053.115  330              2501X         RC                      NOT LOWER CASE
      053.116  376 173          2502X         CPI     'z'+1
      053.120  320              2503X         RNC                     NOT LOWER CASE
      053.121  326 040          2504X         SUI     'a'-'A'
      053.123  311              2505X         RET
      053.124                   2506          XTEXT   INDL


                                2508X **      $INDL - INDEXED LOAD.
                                2509X *
                                2510X *       $INDL LOADS DE WITH THE TWO BYTES AT (HL)+DISPLACMENT
                                2511X *
                                2512X *       THIS ACTS AS AN INDEXED FULL WORD LOAD.
                                2513X *
                                2514X *       (DE) = ( (HL) + DSPLACEMENT )
                                2515X *
                                2516X *       ENTRY   ((RET)) = DISPLACMENT (FULL WORD)
                                2517X *               (HL) = TABLE ADDRESS
                                2518X *       EXIT    TO (RET+2)
                                2519X *       USES    A,F,D,E
                                2520X
                                2521X
      030.234                   2522X $INDL   EQU     30234A          IN H17 ROM
      053.124                   2523          XTEXT   HLIHL
```

```
                        2525X **      $HLIHL - LOAD HL INDIRECT THROUGH HL.
                        2526X *
                        2527X *       (HL) = ((HL))
                        2528X *
                        2529X *       ENTRY   NONE
                        2530X *       EXIT    NONE
                        2531X *       USES    A,H,L
                        2532X
030.211                 2533X $HLIHL  EQU     30211A          IN H17 ROM
053.124                 2534          XTEXT   TYPTX



                        2536X **      $TYPTX - TYPE TEXT.
                        2537X *
                        2538X *       $TYPTX IS CALLED TO TYPE A BLOCK OF TEXT ON THE SYSTEM CONSOLE.
                        2539X *
                        2540X *       IMBEDDED ZERO BYTES INDICATE A CARRIAGE RETURN LINE FEED,
                        2541X *       A BYTE WITH THE 200Q BIT SET IS THE LAST BYTE IN THE MESSAGE.
                        2542X *
                        2543X *       ENTRY   (RET) = TEXT
                        2544X *       EXIT    TO (RET+LENGTH)
                        2545X *       USES    A,F
                        2546X
                        2547X
031.136                 2548X $TYPTX  EQU     31136A          IN H17 ROM
                        2549X
031.144                 2550X $TYPTX. EQU     31144A          IN H17 ROM
053.124                 2551          XTEXT   RCHAR



                        2553X **      $RCHAR - READ SINGLE CHARACTER FROM CONSOLE.
                        2554X *
                        2555X *       ENTRY   NONE
                        2556X *       EXIT    (A) = CHARACTER
                        2557X *       USES    A,F
                        2558X
                        2559X
053.124  377 001       2560X $RCHAR  DB      SYSCALL+.SCIN
053.126  332 124 053   2561X         JC      $RCHAR          NOT READY
053.131  311           2562X         RET
                        2563X
053.132  377 002       2564X $WCHAR  DB      SYSCALL+.SCOUT
053.134  311           2565X         RET
053.135                 2566          XTEXT   CRLF
```

```
                                2568X **      $CRLF - TYPE CARRIAGE RETURN/ LINE FEED
                                2569X *
                                2570X *       $CRLF IS USED TO GENERATE PADDED CRLF'S.
                                2571X *
                                2572X *       ENTRY   NONE
                                2573X *       EXIT    (A) = 0
                                2574X *       USES    A,F
                                2575X
                                2576X
          053.135  076 012      2577X $CRLF   MVI     A,NL
          053.137  377 002      2578X         DB      SYSCALL,.SCOUT
          053.141  257          2579X         XRA     A
          053.142  311          2580X         RET
          053.143               2581          XTEXT   DADA


                                2583X **      $DADA - PERFORM (H,L) = (H,L) + (0,A)
                                2584X *
                                2585X *       ENTRY   (H,L) = BEFORE VALUE
                                2586X *               (A) = BEFORE VALUE
                                2587X *       EXIT    (H,L) = (H,L) + (0,A)
                                2588X *               (C) SET IF OVERFLOW
                                2589X *       USES    F,H,L
                                2590X
                                2591X
          030.072               2592X $DADA   EQU     30072A          IN H17 ROM
          053.143               2593          XTEXT   DADA2


                                2595X **      $DADA. - ADD (0,A) TO (H,L)
                                2596X *
                                2597X *       ENTRY   NONE
                                2598X *       EXIT    (HL) = (HL) + (0A)
                                2599X *       USES    A,F,H,L
                                2600X
                                2601X
          030.101               2602X $DADA.  EQU     30101A          IN H17 ROM
          053.143               2603          XTEXT   INCHA


                                2605X **      $INCHA - READ ONE CHARACTER.
                                2606X *
                                2607X *       $INCHA READS ONE CHARACTER FROM THE TERMINAL.
                                2608X *
                                2609X *       CHAR = CTL-U: ERASE LINE
                                2610X *            = BKSP: BACKSPACE CHARACTER
                                2611X *              = RUBOUT: BACKSPACE CHARACTER
                                2612X
                                2613X ************************************************************************8
                                2614X **
```

```
P  000.001                2615X          ERRNZ   1               THIS ROUTINE IS OBSOLETE
                          2616X
                          2617X **************************************************************************
                          2618X
                          2619X
           053.143  315 124 053   2620X $INCHA  CALL    $RCHAR          READ A CHARACTER
           053.146  376 010       2621X         CPI     BKSP
           053.150  312 211 053   2622X         JE      INC0            IS BKSP
           053.153  376 177       2623X         CPI     RUBOUT
           053.155  312 211 053   2624X         JE      INC0            IS RUBOUT
           053.160  365           2625X         PUSH    PSW             SAVE CODE
           053.161  072 276 053   2626X         LDA     $INCHAA         (A) = RUBOUT FLAG
           053.164  247           2627X         ANA     A
           053.165  304 132 053   2628X         CNZ     $WCHAR          ECHO RUBOUT CHAR, IF ANY
           053.170  257           2629X         XRA     A
           053.171  062 276 053   2630X         STA     $INCHAA         CLEAR FLAG
           053.174  361           2631X         POP     PSW
           053.175  376 025       2632X         CPI     'U'-'@'
           053.177  300           2633X         RNE                     NOT CTL-U, RETURN
                                  2634X
                          2635X *       IS CTL-U
                          2636X
           053.200  041 312 044   2637X         LXI     H,LINE
           053.203  315 135 053   2638X         CALL    $CRLF
           053.206  303 240 053   2639X         JMP     INC1            CLEAR LINE AND SET LINPTR
                                  2640X
                          2641X *       IS BKSP
                          2642X
           053.211  052 020 045   2643X INC0    LHLD    LINPTR
           053.214  076 312       2644X         MVI     A,$LINE
           053.216  275           2645X         CMP     L
           053.217  312 143 053   2646X         JE      $INCHA          IF ALREADY AT FRONT
           053.222  053           2647X         DCX     H
           053.223  072 327 040   2648X         LDA     S.CONTY         SEE IF BACKSPACING
           053.226  247           2649X         ANA     A
           053.227  362 250 053   2650X         JP      INC3            IS NON-CRT
           053.232  315 136 031   2651X         CALL    $TYPTX
           053.235  010 040 210   2652X         DB      BKSP,' ',BKSP+2000      BACKSPACE FOR CRT
           053.240  042 020 045   2653X INC1    SHLD    LINPTR
           053.243  066 000       2654X         MVI     M,0             CLEAR ENTRY
           053.245  303 143 053   2655X         JMP     $INCHA          AGAIN
                                  2656X
                          2657X *       BACKSPACE FOR NON-CRT
                          2658X
           053.250  072 276 053   2659X INC3    LDA     $INCHAA         (A) = FLAG
           053.253  247           2660X         ANA     A
           053.254  302 267 053   2661X         JNZ     INC4            AM STILL BACKSPACING
           053.257  076 057       2662X         MVI     A,'/'
           053.261  062 276 053   2663X         STA     $INCHAA         SET FLAG
           053.264  315 132 053   2664X         CALL    $WCHAR          TYPE
           053.267  176           2665X INC4    MOV     A,M
           053.270  315 132 053   2666X         CALL    $WCHAR          SHOW CHARACTER BEING REMOVED
           053.273  303 240 053   2667X         JMP     INC1            CLEAR IT
                                  2668X
           053.276  000           2669X $INCHAA DB      0               RUBOUT FLAG
           053.277                2670          XTEXT   MU86
```

```
                                2672X **      $MU86 - MULTIPLY 8X16 UNSIGNED.
                                2673X *
                                2674X *       $MU86 MULTIPLIES A 16 BIT VALUE BY A 8
                                2675X *       BIT VALUE.
                                2676X *
                                2677X *       ENTRY   (A) = MULTIPLIER
                                2678X *               (DE) = MULTIPLICAND
                                2679X *       EXIT    (HL) = RESULT
                                2680X *               'Z' SET IF NOT OVERFLOW
                                2681X *       USES    A,F,H,L
                                2682X
                                2683X
    031.007                     2684X $MU86   EQU     31007A          IN H17 ROM
    053,277                     2685          XTEXT   TBLS


                                2687X **      $TBLS - TABLE SEARCH
                                2688X *
                                2689X *       TABLE FORMAT
                                2690X *
                                2691X *       DB      KEY1,VAL1,
                                2692X *       .       .
                                2693X *       .       .
                                2694X *       DB      KEYN,VALN
                                2695X *       DB      0
                                2696X *
                                2697X *       ENTRY   (A) = PATTERN
                                2698X *               (H,L) = TABLE FWA
                                2699X *       EXIT    (A) = PATTERN IF FOUND
                                2700X *               'Z' SET IF FOUND
                                2701X *               'Z' CLEAR IF NOT FOUND OR PATTERN=0          /78.10.GC/
                                2702X *       USES    A,F,H,L
                                2703X
                                2704X
    053.277  305               2705X $TBLS   PUSH    B
    053.300  376 000           2706X         CPI     0                                            /78.10.GC/
    053.302  312 324 053       2707X         JZ      TBL2                                         /78.10.GC/
    053,305  107               2708X         MOV     B,A
    053.306  176               2709X TBL1    MOV     A,M             (A) = CHARACTER
    053,307  043               2710X         INX     H
    053.310  270               2711X         CMP     B
    053.311  312 326 053       2712X         JZ      TBL3            IF MATCH
    053.314  247               2713X         ANA     A
    053.315  043               2714X         INX     H               SKIP PAST
    053.316  302 306 053       2715X         JNZ     TBL1            IF NOT END OF TABLE
    053.321  053               2716X         DCX     H
    053.322  053               2717X         DCX     H
    053.323  257               2718X         XRA     A               SET TO ZERO FOR OLD USERS    /78.10.GC/
    053.324  376 001           2719X TBL2    CPI     1               CLEAR ZERO                   /78.10.GC/
                                2720X
                                2721X *       DONE
                                2722X
    053.326  301               2723X TBL3    POP     B
    053.327  311               2724X         RET
```

```
053.330                 2725            XTEXT   TJMP


                        2727X **        $TJMP - TABLE JUMP.
                        2728X *
                        2729X *         USAGE
                        2730X *
                        2731X *         CALL    $TJMP           (A) = INDEX
                        2732X *         DW      ADDR1
                        2733X *         .         .
                        2734X *         .         .
                        2735X *         .         .
                        2736X *         DW      ADDRN
                        2737X *
                        2738X *         ENTRY   (A) = INDEX
                        2739X *         EXIT    TO PROCESSOR
                        2740X *                 (A) = INDEX*2
                        2741X *         USES    NONE.
                        2742X
                        2743X
031.061                 2744X $TJMP     EQU     31061A          IN H17 ROM, (A) = INDEX*2
                        2745X
031.062                 2746X $TJMP.    EQU     31062A          IN H17 ROM
053.330                 2747            XTEXT   TDD


                        2749X **        $TDD - TYPE DECIMAL DIGITS.
                        2750X *
                        2751X *         $TDD TYPES A 16 BIT VALUE AS 1 TO 5 DECIMAL DIGITS.
                        2752X *
                        2753X *         ENTRY   (D,E) = VALUE
                        2754X *                 (A) = DIGIT COUNT
                        2755X *         EXIT    VALUE TYPED.
                        2756X *         USES    A,B,C,F
                        2757X
                        2758X
053.330  076 005       2759X $TDD.     MVI     A,5
053.332  345           2760X $TDD      PUSH    H
053.333  365           2761X TDD1      PUSH    PSW
053.334  041 377 053   2762X           LXI     H,TDDA-2
053.337  007           2763X           RLC                     (A) = DIGIT NUMBER*2
053.340  315 101 030   2764X           CALL    $DADA.
053.343  176           2765X           MOV     A,M
053.344  043           2766X           INX     H
053.345  146           2767X           MOV     H,M
053.346  157           2768X           MOV     L,A             (HL) = MULTIPLE OF 10
053.347  353           2769X           XCHG                    (DE) = DEVISOR, (HL) = VALUE
053.350  076 377       2770X           MVI     A,377Q
053.352  031           2771X TDD2      DAD     D
053.353  074           2772X           INR     A
053.354  332 352 053   2773X           JC      TDD2            IF MORE TO GO
053.357  306 060       2774X           ADI     '0'
```

```
053.361  315 064 054  2775X          CALL    $TYPC.          TYPE DIGIT
053.364  175         2776X          MOV     A,L
053.365  223         2777X          SUB     E
053.366  137         2778X          MOV     E,A             REMOVE EXTRA SUBTRACTION
053.367  174         2779X          MOV     A,H
053.370  232         2780X          SBB     D
053.371  127         2781X          MOV     D,A
053.372  361         2782X          POP     PSW
053.373  075         2783X          DCR     A
053.374  302 333 053  2784X          JNZ     TDD1            IF MORE DIGITS
053.377  341         2785X          POP     H
054.000  311         2786X          RET                     EXIT
                     2787X
054.001              2788X TDDA     EQU     *
054.001  377 377     2789X          DW      -1
054.003  366 377     2790X          DW      -10
054.005  234 377     2791X          DW      -100
054.007  030 374     2792X          DW      -1000
054.011  360 330     2793X          DW      -10000
054.013              2794           XTEXT   TEA


                     2796X **       $TPA - TYPE PRINTING ASCII.
                     2797X *
                     2798X *        $TPA TYPES AN ASCII CHARACTER. ALL NON-PRINTING CHARACTERS
                     2799X *        ARE TYPED AS BLANKS.
                     2800X *
                     2801X *        ENTRY   (A) = CHARACTER
                     2802X *        EXIT    TYPED
                     2803X *        USES    A,F
                     2804X
                     2805X
054.013  376 040     2806X $TPA     CPI     40Q
054.015  372 025 054  2807X          JM      TPA1            IF BAD
054.020  376 177     2808X          CPI     177Q
054.022  332 044 054  2809X          JC      $TYPC.          OK, TYPE AND RETURN
054.025  076 040     2810X TPA1     MVI     A,' '
054.027  303 044 054  2811X          JMP     $TYPC.          TYPE AND REGUTN
054.032              2812           XTEXT   TBRA


                     2814X **       $TBRA - BRANCH RELATIVE THOUGH TABLE.
                     2815X *
                     2816X *        $TBRA USES THE SUPPLIED INDEX TO SELECT A BYTE FROM THE
                     2817X *        JUMP TABLE. THE CONTENTS OF THIS BYTE ARE ADDED TO THE
                     2818X *        ADDRESS OF THE BYTE, YEILDING THE PROCESSOR ADDRESS.
                     2819X *
                     2820X *        CALL    $TBRA
                     2821X *        DB      LAB1-*          INDEX = 0 FOR LAB1
                     2822X *        DB      LAB2-*          INDEX = 1 FOR LAB2
                     2823X *        DB      LABN-*          INDEX = N-1 FOR LABN
                     2824X *
```

```
                                2825X *       ENTRY   (A) = INDEX
                                2826X *               (RET) = TABLE FWA
                                2827X *       EXIT    TO COMPUTED ADDRESS
                                2828X *       USES    F,H,L
                                2829X
                                2830X
        031.076                 2831X $TBRA   EQU     31076A          IN H17 ROM
        054.032                 2832          XTEXT   TOD


                                2834X **      $TOD - TYPE OCTAL DIGITS.
                                2835X *
                                2836X *       $TOD TYPES AN OCTAL BYTE AS 3 OCTAL DIGITS, ZERO FILL.
                                2837X *
                                2838X *       ENTRY   (A) = VALUE
                                2839X *       EXIT    VALUE TYPES
                                2840X *       USES    A,F
                                2841X
                                2842X
        054.032  305            2843X $TOD    PUSH    B
        054.033  006 003       2844X         MVI     B,3
        054.035  247           2845X         ANA     A               CLEAR CARRY
                                2846X
        054.036  027           2847X TOD1    RAL
        054.037  027           2848X         RAL
        054.040  027           2849X         RAL
        054.041  365           2850X         PUSH    PSW
        054.042  346 007       2851X         ANI     7
        054.044  306 060       2852X         ADI     '0'
        054.046  315 064 054   2853X         CALL    $TYPC.          TYPE CHARACTER
        054.051  361           2854X         POP     PSW
        054.052  005           2855X         DCR     B
        054.053  302 036 054   2856X         JNZ     TOD1            IF MORE TO GO
        054.056  301           2857X         POP     B
        054.057  311           2858X         RET                     EXIT
        054.060                 2859          XTEXT   TYPCH


                                2861X **      $TYPCH - TYPE SINGLE CHARACTER.
                                2862X *
                                2863X *       ENTRY   (RET) = CHARACTER
                                2864X *       EXIT    TO (RET)+1
                                2865X *               (A) = CHARACTER TYPED
                                2866X
                                2867X
        054.060  343           2868X $TYPCH  XTHL                    (HL) = RETURN ADDRESS
        054.061  176           2869X         MOV     A,M             (A) = CHARACTER
        054.062  043           2870X         INX     H
        054.063  343           2871X         XTHL                    RESTORE ADVANCED EXIT ADDRESS
                                2872X
                                2873X **      $TYPC. - TYPE SINGLE CHARACTER.
                                2874X *
```

```
                        2875X *       ENTRY   (A) = CHARACTER
                        2876X *       EXIT    TO (RET)
                        2877X
054.064 377 002         2878X $TYPC.  DB      SYSCALL,.SCOUT
054.066 311             2879X         RET
054.067                 2880          XTEXT   ZERO


                        2882X **      $ZERO - ZERO MEMORY
                        2883X *
                        2884X *       $ZERO ZEROS A BLOCK OF MEMORY.
                        2885X *
                        2886X *       ENTRY   (HL) = ADDRESS
                        2887X *               (B) = COUNT
                        2888X *       EXIT    (A) = 0
                        2889X *       USES    A,B,F,H,L
                        2890X
                        2891X
031.212                 2892X $ZERO   EQU     31212A          IN H17 ROM
054.067                 2893          XTEXT   FOPE


                        2895X **      $FOPEx - OPEN FILE BLOCK FOR I/O
                        2896X *
                        2897X *       $FOPEx IS CALLED BEFORE ANY I/O IS DONE VIA A
                        2898X *       FILE BLOCK.  $FOPEx SETS UP THE FILE BLOCK, AND OPENS
                        2899X *       THE FILE VIA *HDOS*.
                        2900X *
                        2901X *       ENTRY   (DE) = ADDRESS OF DEFAULT BLOCK
                        2902X *               (HL) = ADDRESS OF FILE BLOCK
                        2903X *       EXIT    TO $FERROR IF ERROR
                        2904X *               TO CALLER IF OK
                        2905X *       USES    A,F,B,C,D,E
                        2906X
                        2907X
054.067 315 114 054     2908X $FOPER  CALL    $FOPER.
054.072 320             2909X         RNC
054.073 303 373 055     2910X         JMP     $FERROR         IN ERROR
                        2911X
054.076 315 117 054     2912X $FOPEW  CALL    $FOPEW.
054.101 320             2913X         RNC
054.102 303 373 055     2914X         JMP     $FERROR         IN ERROR
                        2915X
054.105 315 122 054     2916X $FOPEU  CALL    $FOPEU.
054.110 320             2917X         RNC
054.111 303 373 055     2918X         JMP     $FERROR         IN ERROR
                        2919X
                        2920X
054.114 076 002         2921X $FOPER. MVI     A,FT.OR         FILE TYPE OF OPEN FOR READ
054.116 001             2922X         DB      001Q            LXI,B TO SKIP NEXT MVI
054.117 076 004         2923X $FOPEW. MVI     A,FT.OW         OPEN FOR WRITE
054.121 001             2924X         DB      001Q            LXI,B TO SKIP NEXT MIV
```

```
054.122  076 006      2925X $FOPEU. MVI     A,FT.OR+FT.OW
                       2926X
                       2927X *       (A) = FILE FLAGS
                       2928X
054.124  345           2929X         PUSH    H               SAVE FILE BLOCK ADDRESS
054.125  365           2930X         PUSH    PSW             SAVE NEW FLAGS
000.000                2931X         ERRNZ   FB.CHA
054.126  106           2932X         MOV     B,M             (B) = CHANNEL NUMBER
054.127  305           2933X         PUSH    B               SAVE HANNEL NUMBER
000,000                2934X         ERRNZ   FB.FLG-FB.CHA-1
054.130  043           2935X         INX     H
054.131  117           2936X         MOV     C,A             (C) = NEW FILE FLAGS
054.132  176           2937X         MOV     A,M             (A) = CURRENT TYPE
054.133  247           2938X         ANA     A
054.134  171           2939X         MOV     A,C             (A) = NEW FLAGS TO BE SET
054.135  312 147 054   2940X         JZ      $FOPE1          NOT ALREADY OPEN
                       2941X
                       2942X *       ALREADY OPEN. SQUACK
                       2943X
054.140  301           2944X         POP     B               RESTORE (BC)
054.141  361           2945X         POP     PSW             DISCARD NEW FLAGS
054.142  341           2946X         POP     H               (HL) = FB ADDRESS
054.143  076 031       2947X         MVI     A,EC.FAO        FILE ALREADY OPEN
054.145  067           2948X         STC
054.146  311           2949X         RET
                       2950X
000.000                2951X         ERRNZ   FB.FWA-FB.FLG-1
054.147  043           2952X $FOPE1  INX     H               (HL) = $FB.FWA
054.150  116           2953X         MOV     C,M
054.151  043           2954X         INX     H
054.152  106           2955X         MOV     B,M             (BC) = FB.FWA
054.153  043           2956X         INX     H
000.000                2957X         ERRNZ   FB.PTR-FB.FWA-2
054.154  161           2958X         MOV     M,C             SET FB.PTR = FB.FWA
054.155  043           2959X         INX     H
054.156  160           2960X         MOV     M,B
054.157  043           2961X         INX     H
000.000                2962X         ERRNZ   FB.LIM-FB.PTR-2
054.160  161           2963X         MOV     M,C             SET FB.LIM = FB.FWA
054.161  043           2964X         INX     H
054.162  160           2965X         MOV     M,B
054.163  043           2966X         INX     H
000.000                2967X         ERRNZ   FB.NAM-FB.LIM-4
054.164  043           2968X         INX     H
054.165  043           2969X         INX     H               (HL) = $FB.NAM
                       2970X
                       2971X *       FILE BLOCK POINTERS SETUP. OPEN FILE
                       2972X
054.166  345           2973X         PUSH    H               SAVE NEW ADDRESS FOR NAME
054.167  041 220 054   2974X         LXI     H,$FOPEB
054.172  247           2975X         ANA     A                                       /78.10.GC/
054.173  312 202 054   2976X         JZ      $FOPE2
000.000                2977X         ERRNZ   .EXIT
054.176  315 277 053   2978X         CALL    $TBLS           FIND CODE
054.201  176           2979X         MOV     A,M
054.202  062 210 054   2980X $FOPE2  STA     $FOPEA          SET SYSCALL CODE
```

```
054.205  341          2981X          POP     H               (HL) = $FB.NAM
054.206  341          2982X          POP     PSW             (A) = CHANNEL NUMBER
054.207  377 000      2983X          DB      SYSCALL+.EXIT
054.210               2984X $FOPEA   EQU     *-1             SYSCALL CODE
054.211  321          2985X          POP     D               (D) = NEW FLAG
054.212  341          2986X          POP     H               (HL) = FILE BLOCK ADDRESS
054.213  330          2987X          RC                      EXIT IF ERROR
054.214  043          2988X          INX     H
000.000               2989X          ERRNZ   FB.FLG-1
054.215  162          2990X          MOV     M+D             SET NEW FLAGS
054.216  053          2991X          DCX     H               RESTORE (HL)
054.217  311          2992X          RET
                      2993X
054.220  002 042      2994X $FOPEB   DB      FT.OR+.OPENR    TABLE OF SYSCALL CODES
054.222  004 043      2995X          DB      FT.OW+.OPENW
054.224  006 044      2996X          DB      FT.OR+FT.OW+.OPENU
054.226  000          2997X          DB      0               SHOULD NOT OCCUR
054.227               2998           XTEXT   FREAB



                      3000X **       $FREAB - READ BYTES FROM FILE BUFFER.
                      3001X *
                      3002X *        $FREAB IS CALLED TO READ A NUMBER OF BYTES FROM A FILE BUFFER.
                      3003X *
                      3004X *        ENTRY   (BC) = BYTE COUNT
                      3005X *                (DE) = FWA FOR BYTES
                      3006X *                (HL) = ADDRESS OF FILE BUFFER
                      3007X *        EXIT    TO *FERROR* IF ERROR
                      3008X *                TO CALLER IF OK
                      3009X *                 (BC) = UNREAD BYTE COUNT (ONLY IF EOF)
                      3010X *                 (DE) = ADDRESS OF FIRST UNUSED BYTE
                      3011X *                'C' SET IF EOF DURING READ
                      3012X *        USES    A+F+B+C+D+E
                      3013X
                      3014X
054.227  315 242 054  3015X $FREAB   CALL    $FREAB+
054.232  320          3016X          RNC                     RETURN IF OK
054.233  376 001      3017X          CPI     EC+EOF
054.235  302 373 055  3018X          JNE     $FERROR         ERROR IS NOT EOF
054.240  067          3019X          STC
054.241  311          3020X          RET                     ERROR IS SIMPLY EOF
                      3021X
                      3022X
054.242               3023X $FREAB+  EQU     *
054.242  257          3024X          XRA     A
054.243  062 231 056  3025X          STA     EOFFLG          CLEAR EOF FLAG
054.246  345          3026X          PUSH    H
054.247  315 055 056  3027X          CALL    CBT             COPY BUFFER POINTERS TO TEMP CELLS
                      3028X
                      3029X *        COPY DATA FROM BUFFER TO TARGET
                      3030X
054.252  325          3031X $REAB2   PUSH    D               SAVE TARGET ADDRESS
054.253  072 220 056  3032X          LDA     T.FLG
054.256  346 002      3033X          ANI     FT.OR
```

```
054.260  076 011       3034X          MVI     A,EC.FNO        ASSUME FILE NOT OPEN FOR READ
054.262  067           3035X          STC
054.263  312 373 054   3036X          JZ      $REAB8          NOT OPEN FOR READ
054.266  170           3037X          MOV     A,B
054.267  261           3038X          ORA     C
054.270  312 373 054   3039X          JZ      $REAB8          ALL DONE
                       3040X
                       3041X *        COMPUTE MIN( DATA IN BUFFER, DATA REQUESTED)
                       3042X
054.273  052 223 056   3043X $REAB3   LHLD    T.PTR
054.276  353           3044X          XCHG                    (DE) = (FB.PTR) = ADDRESS OF DATA
054.277  052 225 056   3045X          LHLD    T.LIM           (HL) = LIMIT ADDRESS
054.302  175           3046X          MOV     A,L
054.303  223           3047X          SUB     E
054.304  157           3048X          MOV     L,A
054.305  174           3049X          MOV     A,H
054.306  232           3050X          SBB     D
054.307  147           3051X          MOV     H,A             (HL) = NUMBER OF BYTES IN BUFFER
054.310  171           3052X          MOV     A,C
054.311  225           3053X          SUB     L               COMPARE REQUESTED TO AVAILABLE
054.312  170           3054X          MOV     A,B
054.313  234           3055X          SBB     H
054.314  322 321 054   3056X          JNC     $REAB4          MORE REQUESTED THEN AVAILABLE
054.317  140           3057X          MOV     H,B
054.320  151           3058X          MOV     L,C             LIMIT TRANSFER TO REQUEST COUNT
054.321  174           3059X $REAB4   MOV     A,H
054.322  265           3060X          ORA     L
054.323  302 337 054   3061X          JNZ     $REAB6          SOME IN BUFFER
                       3062X
                       3063X *        BUFFER IS EMPTY, RE-FILL IT
                       3064X
054.326  315 135 056   3065X          CALL    $FFB            FILL FILE BUFFER
054.331  332 373 054   3066X          JC      $REAB8          ERROR CONDITION
054.334  303 273 054   3067X          JMP     $REAB3          COUNT NEW DATA
                       3068X
                       3069X *        GOT THE DATA, MOVE IT FROM BUFFER TO TARGET
                       3070X *
                       3071X *        (BC) = REQUESTED COUNT
                       3072X *        (DE) = FROM
                       3073X *        (HL) = COUNT
                       3074X *        ((SP)) = TO
                       3075X
054.337  171           3076X $REAB6   MOV     A,C
054.340  225           3077X          SUB     L
054.341  117           3078X          MOV     C,A
054.342  170           3079X          MOV     A,B
054.343  234           3080X          SBB     H
054.344  107           3081X          MOV     B,A             REMOVE BYTES ABOUT TO BE MOVED FROM REQUEST COUNT
054.345  305           3082X          PUSH    B
054.346  343           3083X          XTHL                    (HL) = REMAINING REQUEST COUNT
054.347  301           3084X          POP     B               (BC) = COUNT FOR THIS COPY
054.350  343           3085X          XTHL                    (HL) = TARGET ADDR, ((SP)) = REMAINING REQ. COUNT
054.351  032           3086X $REAB7   LDAX    D
054.352  167           3087X          MOV     M,A
054.353  023           3088X          INX     D
054.354  043           3089X          INX     H
```

```
054.355  013          3090X          DCX     B
054.356  170          3091X          MOV     A,B
054.357  261          3092X          ORA     C
054.360  302 351 054  3093X          JNZ     $REAB7          MORE TO GO
054.363  353          3094X          XCHG
054.364  042 223 056  3095X          SHLD    T.PTR           UPDATE POINTER
054.367  301          3096X          POP     B               (BC) = REMAINING COUNT
054.370  303 252 054  3097X          JMP     $REAB2          SEE IF MORE IN BUFFER
                      3098X
                      3099X *        READ COMPLETE.
                      3100X *
                      3101X *        (PSW) = COMPLETION FLAGS
                      3102X
054.373  321          3103X $REAB8   POP     D               RESTORE TARGET ADDRESS
054.374  341          3104X          POP     H
054.375  303 103 056  3105X          JMP     CTB             COPY TEMP POINTERS BACK TO BLOCK, EXIT
055.000               3106           XTEXT   FWRIB


                      3108X **       $FWRIB - WRITE BYTES FROM FILE BUFFER.
                      3109X *
                      3110X *        $FWRIB IS CALLED TO WRITE A NUMBER OF BYTES FROM A FILE BUFFER.
                      3111X *
                      3112X *        ENTRY   (BC) = BYTE COUNT
                      3113X *                (DE) = FWA FOR BYTES
                      3114X *                (HL) = ADDRESS OF FILE BUFFER
                      3115X *        EXIT    TO *FERROR* IF ERROR
                      3116X *                TO CALLER IF OK
                      3117X *                 (DE) = ADDRESS OF FIRST UNWRITTEN BYTE
                      3118X *        USES    A,F,B,C,D,E
                      3119X
                      3120X
055.000  315 007 055  3121X $FWRIB   CALL    $FWRIB.
055.003  320          3122X          RNC                     RETURN IF OK
055.004  303 373 055  3123X          JMP     $FERROR         ERROR
                      3124X
                      3125X
055.007               3126X $FWRIB.  EQU     *
055.007  345          3127X          PUSH    H
055.010  315 055 056  3128X          CALL    CBT             COPY BUFFER POINTERS TO TEMP CELLS
                      3129X
                      3130X *        COPY DATA FROM USER AREA TO BUFFER
                      3131X
055.013  325          3132X $WRIB2   PUSH    D               SAVE AREA ADDRESS
055.014  072 220 056  3133X          LDA     T.FLG
055.017  346 004      3134X          ANI     FT.OW           SEE IF OPEN FOR WRITE
055.021  312 155 055  3135X          JZ      $WRIB8          FILE NOT OPEN FOR WRITE
055.024  170          3136X          MOV     A,B
055.025  261          3137X          ORA     C
055.026  312 155 055  3138X          JZ      $WRIB8          ALL DONE
                      3139X
                      3140X *        COMPUTE MIN( ROOM IN BUFFER, WRITE COUNT REQUESTED)
                      3141X
055.031  052 223 056  3142X $WRIB3   LHLD    T.PTR
```

```
055.034  353            3143X          XCHG                    (DE) = (FB.PTR) = ADDRESS OF ROOM
055.035  052 227 056    3144X          LHLD    T.LWA           (HL) = LIMIT ADDRESS
055.040  175            3145X          MOV     A,L
055.041  223            3146X          SUB     E
055.042  157            3147X          MOV     L,A
055.043  174            3148X          MOV     A,H
055.044  232            3149X          SBB     D
055.045  147            3150X          MOV     H,A             (HL) = BYTES OF ROOM IN BUFFER
055.046  171            3151X          MOV     A,C             COMPARE REQUESTED COUNT TO BUFFER ROOM
055.047  225            3152X          SUB     L
055.050  170            3153X          MOV     A,B
055.051  234            3154X          SBB     H
055.052  322 057 055    3155X          JNC     $WRIB4          MORE REQUESTED THEN ROOM
055.055  140            3156X          MOV     H,B
055.056  151            3157X          MOV     L,C             USE REQUESTED COUNT
055.057  174            3158X $WRIB4   MOV     A,H
055.060  265            3159X          ORA     L
055.061  302 121 055    3160X          JNZ     $WRIB6          SOME ROOM IN BUFFER
                        3161X
                        3162X *        BUFFER IS FULL, EMPTY IT
                        3163X
055.064  305            3164X          PUSH    B               SAVE COUNT
055.065  052 221 056    3165X          LHLD    T.FWA
055.070  042 223 056    3166X          SHLD    T.PTR           CLEAR REMOVAL POINTER
055.073  353            3167X          XCHG
055.074  052 227 056    3168X          LHLD    T.LWA
055.077  175            3169X          MOV     A,L
055.100  223            3170X          SUB     E
055.101  117            3171X          MOV     C,A
055.102  174            3172X          MOV     A,H
055.103  232            3173X          SBB     D
055.104  107            3174X          MOV     B,A             (BC) = DATA IN BUFFER
055.105  072 217 056    3175X          LDA     T.CHA
055.110  377 005        3176X          DB      SYSCALL,,WRITE  WRITE BUFFER
055.112  301            3177X          POP     B               (BC) = DESIRED COUNT
055.113  322 031 055    3178X          JNC     $WRIB3          GOT THE DATA
                        3179X
                        3180X *        ERROR ON WRITE.
                        3181X
055.116  303 155 055    3182X          JMP     $WRIB8          HAVE ERROR
                        3183X
                        3184X *        GOT THE DATA, MOVE IT FROM BUFFER TO TARGET
                        3185X *
                        3186X *        (BC) = REQUEST COUNT
                        3187X *        (DE) = TO
                        3188X *        (HL) = COUNT
                        3189X *        ((SP)) = FROM
                        3190X
055.121  171            3191X $WRIB6   MOV     A,C
055.122  225            3192X          SUB     L
055.123  117            3193X          MOV     C,A
055.124  170            3194X          MOV     A,B
055.125  234            3195X          SBB     H
055.126  107            3196X          MOV     B,A             REMOVE BYTES ABOUT TO BE MOVED FROM REQUEST COUNT
055.127  305            3197X          PUSH    B
055.130  343            3198X          XTHL                    (HL) = REMAINING REQUEST COUNT
```

```
055.131  301          3199X          POP     B               (BC) = COUNT FOR THIS COPY
055.132  343          3200X          XTHL                    (HL) = TARGET ADDR, ((SP)) = REMAINING REQ. COUNT
055.133  176          3201X $WRIB7   MOV     A,M
055.134  022          3202X          STAX    D
055.135  023          3203X          INX     D
055.136  043          3204X          INX     H
055.137  013          3205X          DCX     B
055.140  170          3206X          MOV     A,B
055.141  261          3207X          ORA     C
055.142  302.133.055   3208X          JNZ     $WRIB7          MORE TO GO
055.145  353          3209X          XCHG
055.146  042.223.056   3210X          SHLD    T.PTR           UPDATE POINTER
055.151  301          3211X          POP     B               (BC) = REMAINING COUNT
055.152  303.013.055   3212X          JMP     $WRIB2          SEE IF MORE IN BUFFER
                       3213X
                       3214X *        WRITE COMPLETE.
                       3215X *
                       3216X *        (PSW) = COMPLETION FLAGS
                       3217X
055.155  321          3218X $WRIB8   POP     D               RESTORE TARGET ADDRESS
055.156  341          3219X          POP     H
055.157  303.103.056   3220X          JMP     CIB             COPY TEMP POINTERS BACK TO BLOCK, EXIT


                       3222X **       $FWBRK  -  BREAKOUTPUT                                  /80.02.GC/
                       3223X *
                       3224X *        $FWBRK empties the specified buffer by filling it with NULLs
                       3225X *        and then writing it.  Note this is used to insure that block
                       3226X *        mode I/O is output if it is not really a serial device (eg.
                       3227X *        writing to AI:  from  *EDIT*.
                       3228X *
                       3229X *
                       3230X *        ENTRY:  HL      =  FILE BLOCK POINTER
                       3231X *
                       3232X *        EXIT:   HL      =  FILE BLOCK POINTER
                       3233X *                TO  $FERROR IF ERROR
                       3234X *
                       3235X *        USES:   PSW,BC,DE
                       3236X *
                       3237X
055.162  315.171.055   3238X $FWBRK  CALL    $FWBRK.
055.165  320          3239X          RNC                     NO ERROR
                       3240X
055.166  303.373.055   3241X          JMP     $FERROR
                       3242X
055.171  345          3243X $FWBRK,  PUSH    H
055.172  315.055.056   3244X          CALL    CBT             COPY BUFFER TO TEMPORARY
055.175  315.205.055   3245X          CALL    $FWBRK1
055.200  341          3246X          POP     H
055.201  315.103.056   3247X          CALL    CIB             COPY TEMPORARY TO BUFFER
055.204  311          3248X          RET
                       3249X
055.205  052.227.056   3250X $FWBRK1 LHLD    T.LWA
055.210  353          3251X          XCHG                    DE = BUFFER LWA
```

```
055.211  052 223 056  3252X          LHLD    T.PTR           HL = BUFFER PTR
055.214  173          3253X          MOV     A,E
055.215  225          3254X          SUB     L
055.216  117          3255X          MOV     C,A
055.217  172          3256X          MOV     A,D
055.220  234          3257X          SBB     H
055.221  107          3258X          MOV     B,A             BC = DE - HL
055.222  261          3259X          ORA     C
055.223  310          3260X          RZ                      THE BUFFER IS ALREADY FLUSHED
                      3261X
                      3262X *        FILL THE BUFFER WITH NULLS
                      3263X
055.224  170          3264X FWBRK2   MOV     A,B
055.225  261          3265X          ORA     C
055.226  312 240 055  3266X          JZ      FWBRK3          NO MORE LEFT TO FILL
                      3267X
055.231  066 000      3268X          MVI     M,0
055.233  043          3269X          INX     H
055.234  013          3270X          DCX     B
055.235  303 224 055  3271X          JMP     FWBRK2
                      3272X
055.240  052 221 056  3273X FWBRK3   LHLD    T.FWA
055.243  042 223 056  3274X          SHLD    T.PTR
055.246  353          3275X          XCHG                    DE = BUFFER FWA
055.247  052 227 056  3276X          LHLD    T.LWA           HL = BUFFER LWA
055.252  175          3277X          MOV     A,L
055.253  223          3278X          SUB     E
055.254  117          3279X          MOV     C,A
055.255  174          3280X          MOV     A,H
055.256  232          3281X          SBB     D
055.257  107          3282X          MOV     B,A             BC = HL - DE  ( BC = COUNT )
055.260  072 217 056  3283X          LDA     T.CHA
055.263  377 005      3284X          DB      SYSCALL+.WRITE
055.265  311          3285X          RET
055.266               3286           XTEXT   FCLO


                      3288X **       $FCLO - CLOSE FILE BLOCK.
                      3289X *
                      3290X *        $FCLO IS CALLED TO TERMINATE PROCESSING THROUGH A FILE
                      3291X *        BLOCK.
                      3292X *
                      3293X *        ENTRY   (HL) = FILE BLOCK ADDRESS
                      3294X *        EXIT    TO $FERROR IF ERROR
                      3295X *                 TO CALLER IF OK
                      3296X *        USES    A,F,B,C,D,E
                      3297X
                      3298X
055.266  315 275 055  3299X $FCLO    CALL    $FCLO.
055.271  320          3300X          RNC                     NO ERROR
055.272  303 373 055  3301X          JMP     $FERROR
                      3302X
055.275  345          3303X $FCLO.   PUSH    H               SAVE FILE BLOCK ADDRESS
000.000               3304X          ERRNZ   FB.FLG-1
```

```
055.276  043            3305X          INX     H               (HL) = $FB.FLG
055.277  176            3306X          MOV     A,M
055.300  066 000        3307X          MVI     M,0             CLEAR FLAG
055.302  247            3308X          ANA     A
055.303  312 371 055    3309X          JZ      $FCLO4          FILE NOT OPEN
055.306  346 004        3310X          ANI     FT.OW
055.310  312 363 055    3311X          JZ      $FCLO3          NO WRITING, NO FLUSHING NEEDED
                        3312X
                        3313X *        WAS OPEN FOR WRITE, SEE IF NEED FLUSH THE LAST SECTOR
                        3314X
055.313  315 234 030    3315X          CALL    $INDL
055.316  003 000        3316X          DW      FB.PTR-FB.FLG
055.320  325            3317X          PUSH    D               SAVE (FB.PTR)
055.321  315 234 030    3318X          CALL    $INDL           (DE) = (FB.FWA)
055.324  001 000        3319X          DW      FB.FWA-FB.FLG
055.326  341            3320X          POP     H               (HL) = (FB.PTR)
055.327  175            3321X          MOV     A,L
055.330  223            3322X          SUB     E
055.331  117            3323X          MOV     C,A
055.332  174            3324X          MOV     A,H
055.333  232            3325X          SBB     D
055.334  107            3326X          MOV     B,A             (BC) = AMOUNT IN BLOCK
055.335  261            3327X          ORA     C
055.336  312 363 055    3328X          JZ      $FCLO3          NONE TO FLUSH
                        3329X
                        3330X *        NEED TO FLUSH BUFFER
                        3331X *
                        3332X *        (BC) = DATA AMOUNT
                        3333X *        (DE) = FWA
                        3334X *        (HL) = LWA+1
                        3335X
055.341  171            3336X          MOV     A,C
055.342  247            3337X          ANA     A
055.343  312 356 055    3338X          JZ      $FCLO2          DONT HAVE PARTIAL SECTOR
                        3339X
                        3340X *        ZERO FILL PARTIAL SECTOR
                        3341X
055.346  066 000        3342X $FCLO1   MVI     M,0
055.350  043            3343X          INX     H
055.351  014            3344X          INR     C
055.352  302 346 055    3345X          JNZ     $FCLO1
055.355  004            3346X          INR     B               COUNT ANOTHER FULL SECTOR
055.356  341            3347X $FCLO2   POP     H               (HL) = FB FWA
055.357  176            3348X          MOV     A,M             (A) = CHANNEL NUMBER
000.000                 3349X          ERRNZ   FB.CHA
055.360  345            3350X          PUSH    H
055.361  377 005        3351X          DB      SYSCALL+.WRITE          FLUSH
                        3352X
                        3353X *        READY TO CLOSE FILE.
                        3354X *
                        3355X *        'C' SET IF ERROR
                        3356X *        (A) = ERROR CODE
                        3357X
055.363  341            3358X $FCLO3   POP     H               (HL) = FILE BLOCK ADDRESS
055.364  330            3359X          RC                      ERROR
000.000                 3360X          ERRNZ   FB.CHA
```

```
055.365  176          3361X          MOV     A,M             (A) = CHANNEL NUMBER
055.366  345          3362X          PUSH    H
055.367  377 046      3363X          DB      SYSCALL,.CLOSE  CLOSE CHANNEL
055.371  341          3364X $FCLO4   POP     H               (HL) = FILE BLOCK ADDRESS
055.372  311          3365X          RET
055.373               3366           XTEXT   FERROR


                      3368X **       $FERROR - PROCESS FILE ERRORS.
                      3369X *
                      3370X *        $FERROR IS CALLED TO COMPLAIN ABOUT AN ERROR ENCOUNTERED
                      3371X *        WHEN PROCESSING FILES.
                      3372X *
                      3373X *        ENTRY   (A) = ERROR CODE
                      3374X *                (HL) = ADDRESS OF FILE NAME - FB.NAM
                      3375X *        EXIT    TO RESTART
                      3376X *        USES    ALL
                      3377X
                      3378X
055.373  365          3379X $FERROR  PUSH    PSW             SAVE CODE
055.374  315 136 031  3380X          CALL    $TYPTX
055.377  012 007 105  3381X          DB      NL,BELL,'ERROR ON FILE',' '+200Q
056.017  021 012 000  3382X          LXI     D,FB.NAM
056.022  031          3383X          DAD     D
                      3384X
                      3385X *        PRINT FILE NAME
                      3386X
056.023  176          3387X $FERR1   MOV     A,M
056.024  043          3388X          INX     H               ADVANCE MESSAGE
056.025  247          3389X          ANA     A
056.026  312 037 056  3390X          JZ      $FERR2
056.031  315 132 053  3391X          CALL    $WCHAR
056.034  303 023 056  3392X          JMP     $FERR1
                      3393X
                      3394X *        TYPE ERROR MESSAGE
                      3395X
056.037  315 136 031  3396X $FERR2   CALL    $TYPTX
056.042  040 055 240  3397X          DB      ' -',' '+200Q
056.045  046 012      3398X          MVI     H,NL
056.047  361          3399X          POP     PSW             (A) = CODE
056.050  377 057      3400X          DB      SYSCALL,.ERROR
056.052  303 211 045  3401X          JMP     RESTART         EXIT
056.055               3402           XTEXT   FUTIL


                      3404X **       $FUTIL - UTILITY ROUTINES FOR FILE BLOCK ROUTINES.
                      3405X
                      3406X **       CBT - COPY BLOCK POINTERS TO TEMP CELLS.
                      3407X *
                      3408X *        ENTRY   (HL) = FILE BLOK FWA
                      3409X *        EXIT    NONE
                      3410X *        USES    A,F,H,L
```

```
                                3411X
056.055  325                   3412X CBT   PUSH    D
056.056  305                   3413X       PUSH    B               SAVE REGISTERS
000.000                         3414X       ERRNZ   TLEN-10         ASSUME 10 BYTES TO MOVE
056.057  021 217 056            3415X       LXI     D,T.CHA         (DE) = TARGET FOR MOVE
056.062  006 005                3416X       MVI     B,10/2
056.064  176                   3417X CBT1  MOV     A,M             COPY FILE BUFFER INTO WORK AREA
056.065  022                   3418X       STAX    D
056.066  043                   3419X       INX     H
056.067  023                   3420X       INX     D
056.070  176                   3421X       MOV     A,M
056.071  022                   3422X       STAX    D
056.072  043                   3423X       INX     H
056.073  023                   3424X       INX     D
056.074  005                   3425X       DCR     B
056.075  302 064 056            3426X       JNZ     CBT1            MORE TO GO
056.100  301                   3427X       POP     B
056.101  321                   3428X       POP     D               (DE) = DATA TARGET ADDRESS
056.102  311                   3429X       RET
                                3430X
                                3431X
                                3432X **    CTB - COPY TEMP CELLS BACK TO FILE BLOCK.
                                3433X *
                                3434X *     ENTRY   (HL) = FILE BLOCK ADDRESS
                                3435X *     EXIT    NONE
                                3436X *     USES    NONE
                                3437X
056.103  365                   3438X CTB   PUSH    PSW
056.104  325                   3439X       PUSH    D
056.105  305                   3440X       PUSH    B
056.106  345                   3441X       PUSH    H               SAVE REGISTERS
056.107  006 004                3442X       MVI     B,8/2
056.111  021 217 056            3443X       LXI     D,T.CHA
056.114  032                   3444X CTB1  LDAX    D
056.115  167                   3445X       MOV     M,A
056.116  023                   3446X       INX     D
056.117  043                   3447X       INX     H
056.120  032                   3448X       LDAX    D
056.121  167                   3449X       MOV     M,A
056.122  023                   3450X       INX     D
056.123  043                   3451X       INX     H
056.124  005                   3452X       DCR     B
056.125  302 114 056            3453X       JNZ     CTB1            RESTORE FILE BUFFER VALUES
056.130  341                   3454X       POP     H
056.131  301                   3455X       POP     B
056.132  321                   3456X       POP     D
056.133  361                   3457X       POP     PSW
056.134  311                   3458X       RET
```

```
                        3460X **      $FFB - FILE FILE BUFFER.
                        3461X *
                        3462X *       $FFB FILLS THE FILE BUFFER BY READING FROM THE FILE.
                        3463X *
                        3464X *       ENTRY   NONE
                        3465X *       EXIT    'C' SET IF READ INCOMPLETE
                        3466X *                (A) = ERROR CODE
                        3467X *                'C' CLEAR IF READ COMPLETEE
                        3468X *                DATA IN BUFFER
                        3469X *       USES    A,F,D,E,H,L
                        3470X
                        3471X
056.135  072 231 056   3472X $FFB    LDA     EOFFLG
056.140  037           3473X          RAR
056.141  330           3474X          RC                      EOF
                        3475X
                        3476X *       CAN READ MORE, DO SO
                        3477X
056.142  305           3478X          PUSH    B               SAVE COUNT
056.143  052 221 056   3479X          LHLD    T.FWA
056.146  042 223 056   3480X          SHLD    T.PTR           CLEAR REMOVAL POINTER
056.151  353           3481X          XCHG
056.152  052 227 056   3482X          LHLD    T.LWA
056.155  042 225 056   3483X          SHLD    T.LIM           SET DATA LIMIT
056.160  175           3484X          MOV     A,L
056.161  223           3485X          SUB     E
056.162  117           3486X          MOV     C,A
056.163  174           3487X          MOV     A,H
056.164  232           3488X          SBB     D
056.165  107           3489X          MOV     B,A             (BC) = ROOM IN BUFFER
056.166  072 217 056   3490X          LDA     T.CHA
056.171  377 004       3491X          DB      SYSCALL,,READ   READ BUFFER
056.173  120           3492X          MOV     D,B             (D) = SECTORS UNREAD
056.174  301           3493X          POP     B               (BC) = DESIRED COUNT
056.175  320           3494X          RNC                     GOT THE DATA
                        3495X
                        3496X *       ERROR ON READ, SEE IF EOF
                        3497X
056.176  027           3498X          RAL
056.177  062 231 056   3499X          STA     EOFFLG          SET EOF, WE HOPE
056.202  376 003       3500X          CPI     EC.EOF*2+1
056.204  037           3501X          RAR
056.205  300           3502X          RNE                     IS NOT EOF, RETURN NOW!
056.206  072 226 056   3503X          LDA     T.LIM+1
056.211  222           3504X          SUB     D
056.212  062 226 056   3505X          STA     T.LIM+1         SET AMOUNT OF DATA WE DID GET
056.215  247           3506X          ANA     A
056.216  311           3507X          RET                     EXIT WITH DATA
                        3508X
                        3509X
                        3510X **      TEMP CELLS TO HOLD FILE BLOCK POINTERS DURING I/O
                        3511X
000.000                 3512X          ERRNZ   FB.CHA
056.217  000           3513X T.CHA    DB      0               CHANNEL NUMBER
000.000                 3514X          ERRNZ   *-T.CHA-FB.FLG
056.220  000           3515X T.FLG    DB      0               FLAG BYTE
```

```
000.000                 3516X           ERRNZ   *-T.CHA-FB.FWA
056.221  000 000        3517X T.FWA     DW      0
000.000                 3518X           ERRNZ   *-T.CHA-FB.PTR
056.223  000 000        3519X T.PTR     DW      0
000.000                 3520X           ERRNZ   *-T.CHA-FB.LIM
056.225  000 000        3521X T.LIM     DW      0
000.000                 3522X           ERRNZ   *-T.CHA-FB.LWA
056.227  000 000        3523X T.LWA     DW      0
000.012                 3524X TLEN      EQU     *-T.CHA         LENGTH OF TEMP CELLS
                        3525X
056.231  000            3526X EOFFLG    DB      0
```

```
                        3529  **      COMMAND TABLE.
                        3530  *
                        3531
056.232  000          3532  CMDTAB  DB      0               DUMY FIRST ENTRY
                        3533
                        3534  *       0 - [OPT]ADDR
056.233  221 240 040  3535          DB      091H,0A0H,' ',0
                        3536
                        3537  *       1 - [OPT]ADDR-ADDR
056.237  221 241 055  3538          DB      091H,0A1H,'-',0A5H,' ',0
                        3539
                        3540  *       2 - [OPT]ADDR=VAL
056.245  221 240 075  3541          DB      091H,0A0H,'=',0
                        3542
                        3543  *       3 - [OPT]ADDR-ADDR=VAL
056.251  221 241 055  3544          DB      091H,0A1H,'-',0A5H,'=',0
                        3545
                        3546  *       4 - [OPT]CTL-R
056.257  223 022 040  3547          DB      093H,'R'-'@',' ',0
                        3548
                        3549  *       5 - [OPT]REGX
056.263  223 222 224  3550          DB      093H,092H,094H,' ',0
                        3551
                        3552  *       6 - [OPT]REGX=
056.270  223 222 224  3553          DB      093H,092H,094H,'=',0
                        3554
                        3555  *       7 - EXEC A1-A2,...,AN
056.275  105 130 105  3556          DB      'EXEC ',0A1H,'-',0D0H,0
                        3557
                        3558  *       8 - STEP ADDR
056.306  123 124 105  3559          DB      'STEP ',0A0H,NL,0
                        3560
                        3561  *       9 - BKPT A1,...,AN
056.316  225 320 000  3562          DB      095H,0D0H,0
                        3563
                        3564  *       10 - BKPT DSPLY
056.321  225 104 123  3565          DB      095H,'DSPLY ',0
                        3566
                        3567  *       11 - CLEAR A1,,,,,AN
056.331  226 320 000  3568          DB      096H,0D0H,00
                        3569
                        3570  **      12 - CLEAR ALL
056.334  226 101 114  3571          DB      096H,'ALL',NL,0
                        3572
                        3573  *       13 - DUMP
056.342  104 125 115  3574          DB      'DUMP ',0B0H,081H,' ,',081H,0A1H,'-',0A5H,NL,0
                        3575
                        3576  *       14 - LOAD
056.361  114 117 101  3577          DB      'LOAD ',0B0H,NL,0
                        3578
                        3579  *       15 - LOAD PIC
056.371  114 117 101  3580          DB      'LOAD PIC ',0B0H,081H,' ,',081H,0A3H,NL,0
                        3581
                        3582  *       16 - GO
057.012  107 117 040  3583          DB      'GO ',0A1H,NL,0
                        3584
```

```
057.020  000          3585          DB      0               END OF MAIN STRINGS.

                       3587  **      EXTENSION STRINGS.
                       3588
057.020                3589  CMDEXS  EQU     *-1             START TABLE WITH 00
                       3590
                       3591  *       1 - [OPT]
057.021  202 106 202  3592          DB      082H,'F',082H,080H,'DA',080H,0C0H,0
                       3593
                       3594  *       2 - REG
057.032  122 105 107  3595          DB      'REG',0C0H,0
                       3596
                       3597  *       3 - [OPT]
057.037  200 104 101  3598          DB      080H,'DA',080H,0C0H,00
                       3599
                       3600  *       4 - REGISTER ID
057.045  205 101 102  3601          DB      085H,'ABCDEHLSPFM',085H,0C0H,0
                       3602
                       3603  *       5 - BKPT^
057.064  102 113 120  3604          DB      'BKPT ',0C0H,0
                       3605
                       3606  *       6 - CLEAR^
057.073  103 114 105  3607          DB      'CLEAR ',0C0H,0
```

```
                                3610  **      MEMORY TOP AND BOTTOM VALUES
                                3611
057.103  000 000               3612  TOPVAL  DW      0
057.105  000 000               3613  BOTVAL  DW      0
                                3614
057.107  000                   3615  CSLMD   DB      0               SAVED VALUE OF USER S.CSLMD
057.110  000                   3616  CONFL   DB      0               SAVED VALUE OF USER S.CONFL
                                3617
057.111                         3618  DARA    EQU     *               REGISTER TABLE
057.111  101 003               3619          DB      'A',3
057.113  102 005               3620          DB      'B',5
057.115  103 004               3621          DB      'C',4
057.117  104 007               3622          DB      'D',7
057.121  105 006               3623          DB      'E',6
057.123  110 011               3624          DB      'H',9
057.125  114 010               3625          DB      'L',8
057.127  106 002               3626          DB      'F',2
057.131  120 212               3627  DARAP   DB      'P',10+80H
057.133  115 210               3628          DB      'M',08+80H
057.135  123 200               3629          DB      'S',00+80H
057.137  000                   3630          DB      0
000.013                         3631  DARAL   EQU     *-DARA-1/2



                                3633  **      BKPTAB - BREAKPOINT TABLE.
                                3634  *
                                3635  *       BKPTAB CONTAINS INFORMATION ABOUT BREAKPOINTS.
                                3636  *
                                3637  *       BYTE 0 - LOW ORDER ADDRESS
                                3638  *            1 - HIGH ORDER ADDRESS
                                3639  *            2 - BREAKPOINT REPEAT COUNT
                                3640  *            3 - INSTRUCTION AT BREAKPOINT
                                3641  *
                                3642  *       WHEN IN THE DEBUGGER PACKAGE, THE BREAKPOINT ARE NOT
                                3643  *       SET.
                                3644
                                3645
000.010                         3646  BKPTBL  EQU     8
                                3647
057.140  000 000 000 000 000   3648  BKPTAB  DW      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
                                3649
                                3650  *       EXTRA ENTRY TO AUTOMATICALLY SET AND CLEAR BKPFLG
                                3651
057.200  206 057 001           3652          DW      BKPFLG,1,0
                                3653
057.206  000                   3654  BKPFLG  DB      0               NON-ZERO IF BREAKPOINTS ARE SET
                                3655
057.207  000                   3656  $LSTIN  DB      0               LAST READ BYTE
```

```
                        3658  **      LOAD/DUMP FILE BUFFER.
                        3659  *
                        3660
057.210  005           3661  MEMFB   DB      CN.LD           CHANNEL NUMBER
057.211  000           3662          DB      0               FLAGS
057.212  353 057 353   3663          DW      MEMBUF,MEMBUF,MEMBUFE,MEMBUFE
                        3664
057.222                 3665  CMD.BA  DS      FB.NAML         SPACE FOR FILE NAME
                        3666
                        3667
057.243                 3668  BFILHDR DS      ABS.COD         ROOM FOR BINARY AND PIC HEADERS FOR LOAD/DUMP
000.002                 3669          ERRMI   ABS.COD-PIC.COD       MUST HAVE ROOM FOR EITHER
                        3670
057.253                 3671  PATCH   DS      64              PATCH AREA
```

```
                                3675  **      PRS     -  PRESET CODE
                                3676  *
                                3677  *       THIS CODE IS ONLY USED AT ENTRY, IT IS THEN OVERLAID BY BUFFERS
                                3678  *
                                3679
        057.353                 3680  PRS     EQU     *
                                3681
                                3682  *       CHECK THE VERSION OF HDOS
                                3683
        057.353  377 011        3684          DB      SYSCALL,.VERS
        057.355  332 370 057    3685          JC      PRSERR1         NO .VERS SYSTEM CALL
        057.360  376 040        3686          CPI     VERS
        057.362  302 370 057    3687          JNZ     PRSERR1
                                3688
                                3689  *       GO TO THE REAL ENTRY
                                3690
        057.365  303 101 045    3691          JMP     HBUG
                                3692
        057.370  076 050        3693  PRSERR1 MVI     A,EC.NCV
                                3694
        057.372  046 012        3695  PRSERR  MVI     H,NL
        057.374  377 057        3696          DB      SYSCALL,.ERROR
                                3697
        057.376  303 041 046    3698          JMP     EXIT1
                                3699
        060.001                 3700  MEML    EQU     *               END OF LOAD IMAGE
                                3701
        057.353                 3702          ORG     PRS             OVERLAY PRS CODE
                                3703
        057.353                 3704  MEMBUF  DS      256             BUFFER
        060.353                 3705  MEMBUFE EQU     *               END OF BUFFER
                                3706
                                3707
        060.353                 3708  RMEML   EQU     *               RUNNING MEMORY LIMIT
                                3709
        060.353                 3710          END
ASSEMBLY COMPLETE
 3710 STATEMENTS
    1 ERRORS DETECTED
10812 BYTES FREE
```

$CRLF	053135	1170	1314	1949	2577L	2638							
$DADA	030072	799	936	2031	2592E								
$DADA.	030101	2602E	2764										
$FCLO	055266	1670	1795	3299L									
$FCLO.	055275	3299	3303L										
$FCL01	055346	3342L	3345										
$FCL02	055356	3338	3347L										
$FCL03	055363	3311	3328	3358L									
$FCL04	055371	3309	3364L										
$FERR1	056023	3387L	3392										
$FERR2	056037	3390	3396L										
$FERROR	055373	1981	2910	2914	2918	3018	3123	3241	3301	3379L			
$FFB	054135	3065	3472L										
$FOPE1	054147	2940	2952L										
$FOPE2	054202	2976	2980L										
$FOPEA	054210	2980	2984E										
$FOPER	054220	2974	2994L										
$FOPER	054067	1686	1738	2908L									
$FOPER.	054114	2908	2921L										
$FOPEU	054105	2916L											
$FOPEU.	054122	2916	2925L										
$FOPEW	054076	1650	2912L										
$FOPEW.	054117	2912	2923L										
$FREAB	054227	1689	1717	1741	1760	1775	3015L						
$FREAB.	054242	3015	3023E										
$FWBRK	055162	3238L											
$FWBRK.	055171	3238	3243L										
$FWBRK1	055205	3245	3250L										
$FWRIB	055000	1642	1669	3121L									
$FWRIB.	055007	3121	3126E										
$HLIHL	030211	1655	2533E										
$INCHA	053143	564	1890	2620L	2646	2655							
$INCHAA	053276	2626	2630	2659	2663	2669L							
$INDL	030234	2522E	3315	3318									
$LSTIN	057207	1079	1874	3656L									
$MCU	053113	565	1259	2500L									
$MOVE	030252	2456E											
$MU86	031007	931	2684E										
$RCHAR	053124	1076	1258	2560L	2561	2620							
$REAB2	054252	3031L	3097										
$REAB3	054273	3043L	3067										
$REAB4	054321	3056	3059L										
$REAB6	054337	3061	3076L										
$REAB7	054351	3086L	3093										
$REAB8	054373	3036	3039	3066	3103L								
$RSTALL	031047	2473E											
$SAVALL	031054	1139	2487E										
$TBLS	053277	2045	2138	2705L	2978								
$TBRA	031076	673	2156	2831E									
$TDD	053332	2188	2760L										
$TDD.	053330	2759L											
$TJMP	031061	1201	2744E										
$TJMP.	031062	2746E											
$TOD	054032	1954	1956	2174	2178	2843L							
$TPA	054013	2193	2197	2806L									
$TYPC.	054064	985	992	1083	1087	1893	2064	2078	2775	2809	2811	2853	2878L
$TYPCH	054060	613	1074	1425	1844	2065	2868L						
$TYPTX	031136	1140	1191	1228	1256	1573	1579	1642	1722	1800	1957	1988	2354

```
2548E   2651    3380    3396
 $TYPTX. 031144      2550E
 $WCHAR  053132      2564L     2628    2664    2666    3391
 $WRIB2  055013      3132L     3212
 $WRIB3  055031      3142L     3178
 $WRIB4  055057      3155      3158L
 $WRIB6  055121      3160      3191L
 $WRIB7  055133      3201L     3208
 $WRIB8  055155      3135      3138    3182    3218L
 $ZERO   031212       589      1493    2892E
 .       000040      2405S     2406
 .ABUSS  040024       236E
 .ALARM  002136       209E
 .ALEDS  040013       234E
 .BKP.   047045      1504      1535E
 .CHFLG  000060       109L
 .CLEAN  000205       124L
 .CLEAR  000055       106L     1174    1179
 .CLEARA 000056       107L
 .CLOSE  000046        99L     3363
 .CLRCO  000007        83L
 .CONSL  000006        82L
 .CRC    002347       217E
 .CRCSUM 040027       237E
 .CTC    002172       211E
 .CTL2FL 040066       243E
 .CTLC   000041        94L     1145
 .CTLFLG 040011       233E     1587    1590    2288
 .DAD    000206       125L
 .DECODE 000053       104L
 .DELET  000050       101L
 .DISMT  000061       110L
 .DLEDS  040021       235E
 .DLY    000053       206E
 .DMNMS  000203       122L
 .DMOUN  000201       120L
 .DOD    003122       220E
 .DODA   003356       222E
 .DSPMOD 040007       231E
 .DSPROT 040006       230E
 .DUMP   001374       208E
 .ERROR  000057       108L     3400    3696
 .EXIT   000000        76L     1263    2977    2983
 .HORN   002140       210E
 .IDENT  000000       205E
 .IOWRK  040002       228E
 .LINK   000040        93L
 .LOAD   001267       207E
 .LOADD  000062       111L
 .LOADO  000010        84L
 .MFLAG  040010       232E
 .MONMS  000202       121L
 .MOUNT  000200       119L
 .NAME   000054       105L
 .NMIRET 040064       242E
 .OPEN   000063       112L
 .OPENC  000045        98L
 .OPENR  000042        95L     2994
```

```
.OPENU  000044   97L   2996
.OPENW  000043   96L   2995
.PCHL   002264  213E
.POSIT  000047  100L
.PRINT  000003   79L
.RCK    003260  221E
.READ   000004   80L   3491
.REGI   040005  229E
.REGPTR 040035  240E
.RENAM  000051  102L
.RESET  000204  123L
.RNB    002331  216E
.RNP    002325  215E
.SCIN   000001   77L   2560
.SCOUT  000002   78L   2564  2578  2878
.SETTP  000052  103L   1979
.SRS    002265  214E
.START  040000  227E
.SYSRES 000012   86L
.TICCNT 040033  239E
.TPERR  002205  212E
.TPERRX 040031  238E
.UIVEC  040037  241E   1505  1507
.VERS   000011   85L   3684
.WNB    003024  219E
.WNP    003017  218E
.WRITE  000005   81L   3176  3284  3351
ABS.COD 000010  464L    540  1659  1687  3668  3669
ABS.ENT 000006  462L   1656  1698
ABS.ID  000000  458L
ABS.LDA 000002  460L   1619  1628  1666  1711
ABS.LEN 000004  461L   1637  1663  1708
ACN     043353  917L   1009  1048  1857
ACN1    043361  921L    942
ACN2    044027  924    926   948L
ACN3    044051  955    966L
AEC     044061  628    980L
AEC1    044110  984    993L
ANV     050371 1304   1341  1834E
ANV1    050376 1840L  1879
ANV2    051031 1849   1856L
ANV3    051047 1866L  1885
ANV4    051054 1866   1873L
ANV4$5  051063 1877L  1899
ANV5    051072 1859   1883L
ANV6    051077 1852   1890L
BELL    000007   41E   1257  1643  1723  1989  2355  3381
BFILHDR 057243 1619   1628  1637  1656  1658  1660  1663  1666  1688  1691  1698  1708
                1711   1740  1743  1752  3668L
BKP1    047155 1558   1585L
BKP2    047160 1371   1586L
BKPFLG  057206 2253   2366  3652  3654L
BKPTAB  057140 1410   1491  2088  2251  2364  3648L
BKPTBL  000010 1411   1492  2090  2252  2365  3646E
BKSP    000010   43E  2621  2652  2652
BOTVAL  057105 1924   2240  3613L
C.STX   000002   45E
C.SYN   000026   44E
```

```
CB.CLI  000100  151E  174  1589
CB.MTL  000040  150E
CB.SPK  000200  152E
CB.SSI  000020  149E  1589
CB2.CLI 000002  155E
CB2.ORG 000040  158E
CB2.SID 000100  157E
CB2.SSI 000001  154E
CBL1    046316  1456L  1482
CBL2    046333  1471L
CBL3    046341  1467  1480L
CBT     056055  3027  3128  3244  3412L
CBT1    056064  3417L  3426
CCP     042200  555E  1196
CEA     051123  1907L  2209  2218  2423
CEA1    051142  1909  1924L
CEA2    051146  1911  1929L
CLL     051152  1940E  2128
CLR     051216  1716  1759  1975L
CMD.8   043075  677  727E
CMD.81  043101  732L  737
CMD.82  043121  735  748L
CMD.83  043135  760L  763
CMD.8A  045022  587  588  751  754  1113L  1340  1417  1429  1576  1797  1846  1858
                2043  2069  2134  2146
CMD.9   043146  678  772L
CMD.91  043161  778L  780
CMD.9A  045026  774  862  1115L
CMD.A   043173  679  796L
CMD.AA  045030  798  1125L  1200  1368  1615  1618  1623  1626  1757  1766  2211
CMD.B   043212  680  811L
CMD.B1  043224  817L  824
CMD.B2  043243  818  828L
CMD.B5  043250  811  817  839L
CMD.BA  057222  814  822  3665L
CMD.C   043276  681  862L
CMD.D   043305  682  873L
CMD.D1  043310  874L  893
CMD.D2  043336  891  894L
CMD.DA  045040  873  1127L  1452  2316
CMD.DA2 045100  887  1128L
CMD.DB  043352  875  901L
CMD.NG  043053  655  698L  1016  1027  1049  1051
CMD.OK  043045  665  690L  744  764  801  829  864  896
CMD.OK. 043046  691L  781
CMD.RA  043063  709L  899  1068
CMD.TL  045101  588  1130L
CMD1    042205  559L
CMD2    042213  564L  615  630  633  647
CMD3    042226  571L  612
CMD4    042251  587L  641  646  701
CMD5    042325  603  619L  693
CMD6    042340  628L  713
CMD7    042376  621  627  651L
CMD8    043027  660  669L
CMDADR  044310  583  1108L
CMDEXS  057020  775  3589E
CMDTAB  056232  582  3532L
```

```
CN.170M 000014  192E
CN.174M 000003  191E
CN.ABQ  000200  196E
CN.BAU  000100  195E
CN.LD   000005  30E   1173  1178  3661
CN.MEM  000040  194E
CN.PRI  000020  193E
CND.H17 000000  198E
CND.H47 000001  200E
CND.NDI 000000  199E
CO.FLG  000001  322E
CONFL   057110  1809  2302  2408  3614L
CR      000015  37E
CS.FLG  000200  323E
CSL.CHR 000001  279E  2403
CSL.ECH 000200  296E  2403
CSL.RAW 000004  297E
CSL.WRP 000002  298E
CSLMD   057107  1808  2300  2402  3615L
CSR1    046135  1337  1339L
CTB     056103  3105  3220  3247  3438L
CTB1    056114  3444L 3453
CTLA    000001  52E
CTLB    000002  53E
CTLC    000003  54E
CTLD    000004  55E   566   1077  1891
CTLO    000017  56E
CTLP    000020  57E
CTLQ    000021  58E
CTLS    000023  59E
CTLZ    000032  60E
CTP.2SB 000010  308E
CTP.BKM 000002  309E
CTP.BKS 000200  304E
CTP.FF  000100  305E
CTP.MLI 000040  306E
CTP.MLO 000020  307E
CTP.TAB 000001  310E
CUB     051305  1283  1877  2005L
D.CON   040110  258L
D.RAM   040240  261L
D.VEC   040130  260L
DARA    057111  1312  2044  3618E 3631
DARAL   000013  1313  3631E
DARAP   057131  2026  3627L
DAS     044112  800   876   1004L
DAS1    044151  1011  1026L
DAS2    044155  1023  1028L
DBL1    046241  1415L 1441
DBL2    046304  1422  1437L
DF.CLR  000376  485E
DF.EMP  000377  484E
DIR.ALD 000025  500L
DIR.CLU 000015  493L
DIR.CRD 000023  499L
DIR.EXT 000010  488L
DIR.FGN 000020  496L
DIR.FLG 000016  494L
```

```
DIR.LGN 000021  497L
DIR.LSI 000022  498L
DIR.NAM 000000  487L
DIR.PRO 000013  489L
DIR.VER 000014  490L
DIRELEN 000027  502E    534
DIRIDL  000015  491E
DM.MR   000000  164E
DM.MW   000001  165E
DM.RR   000002  166E
DM.RW   000003  167E
DMP0    047233  1617    1623L
DMP1    047271  1636    1642L   2224
DMP2    047312  1625    1638    1648L
DMPA    050000  1648    1672L
DRA     051322  1334    2027L   2067
DRA.    051317  1577    1654    1700    2026L   2426
DRI     051337  1325    1333    2043L
DRV     051353  1315    2062E
DRV.    051357  1328    2065L
DVB     051373  1282    1431    2076L
DVP2    046051  1282L   1285
EC.CNA  000004  340L
EC.DDA  000027  359L
EC.DIF  000017  351L
EC.DIW  000035  365L
EC.DNI  000045  373L
EC.DNR  000046  374L
EC.DNS  000005  341L
EC.DSC  000047  375L
EC.EOF  000001  337L    3017    3500
EC.EOM  000002  338L
EC.FAO  000031  361L    2947
EC.FAP  000026  358L
EC.FL   000030  360L
EC.FNF  000014  348L
EC.FNO  000011  345L    3034
EC.FNR  000034  364L
EC.FOD  000043  371L
EC.FUC  000013  347L
EC.ICN  000016  350L
EC.IDN  000006  342L
EC.IFC  000020  352L
EC.IFN  000007  343L
EC.ILC  000003  339L
EC.ILO  000040  368L
EC.ILR  000012  346L
EC.ILV  000037  367L
EC.IOI  000052  378L
EC.IS   000032  362L
EC.NCV  000050  376L    3693
EC.NEM  000021  353L
EC.NOS  000051  377L
EC.NPM  000044  372L
EC.NRD  000010  344L
EC.NVM  000042  370L
EC.OTL  000053  379L
EC.RF   000022  354L
```

EC.UNA	000036	366L								
EC.UND	000015	349L								
EC.UUN	000033	363L								
EC.VPM	000041	369L								
EC.WF	000023	355L								
EC.WP	000025	357L								
EC.WPV	000024	356L								
ENL	000212	50E	1142	1580	1643	1989	2355			
EOFFLG	056231	3025	3472	3499	3526L					
ESC	000033	48E	1895							
EXIT	046001	567	1078	1256L	1892					
EXIT1	046041	1262L	3698							
FB.CHA	000000	385L	2931	2934	3349	3360	3512			
FB.FLG	000001	386L	1181	2934	2951	2989	3304	3316	3319	3514
FB.FWA	000002	387L	2951	2957	3319	3516				
FB.LIM	000006	389L	2962	2967	3520					
FB.LWA	000010	390L	3522							
FB.NAM	000012	391L	392	2967	3382					
FB.NAML	000021	392E	822	3665						
FB.PTR	000004	388L	2957	2962	3316	3518				
FBENL	000033	393E								
FBT	052003	1466	1552	2088L	2330	2334				
FBT1	052011	2092L	2108							
FBT2	052025	2094	2103L							
FBT3	052036	2099	2110L							
FF	000014	51E								
FIC	044217	922	1063E							
FIC1	044231	1074L	1085	1088						
FIC2	044235	1065	1076L							
FIC2.5	044245	1079L								
FICA	044217	558	1064E	1836						
FNRA	044312	1110E								
FT.ABS	000000	401E	541	1657	1694					
FT.BAC	000003	404E								
FT.DD	000001	513E								
FT.OC	000020	517E								
FT.OR	000002	514E	2921	2925	2994	2996	3033			
FT.OU	000010	516E								
FT.OW	000004	515E	2923	2925	2995	2996	3134	3310		
FT.PIC	000001	402E	1746							
FT.REL	000002	403E								
FVD	052040	1424	1578	1802	1842	2076	2127E			
FVD.A	052156	2160	2192L							
FVD.D	052140	2159	2182L							
FVD.D1	052153	2184	2188L							
FVD.Q	052126	2158	2173L							
FVD0	052111	2152	2156L							
FVD0.1	052066	2137	2142L							
FVD0.2	052067	2140	2144L							
FVD1	052117	2144	2163L							
FVDA	052121	2135	2167L							
FWBRK2	055224	3264L	3271							
FWBRK3	055240	3266	3273L							
GO	046377	1508L								
GO0	046361	1503L	1600							
GO2	046367	1505L	1592							
GO3	047201	1585	1596L							
GOA	047033	1514	1524E							

```
GOB      047037   1520     1527E
GOC      047043   1522     1530E
HBUG     045101   1134E    3691
HBUG1    045171   1155L
HBUGA    045270   1203L    1207
I.CONFL  000004    325E     326
I.CONTY  000001    312E     313
I.CONWI  000003    318E     319
I.CSLMD  000000    301E
I.CUSOR  000002    315E     316
INC0     053211   2622     2624    2643L
INC1     053240   2639     2653L   2667
INC3     053250   2650     2659L
INC4     053267   2661     2665L
INTRPT   045332   1144     1228L
IOC.CGN  000010    522L
IOC.CSI  000011    523L
IOC.DDA  000002    510L     518     532
IOC.DES  000016    529L
IOC.DEV  000020    530L
IOC.DIL  000021    532E
IOC.DIR  000023    534L
IOC.DRL  000010    526E
IOC.DTA  000014    528L
IOC.FLG  000004    512L     526
IOC.GRT  000005    520L
IOC.LGN  000012    524L
IOC.LNK  000000    509L
IOC.LSI  000013    525L
IOC.SPG  000007    521L
IOC.SQL  000003    518E
IOC.UNI  000022    531L
IOCCTD   000001    538E
IOCELEN  000052    536E
IP.CON   000362    140E
IP.PAD   000360    136E
ISSUEA   045142   1142L
LF       000012     38E     990
LINE     044312    559      600    1109L   1110    1774    1778    2637    2644
LINPTR   045020    560      571     981     988    1111L   2643    2653
LOA.1    050250   1771L    1790
LOA.2    050314   1718     1781    1794L
LOA2     050107   1690     1693    1697    1722L   1742    1745    1748    1761    1777
LOAA     050142   1684     1726L
LOAB     050363   1736     1813L
M.FOX    000303    184E
M.PAM8   000021    183E
MEMBUF   057353   3663     3663    3704L
MEMBUFE  060353   3663     3663    3705E
MEMFB    057210   1181     1649    1661    1668    1685    1737    1771    1794    1980    1986    3661L
MEML     060001    543     3700E
MI.BKP   000327     26E     2383
MI.JMP   000303     24E     1506    1835
MI.LDA   000072     25E      557
NL       000012     49E       50     629     895     989     990    1080    1141    1141    1142    1257    2577
                   3381     3398    3559    3571    3574    3577    3580    3583    3695
NUL2     000000     40E
NULL     000200     39E
```

```
NXTCHA  044306  580   610   634   1106L
OP.CTL  000360  137E  1591  2289
OP.DIG  000360  138E
OP.SEG  000361  139E
OP2.CTL 000362  141E
PATCH   057253  3671L
PATCNT  044307  590   1107L 1127  1950
PIC.COD 000006  476L  1739  1753  1767  3669
PIC.ID  000000  471L
PIC.LEN 000002  473L
PIC.PTR 000004  474L  1752
PRS     057353  544   3680E 3702
PRSERR  057372  3695L
PRSERR1 057370  3685  3687  3693L
QUOTE   000047  46E
RAS     052170  1281  1303  2207E
RAS1    052223  2214  2229L
RAS2    052233  2225  2237L
RBM     052246  1187  1542  2251L
RBM1    052261  2258L 2272
REGPTR  045226  1164  1185E 1248  1382  1508  1541  1598  2030
RESTART 045211  1172E 1990  3401
REX     047123  1249  1388  1553  1572E
RFD     052301  1383  1599  2286L
RFDA    052302  1588  2287E
RMEML   060353  1984  3708E
ROMBOOT 030000  253E
RUBOUT  000177  42E   2623
RUC     052310  1515  2300L
S.CAADR 040333  329L
S.CCTAB 040335  330L
S.CONFL 040332  327L  2303  2404  2406
S.CONTY 040327  314L  2648
S.CONWI 040331  320L  1945
S.CSLMD 040326  302L  313   316   319   326   2301  2400  2405
S.CUSOR 040330  317L  1188  1941
S.DAIC  040310  283L
S.DATE  040277  282L
S.GRT0  024000  249E
S.GRT1  025000  250E
S.GRT2  026000  251E
S.HIMEM 040316  285L
S.INT   040343  263L
S.OMAX  040324  291L
S.SOVR  041146  265L  267
S.SYSM  040320  287L
S.TIME  040312  284L
S.USRM  040322  289L
S.VAL   040277  262L  280
SBL     052326  1352  1398  2315E
SBL1    052331  2320L 2350
SBL2    052361  2331  2340L
SBL3    052376  2335  2354L
SBM     053015  1503  2364L
SBM1    053030  2371L 2386
SBM2    053043  2384L
SC.UART 000372  416E
SDC     053052  1146  1236  1379  1538  2400L
```

```
SRC      044275   596     778     1098L   1101
SSA      053074  1367    1502     2420L
STACK    042200   269E    1138
STACKL   001032   267E
START    045101  1136E
STKPTR   044304   654     698      709    1105L
STP1     046163  1369L   1387
STPA     046174  1369    1373L    1384
STPRTN   046175  1370    1377L
SYDD     040130   259E
SYSCALL  000377    69E    1145     1174    1179    1263    1979    2560    2564    2578    2878    2983    3176
                  3284    3351     3363    3400    3491    3684    3696
T.CHA    056217  3175    3283     3415    3443    3490    3513L   3514    3516    3518    3520    3522    3524
T.FLG    056220  3032    3133     3515L
T.FWA    056221  3165    3273     3479    3517L
T.LIM    056225  3045    3483     3503    3505    3521L
T.LWA    056227  3144    3168     3250    3276    3482    3523L
T.PTR    056223  3043    3095     3142    3166    3210    3252    3274    3480    3519L
TAB      000011    47E
TB.CAB   046346  1216    1491L
TB.CBL   046314  1215    1451E
TB.CMP   046063  1206    1301E
TB.CMS   046063  1205    1294E
TB.CSR   046121  1210    1332E
TB.DAR   046072  1208    1312L
TB.DARI  046102  1315L   1318
TB.DARI  000004  1207E   1951
TB.DBL   046234  1214    1409E
TB.DMP   047216  1217    1611E
TB.DSR   046113  1209    1325L
TB.DVP   046045  1204    1279E
TB.DVS   046045  1203    1272E
TB.EXE   046145  1211    1350E
TB.GU    046356  1220    1354    1501E
TB.LOA   050006  1218    1683E
TB.LOA.  050150  1219    1735E
TB.SBL   046231  1213    1397E
TB.STP   046155  1212    1366E
TBGX     045225  1183E   1198    1233    1261    1581    1644    1724    2356
TBL1     053306  2709L   2715
TBL2     053324  2707    2719L
TBL3     053326  2712    2723L
TDD1     053333  2761L   2784
TDD2     053352  2771L   2773
TDDA     054001  2762    2788E
TLEN     000012  3414    3524E
TOD1     054036  2847L   2856
TOFVAL   057103  1929    2237    3512L
TPA1     054025  2807    2810L
UCI.ER   000020   438E
UCI.IE   000002   440E
UCI.IR   000100   436E
UCI.RE   000004   439E
UCI.RO   000040   437E
UCI.TE   000001   441E
UDR      000000   413E
UMI.16X  000002   431E
UMI.IB   000100   421E
```

```
UMI.1X  000001  430E
UMI.2B  000300  423E
UMI.64X 000003  432E
UMI.HB  000200  422E
UMI.L5  000000  426E
UMI.L6  000004  427E
UMI.L7  000010  428E
UMI.L8  000014  429E
UMI.PA  000020  425E
UMI.PE  000040  424E
UO.CLK  000001  176E
UO.DDU  000002  175E
UO.HLT  000200  173E
UO.NFR  000100  174E
USERFWA 042200  270E  540  542  543  1153
USERMD  045340  1231E 1235 1378 1511 1537
USR     000001  414E
USR.BD  000100  445E
USR.FE  000040  446E
USR.OE  000020  447E
USR.PE  000010  448E
USR.RXR 000002  450E
USR.TXE 000004  449E
USR.TXR 000001  451E
VERS    000040  67E  3686

19644 BYTES FREE
```