```
                        1
                        4  ***     H8ASM - H8 ASSEMBLER.
                        5  *
                        6  *          J. G. L.,  09/76, FOR *WINTEK* CORP.
                        7  *
                        8  *        COPYRIGHT  09/76, *WINTEK* CORPORATION,
                        9  *                                LAFAYETTE, IND.


                       11  ***     H8ASM - H8 RESIDENT ASSEMBLER.
                       12  *
                       13  *       SOFTWARE ISSUE NUMBER:
                       14  *
                       15  *       0104.02.00.                                        /78.10.GC/
                       16  *                            79/12   --.05.00
                       17  *                            80/03   --.06.00              /80.03.GC/
                       18  *         W. Z.,             80/06   UPDATES FOR XREF, NOREF, & XIEXT


                       20  ****    ASSEMBLY CONSTANTS.
                       21
000.002                22  VER     EQU     2               VERSION NUMBER
000.000                23  LEV     EQU     0               VERSION LEVEL
000.101                24  MOD     EQU     'A'             MODIFICATION LEVEL
                       25
                       26  **      ERROR FLAGS
                       27
000.001                28  ERR.U   EQU     001Q            *U* - UNDEFINED
000.002                29  ERR.R   EQU     002Q            *R* - ILLEGAL REGISTER SPECIFICATION
000.004                30  ERR.D   EQU     004Q            *D* - DOUBLY DEFINED SYMBOL
000.010                31  ERR.A   EQU     010Q            *A* - EXPRESSION ERROR
000.020                32  ERR.V   EQU     020Q            *V* - VALUE TOO LARGE FOR FIELD
000.040                33  ERR.F   EQU     040Q            *F* - ILLEGAL STATEMENT FORMAT
000.100                34  ERR.O   EQU     100Q            *O* - OPCODE ERROR
000.200                35  ERR.P   EQU     200Q            *P* - PROGRAMMER FLAGGED ERROR
                       36
                       37  **      LIST OPTIONS.
                       38
000.001                39  LST.L   EQU     001Q            MASTER LIST FLAG
000.002                40  LST.I   EQU     002Q            LIST IF-SKIPPED LINES
000.004                41  LST.C   EQU     004Q            LIST INCLUDED CODE
000.010                42  LST.R   EQU     010Q            List Cross-References       /80.03.sc/
000.200                43  LST.G   EQU     200Q            LIST ALL GENERATED BYTES
                       44
000.000                45          XTEXT   STDEF                                       /80.03.GC/
```

```
                        47X **      SYMBOL DEFINITION TYPES.                                    /80.03.GC/
                        48X
000.000                 49X ST.UND  EQU     0           UNDEFINED
000.001                 50X ST.LAB  EQU     1           LABEL
000.002                 51X ST.EQU  EQU     2           DEFINED VIA *EQU*
000.003                 52X ST.SET  EQU     3           DEFINED VIA *SET*
000.007                 53X ST.MSK  EQU     00000111B   Mask for definition classes
                        54X
000.010                 55X ST.NRF  EQU     00001000B   No Cross References are to be taken
000.020                 56X ST.DNA  EQU     00010000B   Definition of symbol not allowed
                        57X
000.100                 58X ST.REL  EQU     01000000B   Relocatable
000.200                 59X ST.DBL  EQU     10000000B   Doubly defined
000.000                 60          XTEXT   XTDEF                                               /80.03.GC/


                        62X **      XREF HISTORY REFERENCE-TYPE FLAGS                           /80.03.GC/
                        63X
000.000                 64X XT.REF  EQU     0           REFERENCED IN EXPRESSION
000.001                 65X XT.LAB  EQU     1           DEFINED AS  LABEL
000.002                 66X XT.EQU  EQU     2           DEFINED VIA *EQU* PSEUDO
000.003                 67X XT.SET  EQU     3           DEFINED VIA *SET* PSEUDO
000.004                 68X XT.NRF  EQU     4           REFERENCED VIA *NOREF* PSEUDO           /WCZ062680/


                        70  **      CHARACTER TYPES
                        71
000.200                 72  CT.ALPH EQU     10000000B   ALPHA CHARACTER
                        73
                        74
                        75  **      CHANNEL NUMBERS
                        76
000.000                 77  CN.BIN  EQU     0           BINARY FILE
000.001                 78  CN.LST  EQU     1           LISTING FILE
000.002                 79  CN.SOU  EQU     2           SOURCE INPUT FILE
000.003                 80  CN.XTX  EQU     3           XTEXT
000.004                 81  CN.TMP  EQU     4           TEMP FILE                               /80.03.GC/
                        82
                        83
                        84  **      MACHINE INSTRUCTIONS
                        85
000.303                 86  MI.JMP  EQU     303Q        JMP
000.072                 87  MI.LDA  EQU     072Q        LDA
000.311                 88  MI.RET  EQU     311Q        RET
000.043                 89  MI.INXH EQU     43Q         INX H INSTRUCTIIN
000.325                 90  MI.PSHD EQU     325Q        PUSH D
                        91
000.000                 92          XTEXT   ASCII
```

```
                        94X **      ASCII CHARACTER EQUIVALENCES.
                        95X
000.015                 96X CR      EQU     13              CARRIAGE RETURN
000.012                 97X LF      EQU     10              LINE FEED
000.200                 98X NULL    EQU     200Q            PAD CHARACTER
000.000                 99X NUL2    EQU     0
000.007                100X BELL    EQU     7               BELL CHARACTER
000.177                101X RUBOUT  EQU     177Q
000.010                102X BKSP    EQU     10Q             CTL-H
000.026                103X C.SYN   EQU     26Q             SYNC
000.002                104X C.STX   EQU     2               STX
000.047                105X QUOTE   EQU     47Q
000.011                106X TAB     EQU     11Q
000.033                107X ESC     EQU     33Q
000.012                108X NL      EQU     12Q             NEW LINE (HDOS SYSTEMS)
000.212                109X ENL     EQU     NL+200Q         NL + END-OF-LINE-FLAG
000.014                110X FF      EQU     14Q             FORM FEED
000.001                111X CTLA    EQU     01Q             CTL-A
000.002                112X CTLB    EQU     02Q             CTL-B
000.003                113X CTLC    EQU     03Q             CTL-C
000.004                114X CTLD    EQU     04Q             CTL-D
000.017                115X CTLO    EQU     17Q             CTL-O
000.020                116X CTLP    EQU     20Q             CTL-P
000.021                117X CTLQ    EQU     21Q             CTL-Q
000.023                118X CTLS    EQU     23Q             CTL-S
000.032                119X CTLZ    EQU     32Q             CTL-Z
000.000                120          XTEXT   DIRDEF


                       122X **      DIRECTORY ENTRY FORMAT.
                       123X
000.000                124X         ORG     0
                       125X
                       126X
000.377                127X DF.EMP  EQU     377Q            FLAGS ENTRY EMPTY
000.376                128X DF.CLR  EQU     376Q            FLAGS ENTRY EMPTY, REST OF DIR ALSO CLEAR
                       129X
000.000                130X DIR.NAM DS      8               NAME
000.010                131X DIR.EXT DS      3               EXTENSION
000.013                132X DIR.PRO DS      1               PROJECT
000.014                133X DIR.VER DS      1               VERSION
000.015                134X DIRIDL  EQU     *               FILE IDENTIFICATION LENGTH
                       135X
000.015                136X DIR.CLU DS      1               CLUSTER FACTOR
000.016                137X DIR.FLG DS      1               FLAGS
000.017                138X         DS      1               RESERVED
000.020                139X DIR.FGN DS      1               FIRST GROUP NUMBER
000.021                140X DIR.LGN DS      1               LAST GROUP NUMBER
000.022                141X DIR.LSI DS      1               LAST SECTOR INDEX (IN LAST GROUP)
000.023                142X DIR.CRD DS      2               CREATION DATE
000.025                143X DIR.ALD DS      2               LAST ALTERATION DATE
                       144X
000.027                145X DIRELEN EQU     *               DIRECTORY ENTRY LENGTH
000.027                146          XTEXT   DEVDEF
```

```
                        148X **      DEVICE TABLE ENTRYS.
                        149X
000.000                 150X         ORG     0
                        151X
000.000                 152X DEV.NAM DS      2          DEVICE NAME
000.000                 153X DV.EL   EQU     00000000B  END OF DEVICE LIST FLAG
000.001                 154X DV.NU   EQU     00000001B  DEVICE ENTRY NOT IN USE
                        155X
000.002                 156X DEV.RES DS      1          DRIVER RESIDENSE CODE
000.001                 157X DR.IM   EQU     00000001B  DRIVER IN MEMORY
000.002                 158X DR.PR   EQU     00000010B  DRIVER PERMINANTLY RESIDENT
                        159X
000.003                 160X DEV.JMP DS      1          JMP TO PROCESSOR
000.004                 161X DEV.DDA DS      2          DRIVER ADDRESS
000.006                 162X DEV.FLG DS      1          FLAG BYTE
000.001                 163X DT.DD   EQU     00000001B  DIRECTORY DEVICE
000.002                 164X DT.CR   EQU     00000010B  CAPABLE OF READ OPERATION
000.004                 165X DT.CW   EQU     00000100B  CAPABLE OF WRITE OPERATION
000.010                 166X DT.RN   EQU     00001000B  Capable of random access        /80.02.sc/
000.020                 167X DT.CH   EQU     00010000B  Capable of Character mode       /80.02.sc/
                        168X
000.007                 169X DEV.MUM DS      1          MOUNTED UNIT MASK
000.010                 170X DEV.MNU DS      1          MAXIMUM NUMBER OF UNITS
000.011                 171X DEV.UNT DS      2          ADDRESS OF UNIT SPECIFIC DATA TABLE
                        172X
000.013                 173X DEV.DVL DS      2          DRIVER BYTE LENGTH
000.015                 174X DEV.DVG DS      1          DRIVER ROUTINE GROUP ADDRESS
                        175X
000.016                 176X DEVELEN EQU     *          DEVICE TABLE ENTRY LENGTH


                        178X **      UNIT SPECIFIC DEVICE DATA TABLE ENTRIES
                        179X
000.000                 180X         ORG     0
                        181X
000.000                 182X UNT.FLG DS      1          UNIT SPECIFIC  *DEV.FLG*
000.001                 183X UNT.SPG DS      1          Sectors Per Group               /80.04.SC/
000.002                 184X UNT.GRT DS      2          ADDRESS OF GROUP RESERVATION TABLE (IF DT.DD)
000.004                 185X UNT.GTS DS      2          GRT SECTOR NUMBER
000.006                 186X UNT.DIS DS      2          DIRECTORY FIRST SECTOR NUMBER
                        187X
000.010                 188X UNT.SIZ EQU     *          SIZE OF UNIT SPECIFIC DATA TABLE PER UNIT
000.010                 189          XTEXT   HUSDEF


                        191X **      HUSDEF - DEFINE HUS PARAMETER.
                        192X *
                        193X
                        194X
000.040                 195X VERS    EQU     2*16+0     VERSION 2.0
                        196X
000.377                 197X SYSCALL EQU     377Q       SYSCALL INSTRUCTION
                        198X
```

```
                        199X
000.000                 200X            ORG     0
                        201X
                        202X *          RESIDENT FUNCTIONS
                        203X
000.000                 204X .EXIT      DS      1               EXIT (MUST BE FIRST)
000.001                 205X .SCIN      DS      1               SCIN
000.002                 206X .SCOUT     DS      1               SCOUT
000.003                 207X .PRINT     DS      1               PRINT
000.004                 208X .READ      DS      1               READ
000.005                 209X .WRITE     DS      1               WRITE
000.006                 210X .CONSL     DS      1               SET/CLEAR CONSOLE OPTIONS
000.007                 211X .CLRCO     DS      1               CLEAR CONSOLE BUFFER
000.010                 212X .LOADO     DS      1               LOAD AN OVERLAY
000.011                 213X .VERS      DS      1               RETURN HDOS VERSION NUMBER
000.012                 214X .SYSRES    DS      1               PRECEDING FUNCTIONS ARE RESIDENT
                        215X
                        216X
                        217X *          *HDOSOVL0.SYS*  FUNCTIONS
                        218X
000.040                 219X            ORG     40A
                        220X
000.040                 221X .LINK      DS      1               LINK  (MUST BE FIRST)
000.041                 222X .CTLC      DS      1               CTL-C
000.042                 223X .OPENR     DS      1               OPENR
000.043                 224X .OPENW     DS      1               OPENW
000.044                 225X .OPENU     DS      1               OPENU
000.045                 226X .OPENC     DS      1               OPENC
000.046                 227X .CLOSE     DS      1               CLOSE
000.047                 228X .POSIT     DS      1               POSITION
000.050                 229X .DELET     DS      1               DELETE
000.051                 230X .RENAM     DS      1               RENAME
000.052                 231X .SETTP     DS      1               SETTOP
000.053                 232X .DECODE    DS      1               NAME DECODE
000.054                 233X .NAME      DS      1               GET FILE NAME FROM CHANNEL
000.055                 234X .CLEAR     DS      1               CLEAR CHAN
000.056                 235X .CLEARA    DS      1               CLEAR ALL CHANS
000.057                 236X .ERROR     DS      1               LOOKUP ERROR
000.060                 237X .CHFLG     DS      1               CHANGE FLAGS
000.061                 238X .DISMT     DS      1               FLAG SYSTEM DISK DISMOUNTED
000.062                 239X .LOADD     DS      1               LOAD DEVICE DRIVER
000.063                 240X .OPEN      DS      1               Parametrized Open
                        241X
                        242X
                        243X *          *HDOSOVL1.SYS*  FUNCTIONS
                        244X
000.200                 245X            ORG     200Q
                        246X
000.200                 247X .MOUNT     DS      1               MOUNT  (MUST BE FIRST)
000.201                 248X .DMOUN     DS      1               DISMOUNT
000.202                 249X .MONMS     DS      1               MOUNT/NO MESSAGE
000.203                 250X .DMNMS     DS      1               DISMOUNT/NO MESSAGE
000.204                 251X .RESET     DS      1               RESET = DISMOUNT/MOUNT OF UNIT
000.205                 252X .CLEAN     DS      1               Clean device
000.206                 253X .DAD       DS      1               Dismount All Disks              /80.08.sc/
000.207                 254             XTEXT   HOSERV
```

```
                    256X **     HDOS SYSTEM EQUIVALENCES.
                    257X *
                    258X
024.000             259X S.GRT0  EQU    24000A          SYSTEM AREA FOR  GRT0
025.000             260X S.GRT1  EQU    25000A          SYSTEM AREA FOR  GRT1
026.000             261X S.GRT2  EQU    26000A          SYSTEM AREA FOR  GRT2
                    262X
030.000             263X ROMBOOT EQU    30000A          ROM BOOT ENTRY
                    264X
040.100             265X         ORG    40100A          FREE SPACE FROM PAM-8
                    266X
040.100             267X         DS     8               JUMP TO SYSTEM EXIT
040.110             268X D.CON   DS     16              DISK CONSTANTS
040.130             269X SYDD    EQU    *               SYSTEM DISK ENTRY POINT
040.130             270X D.VEC   DS     24*3            SYSTEM ROM ENTRY VECTORS
040.240             271X D.RAM   DS     31              SYSTEM ROM WORK AREA
040.277             272X S.VAL   DS     36              SYSTEM VALUES
040.343             273X S.INT   DS     115             SYSTEM INTERNAL WORK AREAS
041.126             274X         DS     16
041.146             275X S.SOVR  DS     2               STACK OVERFLOW WARNING
041.150             276X         DS     42200A-*        SYSTEM STACK
001.032             277X STACKL  EQU    *-S.SOVR        STACK SIZE
                    278X
042.200             279X STACK   EQU    *               LWA+1 SYSTEM STACK
042.200             280X USERFWA EQU    *               USER FWA
042.200             281          XTEXT  ESVAL


                    283X **     S.VAL - SYSTEM VALUE DEFINTIONS.
                    284X *
                    285X *      THESE VALUES ARE SET AND MAINTAINED BY THE SYSTEM.
                    286X *
                    287X *      THE DECK HOSEQU MUST BE MODIFIED WHEN THIS IS MODIFIED.
                    288X
                    289X
040.277             290X         ORG    S.VAL
                    291X
040.277             292X S.DATE  DS     9               SYSTEM DATE (IN ASCII)
040.310             293X S.DATC  DS     2               CODED DATE
040.312             294X S.TIME  DS     4               TIME FROM MIDNIGHT (IN TICS)
040.316             295X S.HIMEM DS     2               HARDWARE HIGH MEMORY ADRESS+1
                    296X
040.320             297X S.SYSM  DS     2               FWA RESIDENT SYSTEM
                    298X
040.322             299X S.USRM  DS     2               LWA USER MEMORY
                    300X
040.324             301X S.OMAX  DS     2               MAX OVERLAY SIZE FOR SYSTEM
                    302X
                    303X
                    304X **     THE FOLLOWING FIVE CELLS SHOULD BE MODIFIED/READ ONLY VIA THE .CONSL SYSCALL
                    305X
000.200             306X CSL.ECH EQU    10000000B       SUPPRESS ECHO
000.004             307X CSL.RAW EQU    00000100B       Raw Mode I/O                    /80.09.sc/
000.002             308X CSL.WRP EQU    00000010B       WRAP LINES AT WIDTH
```

```
000.001                 309X CSL.CHR EQU     00000001B      OPERATE IN CHARACTER MODE
                        310X
000.000                 311X I.CSLMD EQU     0              S.CSLMD IS FIRST BYTE
040.326                 312X S.CSLMD DS      1              CONSOLE MODE
                        313X
000.200                 314X CTP.BKS EQU     10000000B      TERMINAL PROCESSES BACKSPACES
000.100                 315X CTP.FF  EQU     01000000B      Terminal Processes Form-Feed    /80.09.sc/
000.040                 316X CTP.MLI EQU     00100000B      MAP LOWER CASE TO UPPER ON INPUT
000.020                 317X CTP.MLO EQU     00010000B      MAP LOWER CASE TO UPPER ON OUTPUT
000.010                 318X CTP.2SB EQU     00001000B      TERMINAL NEEDS TWO STOP BITS
000.002                 319X CTP.BKM EQU     00000010B      MAP BKSP (UPON INPUT) TO RUBOUT
000.001                 320X CTP.TAB EQU     00000001B      TERMINAL SUPPORTS TAB CHARACTERS
                        321X
000.001                 322X I.CONTY EQU     1              S.CONTY IS 2ND BYTE
000.000                 323X         ERRNZ   *-S.CSLMD-I.CONTY
040.327                 324X S.CONTY DS      1              CONSOLE TYPE FLAGS
000.002                 325X I.CUSOR EQU     2              S.CUSOR IS 3RD BYTE
000.000                 326X         ERRNZ   *-S.CSLMD-I.CUSOR
040.330                 327X S.CUSOR DS      1              CURRENT CURSOR POSITION
000.003                 328X I.CONWI EQU     3              S.CONWI IS 4TH BYTE
000.000                 329X         ERRNZ   *-S.CSLMD-I.CONWI
040.331                 330X S.CONWI DS      1              CONSOLE WIDTH
                        331X
000.001                 332X CO.FLG  EQU     00000001B      CTL-O FLAG
000.200                 333X CS.FLG  EQU     10000000B      CTL-S FLAG
                        334X
000.004                 335X I.CONFL EQU     4              S.CONFL IS 5TH BYTE
000.000                 336X         ERRNZ   *-S.CSLMD-I.CONFL
040.332                 337X S.CONFL DS      1              CONSOLE FLAGS
                        338X
040.333                 339X S.CAADR DS      2              ADDRESS FOR ABORT PROCESSING (>256 IF VALID)
040.335                 340X S.CCTAB DS      6              ADDR FOR CTL-A, CTL-B, CTL-C PROCESSING
040.343                 341          XTEXT   ESINT


                        343X **      S.INT - SYSTEM INTERNAL WORKAREA DEFINITIONS.
                        344X *
                        345X *       THESE CELLS ARE REFERENCED BY OVERLAYS AND MAIN CODE, AND
                        346X *       MUST THEREFORE RESIDE IN FIXED LOW MEMORY.
                        347X
                        348X
040.343                 349X         ORG     S.INT
                        350X
                        351X **      CONSOLE STATUS FLAGS
                        352X
040.343                 353X S.CDB   DS      1              CONSOLE DESCRIPTOR BYTE
000.000                 354X CDB.H85 EQU     00000000B
000.001                 355X CDB.H84 EQU     00000001B      =0 IF H8-5, =1 IF H8-4
040.344                 356X S.BAUD  DS      2              [0-14]  H8-4 BAUD RATE, =0 IF H8-5
                        357X *                              [15]    =1 IF BAUD RATE => 2 STOP BITS
                        358X
                        359X **      TABLE ADDRESS WORDS
                        360X
040.346                 361X S.DLINK DS      2              ADDRESS OF DATA IN HDOS CODE
```

```
040.350                362X S.OFWA  DS      2               FWA  OVERLAY  TABLE
040.352                363X S.CFWA  DS      2               FWA  CHANNEL  TABLE
040.354                364X S.DFWA  DS      2               FWA  DEVICE   TABLE
040.356                365X S.RFWA  DS      2               FWA  RESIDENT HDOS CODE
                       366X
                       367X **      DEVICE DRIVER DELAYED LOAD FLAGS
                       368X
040.360                369X S.DDLDA DS      2               DRIVER LOAD ADDRESS (HIGH BYTE=0 IF NO LOAD PENDING)
040.362                370X S.DDLEN DS      2               CODE LENGTH IN BYTES
040.364                371X S.DDGRP DS      1               GROUP NUMBER FOR DRIVER
040.365                372X         DS      1               HOLD PLACE
                       373X *S.DDSEC        DS      2               SECTOR NUMBER FOR DRIVER ( * OBSOLETE ! * )
040.366                374X S.DDDTA DS      2               DEVICE'S ADDRESS IN DEVLST +DEV.RES
040.370                375X S.DDOPC DS      1               OPEN OPCODE PENDEDING
                       376X
                       377X **      OVERLAY MANAGEMENT FLAGS
                       378X
000.001                379X OVL.IN  EQU     00000001B       IN MEMORY
000.002                380X OVL.RES EQU     00000010B       PERMINANTLY RESIDENT
000.014                381X OVL.NUM EQU     00001100B       OVERLAY NUMBER MASK
000.200                382X OVL.UCS EQU     10000000B       USER CODE SWAPPED FOR OVERLAY
                       383X
040.371                384X S.OVLFL DS      1               OVERLAY FLAG
040.372                385X S.UCSF  DS      2               FWA SWAPPED USER CODE
040.374                386X S.UCSL  DS      2               LENGTH SWAPPED USER CODE
040.376                387X S.OVLS  DS      2               SIZE OF OVERLAY CODE
041.000                388X S.OVLE  DS      2               ENTRY POINT OF OVERLAY CODE
                       389X
041.002                390X S.SSN   DS      2               SWAP AREA SECTOR NUMBER
041.004                391X S.OSN   DS      2               OVERLAY SECTOR NUMBER
                       392X
                       393X *       SYSCALL PROCESSING WORK AREAS
                       394X
041.006                395X S.CACC  DS      1               (ACC) UPON SYSCALL
041.007                396X S.CODE  DS      1               SYSCALL INDEX IN PROGRESS
                       397X
                       398X *       JUMPS TO ROUTINES IN RESIDENT HDOS CODE
                       399X
041.010                400X S.JUMPS DS      0               START OF JUMP VECTORS
041.010                401X S.SDD   DS      3               JUMP TO STAND-IN DEVICE DRIVER
041.013                402X S.FASER DS      3               JUMP TO FATSERR (FATAL SYSTEM ERROR)
041.016                403X S.DIREA DS      3               JUMP TO DIREAD (DISK FILE READ)
041.021                404X S.FCI   DS      3               JUMP TO FCI (FETCH CHANNEL INFO)
041.024                405X S.SCI   DS      3               JUMP TO SCI (STORE CHANNEL INFO)
041.027                406X S.GUP   DS      3               JUMP TO GUP (GET UNIT POINTER)
                       407X
041.032                408X S.MOUNT DS      1               <>0 IF THE SYSTEM DISK IS MOUNTED
041.033                409X S.DCS   DS      1               DEFAULT CLUSTER SIZE-1
                       410X
041.034                411X S.BOOTF DS      1               BOOT FLAGS
000.001                412X BOOT.P  EQU     00000001B       EXECUTE PROLOGUE UPON BOOTUP
                       413X
                       414X *       STACK VALUE SAVED FOR OVERLAY SYSCALLS
                       415X
041.035                416X S.OVSTK DS      2               VALUE OF SP UPON SYSCALLS USING OVERLAY
                       417X
```

```
041.037                 418X            DS      1               RESERVED


                        420X **         ACTIVE I/O AREA.
                        421X *
                        422X *          THE AIO.XXX AREA CONTAINS INFORMATION ABOUT THE I/O OPERATION
                        423X *          CURRENTLY BEING PERFORMED. THE INFORMATION IS OBTAINED FROM
                        424X *          THE CHANNEL TABLE, AND WILL BE RESTORED THERE WHEN DONE.
                        425X *
                        426X *          NORMALLY, THE AIO.XXX INFORMATION WOULD BE OBTAINED DIRECTLY
                        427X *          FROM VARIOUS SYSTEM TABLES VIA POINTER REGISTERS. SINCE THE
                        428X *          8080 HAS NO GOOD INDEXED ADDRESSING, THE DATA IS MANUALLY
                        429X *          COPIED INTO THE AIO.XXX CELLS BEFORE PROCESSING, AND
                        430X *          BACKDATED AFTER PROCESSING.
                        431X
041.040                 432X AIO.VEC DS         3               JUMP INSTRUCTION
041.041                 433X AIO.DDA EQU        *-2             DEVICE DRIVER ADDRESS
041.043                 434X AIO.FLG DS         1               FLAG BYTE
041.044                 435X AIO.GRT DS         2               ADDRESS OF GROUP RESERV TABLE
041.046                 436X AIO.SPG DS         1               SECTORS PER GROUP
041.047                 437X AIO.CGN DS         1               CURRENT GROUP NUMBER
041.050                 438X AIO.CSI DS         1               CURRENT SECTOR INDEX
041.051                 439X AIO.LGN DS         1               LAST GROUP NUMBER
041.052                 440X AIO.LSI DS         1               LAST SECTOR INDEX
041.053                 441X AIO.DTA DS         2               DEVICE TABLE ADDRESS
041.055                 442X AIO.DES DS         2               DIRECTORY SECTOR
041.057                 443X AIO.DEV DS         2               DEVICE CODE
041.061                 444X AIO.UNI DS         1               UNIT NUMBER (0-9)
                        445X
041.062                 446X AIO.DIR DS         DIRELEN         DIRECTORY ENTRY
                        447X
041.111                 448X AIO.CNT DS         1               SECTOR COUNT
041.112                 449X AIO.EOM DS         1               END OF MEDIA FLAG
041.113                 450X AIO.EOF DS         1               END OF FILE FLAG
041.114                 451X AIO.TFP DS         2               TEMP FILE POINTERS
041.116                 452X AIO.CHA DS         2               ADDRESS OF CHANNEL BLOCK (IOC.DDA)


041.120                 454X S.BDA   DS         1               Boot Device Address (Setup by ROM) /80.09.sc/
041.121                 455X S.SCR   DS         2               SYSTEM SCRATCH AREA ADDRESS
041.123                 456          XTEXT      FILDEF


                        458X **         FILDEF - FILE TYPE DEFINITIONS.
                        459X *
                        460X *          DB      3770,FT.XXX
                        461X
                        462X
000.000                 463X FT.ABS  EQU        0               ABSOLUTE BINARY
000.001                 464X FT.PIC  EQU        1               POSITION INDEPENDANT CODE
```

```
000.002                 465X FT.REL  EQU     2               RELOCATABLE CODE
000.003                 466X FT.BAC  EQU     3               COMPILED BASIC CODE
041.123                 467          XTEXT   ABSDEF


                        469X **      ABS FORMAT EQUIVALENCES.
                        470X
000.000                 471X         ORG     0
                        472X
000.000                 473X ABS.ID  DS      1               377Q = BINARY FILE FLAG
000.001                 474X         DS      1               FILE TYPE (FT.ABS)
000.002                 475X ABS.LDA DS      2               LOAD ADDRESS
000.004                 476X ABS.LEN DS      2               LENGTH OF ENTIRE RECORD
000.006                 477X ABS.ENT DS      2               ENTRY POINT
                        478X
000.010                 479X ABS.COD DS      0               CODE STARTS HERE
000.010                 480          XTEXT   PICDEF


                        482X **      PIC FORMAT EQUIVALENCES.
                        483X
000.000                 484X         ORG     0
                        485X
000.000                 486X PIC.ID  DS      1               377Q = BINARY FILE FLAG
000.001                 487X         DS      1               FILE TYPE (FT.PIC)
000.002                 488X PIC.LEN DS      2               LENGTH OF ENTIRE RECORD
000.004                 489X PIC.PTR DS      2               INDEX OF START OF PIC TABLE
                        490X
000.006                 491X PIC.COD DS      0               CODE STARTS HERE
000.006                 492          XTEXT   FBDEF


                        494X **      FILE BLOCK DEFINITIONS.
                        495X
000.000                 496X         ORG     0
000.000                 497X FB.CHA  DS      1               CHANNEL NUMBER
000.001                 498X FB.FLG  DS      1               FLAGS
000.002                 499X FB.FWA  DS      2               BUFFER FWA
000.004                 500X FB.PTR  DS      2               BUFFER POINTER
000.006                 501X FB.LIM  DS      2               LIMIT OF DATA IN BUFFER (READ OPERATIONS)
000.010                 502X FB.LWA  DS      2               LWA OF BUFFER
000.012                 503X FB.NAM  DS      4+8+4+1         NAME OF FILE
000.021                 504X FB.NAML EQU     *-FB.NAM
000.033                 505X FBENL   EQU     *               ENTRY LENGTH
000.033                 506          XTEXT   ECDEF
```

```
                        508X **     ERROR CODE DEFINITIONS.
                        509X
000.000                 510X        ORG     0
000.000                 511X        DS      1           NO ERROR #0
000.001                 512X EC.EOF DS      1           END OF FILE
000.002                 513X EC.EOM DS      1           END OF MEDIA
000.003                 514X EC.ILC DS      1           ILLEGAL SYSCALL CODE
000.004                 515X EC.CNA DS      1           CHANNEL NOT AVAILABLE
000.005                 516X EC.DNS DS      1           DEVICE NOT SUITABLE
000.006                 517X EC.IDN DS      1           ILLEGAL DEVICE NAME
000.007                 518X EC.IFN DS      1           ILLEGAL FILE NAME
000.010                 519X EC.NRD DS      1           NO ROOM FOR DEVICE DRIVER
000.011                 520X EC.FNO DS      1           CHANNEL NOT OPEN
000.012                 521X EC.ILR DS      1           ILLEGAL REQUEST
000.013                 522X EC.FUC DS      1           FILE USAGE CONFLICT
000.014                 523X EC.FNF DS      1           FILE NAME NOT FOUND
000.015                 524X EC.UND DS      1           UNKNOWN DEVICE
000.016                 525X EC.ICN DS      1           ILLEGAL CHANNEL NUMBER
000.017                 526X EC.DIF DS      1           DIRECTORY FULL
000.020                 527X EC.IFC DS      1           ILLEGAL FILE CONTENTS
000.021                 528X EC.NEM DS      1           NOT ENOUGH MEMORY *
000.022                 529X EC.RF  DS      1           READ FAILURE
000.023                 530X EC.WF  DS      1           WRITE FAILURE
000.024                 531X EC.WPV DS      1           WRITE PROTECTION VIOLATION
000.025                 532X EC.WP  DS      1           DISK WRITE PROTECTED
000.026                 533X EC.FAP DS      1           FILE ALREADY PRESENT
000.027                 534X EC.DDA DS      1           DEVICE DRIVER ABORT
000.030                 535X EC.FL  DS      1           FILE LOCKED
000.031                 536X EC.FAO DS      1           FILE ALREADY OPEN
000.032                 537X EC.IS  DS      1           ILLEGAL SWITCH
000.033                 538X EC.UUN DS      1           UNKNOWN UNIT NUMBER
000.034                 539X EC.FNR DS      1           FILE NAME REQUIRED
000.035                 540X EC.DIW DS      1           DEVICE IS NOT WRITABLE (OR WRITE LOCKED)
000.036                 541X EC.UNA DS      1           UNIT NOT AVAILABLE
000.037                 542X EC.ILV DS      1           ILLEGAL VALUE
000.040                 543X EC.ILO DS      1           ILLEGAL OPTION
000.041                 544X EC.VPM DS      1           VOLUME PRESENTLY MOUNTED ON DEVICE
000.042                 545X EC.NVM DS      1           NO VOLUME PRESENTLY MOUNTED
000.043                 546X EC.FOD DS      1           FILE OPEN ON DEVICE
000.044                 547X EC.NPM DS      1           NO PROVISIONS MADE FOR REMOUNTING MORE DISKS
000.045                 548X EC.DNI DS      1           DISK NOT INITIALIZED
000.046                 549X EC.DNR DS      1           DISK IS NOT READABLE
000.047                 550X EC.DSC DS      1           DISK STRUCTURE IS CORRUPT
000.050                 551X EC.NCV DS      1           NOT CORRECT VERSION OF HDOS
000.051                 552X EC.NOS DS      1           NO OPERATING SYSTEM MOUNTED
000.052                 553X EC.IOI DS      1           ILLEGAL OVERLAY INDEX
000.053                 554X EC.OTL DS      1           OVERLAY TO LARGE
000.054                 555         XTEXT   IOCDEF
```

```
                        557X **      I/O CHANNEL DEFINITIONS.
                        558X
        000.000         559X         ORG     0
                        560X
        000.000         561X IOC.LNK DS      2              ADDRESS OF NEXT CHANNEL, =0 IF LAST
        000.002         562X IOC.DDA DS      2              THREAD JUMP TO DEVICE DRIVER (VIA DEV TABLE)
                        563X
        000.004         564X IOC.FLG DS      1              FILE TYPE FLAGS
        000.001         565X FT.DD   EQU     00000001B      =1 IF DIRECTORY DEVICE
        000.002         566X FT.OR   EQU     00000010B      =1 IF OPEN FOR READ
        000.004         567X FT.OW   EQU     00000100B      =1 IF OPEN FOR WRITE
        000.010         568X FT.OU   EQU     00001000B      =1 IF OPEN FOR UPDATE
        000.020         569X FT.OC   EQU     00010000B      =1 IF OPEN FOR CHARACTER MODE  ;80,02,0C/
        000.003         570X IOC.SQL EQU     *-IOC.DDA      LENGTH OF INFO FOR SEQUENTIAL FILE (FROM IOC)
                        571X
        000.005         572X IOC.GRT DS      2              ADDRESS OF GROUP RESERVATION TABLE
        000.007         573X IOC.SPG DS      1              SECTORS PER GROUP, THIS DEVICE
        000.010         574X IOC.CGN DS      1              CURRENT GROUP NUMBER
        000.011         575X IOC.CSI DS      1              CURRENT SECTOR INDEX (IN CURRENT GROUP)
        000.012         576X IOC.LGN DS      1              LAST GROUP NUMBER
        000.013         577X IOC.LSI DS      1              LAST SECTOR INDEX (IN LAST GROUP)
        000.010         578X IOC.DRL EQU     *-IOC.FLG      LENGTH OF INFO NORMALLY COPIED BACK TO
                        579X *                              THE CHANNEL TABLE
        000.014         580X IOC.DTA DS      2              DEVICE TABLE ADDRESS FOR THIS DEVICE
        000.016         581X IOC.DES DS      2              SECTOR NUMBER OF DIRECTORY ENTRY
        000.020         582X IOC.DEV DS      2              DEVICE CODE
        000.022         583X IOC.UNI DS      1              UNIT NUMBER (0-9)
        000.021         584X IOC.DIL EQU     *-IOC.DDA      LENGTH OF INFO FOR DIRECTORY FILE (FROM IOC)
                        585X
        000.023         586X IOC.DIR DS      DIRELEN        DIRECTORY ENTRY
                        587X
        000.052         588X IOCELEN EQU     *              IOC ENTRY LENGTH
                        589X
        000.001         590X IOCCTD  EQU     1              INDEX OF USER CHANNEL #0 IN CHANTAB (FIRST = 0)
                        591
                        592  ****
```

```
                                595
042.170                         596             ORG     USERFWA-ABS.COD
042.170  377 000                597             DB      377Q,FT.ABS             ABS HEADER
042.172  200 042                598             DW      USERFWA                 LOAD ADDRESS
042.174  207 037                599             DW      MEML-USERFWA            SIZE
042.176  154 073                600             DW      PRS                     ENTRY
                                601
042.200                         602 START       EQU     *                       START HERE AFTER *PRS*
                                603
042.200  315 052 055           604             CALL    BDT                     BUILD DYNAMIC TABLES
042.203  332 211 043           605             JC      EXIT                    PROBLEMS
042.206  076 001                606             MVI     A,1                     PASS 1
042.210  315 251 043           607             CALL    ASM                     ASSEMBLE PASS 1
                                608
042.213  315 011 062           609             CALL    RSF                     REWIND SOURCE FILE
042.216  076 002                610             MVI     A,2
042.220  315 251 043           611             CALL    ASM                     ASSEMBLE PASS 2
042.223  072 210 072           612             LDA     FTFLAG
042.226  376 001                613             CPI     FT.PIC
042.230  302 301 042           614             JNE     H8ASM1                  NOT PIC CODE
                                615
                                616 *           IS PIC CODE, UPDATE LENGTH POINTER IN HEADER
                                617
042.233  052 176 072           618             LHLD    ORG                     LAST BYTE GENERATED             /80.09.BB/
042.236  353                    619             XCHG                            (DE) = LWA                      /80.09.BB/
042.237  052 241 072           620             LHLD    ABSFWA                  (HL) = FWA                      /80.09.BB/
042.242  315 224 030           621             CALL    $CHL                    - (HL)                          /80.09.BB/
042.245  031                   622             DAD     D                       HL = CODE LENGTH                /80.09.BB/
042.246  353                    623             XCHG                            DE = CODE LENGTH                /80.09.BB/
042.247  325                    624             PUSH    D                       SAVE IT.                        /80.09.BB/
042.250  052 241 072           625             LHLD    ABSFWA                  HL = FWA                        /80.09.BB/
042.253  021 003 000           626             LXI     D,PIC.LEN+1             DE = OFFSET FOR LEN HDR         /80.09.BB/
042.256  031                   627             DAD     D                       HL = ORG FOR HEADER BYTES       /80.09.BB/
042.257  042 176 072           628             SHLD    ORG                     SETUP FOR ABV                   /80.09.BB/
042.262  321                   629             POP     D                       REFRESH LENGTH                  /80.09.BB/
042.263  345                   630             PUSH    H                       SAVE ORG                        /80.09.BB/
042.264  173                   631             MOV     A,E                     A = LSB LENGTH                  /80.09.BB/
042.265  315 232 054           632             CALL    ABV                     PLACE IN HEADER                 /80.09.BB/
042.270  341                   633             POP     H                       RESTORE ORG ADDR                /80.09.BB/
042.271  043                   634             INX     H                       BUMP TO NEXT                    /80.09.BB/
042.272  042 176 072           635             SHLD    ORG                     SET IT                          /80.09.BB/
042.275  172                   636             MOV     A,D                     A = MSB LENGTH                  /80.09.BB/
042.276  315 232 054           637             CALL    ABV                     PLACE IN HEADER                 /80.09.BB/
                                638
042.301  072 214 072           639 H8ASM1      LDA     BINFNAM
042.304  247                   640             ANA     A
042.305  312 317 042           641             JZ      H8ASM2                  NO FILE, DONT FINISH IT
042.310  315 146 063           642             CALL    WBB                     FLUSH BUFFER TO DISK
042.313  076 000                643             MVI     A,CN.BIN
042.315  377 046                644             DB      SYSCALL,.CLOSE          CLOSE FILE
                                645
                                646 *           ASSEMBLY COMPLETE,
                                647
042.317  315 054 061           648 H8ASM2      CALL    PAS                     PRINT ASSEMBLY STATISTICS
042.322  315 163 060           649             CALL    FNF6                    FORCE NEW PAGE WITHOUT HEADING  /WCZ062680/
042.325  072 273 072           650             LDA     LISTFB+FB.FLG                                   /WCZ062680/
```

```
042.330  247              651          ANA     A                                          /WCZ062680/
042.331  041 272 072      652          LXI     H,LISTFB                                   /WCZ062680/
042.334  304 254 066      653          CNZ     $FWBRK          EMPTY LISTING BUFFER       /WCZ062680/
                          654  *       LXI     H,LISTFB                                   /80.03.GC/
                          655  *       CALL    $FCLO           CLOSE LISTING FILE         /80.03.GC/
042.337  315 026 064      656          CALL    $CCO            CLEAR CTL-O
                          657
042.342  041 206 061      658          LXI     H,PASA
042.345  377 003          659          DB      SYSCALL,.PRINT  PRINT FINAL MESSAGE
                          660
                          661  *       PROCESS XREF TEMP FILE                             /80.03.GC/
                          662
042.347  072 014 073      663          LDA     TEMPFB+FB.FLG                              /80.03.GC/
042.352  247              664          ANA     A                                          /80.03.GC/
042.353  312 203 043      665          JZ      H8ASM4          NO XREF FILE OPEN          /80.03.GC/
                          666
042.356  052 204 072      667          LHLD    XREFCNT                                    /80.03.GC/
042.361  174              668          MOV     A,H                                        /80.03.GC/
042.362  265              669          ORA     L                                          /80.03.GC/
042.363  302 375 042      670          JNZ     H8ASM3          DATA REALLY WRITTEN OUT    /80.03.GC/
                          671
042.366  076 004          672          MVI     A,CN.TMP                                   /80.03.GC/
042.370  377 055          673          SCALL   .CLEAR          WIPE THE TEMP FILE OUT     /80.03.GC/
042.372  303 203 043      674          JMP     H8ASM4          IGNORE XREF GENERATION     /80.03.GC/
                          675
042.375  001 001 000      676  H8ASM3  LXI     B,1                                        /80.03.GC/
043.000  021 320 031      677          LXI     D,$ZEROS                                   /80.03.GC/
043.003  041 013 073      678          LXI     H,TEMPFB                                   /80.03.GC/
043.006  315 072 066      679          CALL    $FWRIB          WRITE TERMINATOR BYTES     /80.03.GC/
                          680
043.011  041 013 073      681          LXI     H,TEMPFB                                   /80.03.GC/
043.014  315 254 066      682          CALL    $FWBRK          EMPTY TEMP BUFFER          /80.03.GC/
                          683
                          684  *       EXECUTE THE XREF PROGRAM
                          685
                          686  *       PUSH INFORMATION ON THE STACK TO BE PASSED TO 'XREF'   /WCZ062580/
                          687  *       FORMAT OF STACK:
                          688  *       POINTER         DESCRIPTION
                          689  *       -------  ------------------------------
                          690  *       SP+20   WIDE SWITCH
                          691  *       SP+18   PAGE DEPTH
                          692  *       SP+16   FORM DEPTH
                          693  *       SP+14   CURRENT PAGE NUMBER
                          694  *       SP+12   ADDRESS OF TITLE PORTION OF HEADING
                          695  *       SP+10   ADDRESS OF LIST FILE BLOCK
                          696  *       SP+8    ADDRESS OF TEMP FILE BLOCK
                          697  *       SP+6    FWA OF SYMBOL TABLE
                          698  *       SP+4    LIMIT OF SYMBOL TABLE
                          699  *       SP+2    LOWEST ADDRESS OF MEMORY THAT
                          700  *               CAN'T BE OVERLAYED BY 'XREF'
                          701  *       SP      HANDSHAKE VALUE 'XA'                       /WCZ062580/
                          702
043.017  072 257 072      703          LDA     WIDE            PUSH VALUE OF WIDE SWITCH  /WCZ062480/
043.022  365              704          PUSH    PSW               ONTO STACK               /WCZ062480/
043.023  072 260 072      705          LDA     PAGEDP          PUSH VALUE OF PAGE DEPTH   /WCZ062480/
043.026  365              706          PUSH    PSW               ONTO STACK               /WCZ062480/
```

```
043.027  072 261 072   707          LDA     FORMDP          PUSH VALUE OF FORM DEPTH        /WCZ062480/
043.032  365           708          PUSH    PSW               ONTO STACK                    /WCZ062480/
043.033  072 051 073   709          LDA     PAGNUM          PUSH VALUE OF CURRENT PAGE      /WCZ062480/
043.036  365           710          PUSH    PSW               NUMBER ONTO STACK             /WCZ062480/
043.037  041 277 071   711          LXI     H,TTLTXT        PUSH ADDRESS OF TITLE           /WCZ062480/
043.042  345           712          PUSH    H                 ONTO STACK                    /WCZ062480/
043.043  041 272 072   713          LXI     H,LISTFB        PUSH ADDRESS OF LIST FILE       /WCZ062480/
043.046  345           714          PUSH    H                 BLOCK ONTO STACK              /WCZ062480/
043.047  041 013 073   715          LXI     H,TEMPFB        PUSH ADDRESS OF TEMPORARY       /WCZ062480/
043.052  345           716          PUSH    H                 FILE BLOCK ONTO STACK         /WCZ062480/
043.053  052 262 072   717          LHLD    SYMFWA                                          /80.03.GC/
043.056  345           718          PUSH    H               SAVE POINTER TO SYMTAB          /80.03.GC/
043.057  052 264 072   719          LHLD    SYMPTR                                          /80.03.GC/
043.062  345           720          PUSH    H               SAVE LIMIT OF SYMTAB            /80.03.GC/
                       721
                       722  *  DETERMINE LOWEST VALUE OF ADDRESSES PUSHED ONTO THE STACK,  /WCZ062480/
                       723  *  THEN PUSH IT ONTO THE STACK.  THE LOWEST ADDRESS VALUE IS   /WCZ062480/
                       724  *  USED BY 'XREF' TO DETERMINE IF THE XREF PROGRAM WILL OVERLAY /WCZ062480/
                       725  *  ANY OF THE TABLES NEEDED FROM 'ASM'.                        /WCZ062480/
                       726
043.063  021 277 071   727          LXI     D,TTLTXT                                        /WCZ062480/
043.066  041 106 305   728          LXI     H,-LISTFB                                       /WCZ062480/
043.071  031           729          DAD     D                                               /WCZ062480/
043.072  322 100 043   730          JNC     H8ASM3M                                         /WCZ062480/
043.075  021 272 072   731          LXI     D,LISTFB                                        /WCZ062480/
043.100  041 365 304   732  H8ASM3M LXI     H,-TEMPFB                                       /WCZ062480/
043.103  031           733          DAD     D                                               /WCZ062480/
043.104  322 112 043   734          JNC     H8ASM3N                                         /WCZ062480/
043.107  021 013 073   735          LXI     D,TEMPFB                                        /WCZ062480/
043.112  052 262 072   736  H8ASM3N LHLD    SYMFWA                                          /WCZ062480/
043.115  315 224 030   737          CALL    $CHL                                            /WCZ062480/
043.120  031           738          DAD     D                                               /WCZ062480/
043.121  322 130 043   739          JNC     H8ASM30                                         /WCZ062480/
043.124  052 262 072   740          LHLD    SYMFWA                                          /WCZ062480/
043.127  353           741          XCHG                                                    /WCZ062480/
043.130  325           742  H8ASM30 PUSH    D                                               /WCZ062480/
                       743
                       744  *  PLACE 'XA' ON STACK TO SHOW 'XREF' I CALLED HIM.            /WCZ062480/
                       745
043.131  041 101 130   746          LXI     H,'XA'                                          /WCZ062480/
043.134  345           747          PUSH    H                                               /WCZ062480/
                       748
043.135  041 214 043   749          LXI     H,XREF                                          /80.06.GC/
043.140  377 040       750          SCALL   .LINK           Try and run XREF                /80.06.GC/
                       751
043.142  315 136 031   752          CALL    $TYPTX                                          /80.06.sc/
043.145  012 125 156   753          DB      NL,'Unable to run',' '+200Q                     /80.06.sc/
043.148  041 214 043   754          LXI     H,XREF                                          /80.06.sc/
043.151  315 144 031   755          CALL    $TYPTX.                                         /80.06.sc/
043.154  315 251 064   756          CALL    $CRLF                                           /80.06.sc/
043.157  041 013 073   757          LXI     H,TEMPFB                                        /80.06.sc/
043.162  315 075 065   758          CALL    $FCLO           Close the temp file block       /80.06.sc/
000.000                759          ERRNZ   *-H8ASM4                                      /80.06.sc/
                       760
043.165  041 272 072   761  H8ASM4  LXI     H,LISTFB                                        /80.03.GC/
043.170  315 075 065   762          CALL    $FCLO                                           /80.03.GC/
```

```
000.000                  763           ERRNZ   *-EXIT                                  /80.03.GC/
                         764
043.211  257            765  EXIT     XRA     A
043.212  377 000        766  EXIT.    DB      SYSCALL,.EXIT                           /80.03.GC/
                         767
043.214  170 170 156    768  XREF     DB      'xxn:XREF.ABS',0,NL+200Q                /80.06.GC/


                         770  **       CCHIT - CTL-C HIT.
                         771  *
                         772
                         773
043.232  315 136 031    774  CCHIT    CALL    $TYPTX
043.235  136 303        775           DB      '^','C'+200Q
043.237  076 001        776           MVI     A,1
043.241  303 212 043    777           JMP     EXIT.                                   /80.03.GC/


                         779  **       RESTART - RECOVER FROM ERRORS.
                         780  *
                         781  *        THIS CODE IS ENTERED AFTER FILE ERRORS ARE
                         782  *        DISCOVERED AND COMPLAINED ABOUT.
                         783
                         784
043.244  076 001        785  RESTART  MVI     A,1
043.246  303 212 043    786           JMP     EXIT.                                   /80.03.GC/
```

```
                                    790  **      ASM IS CALLED TO MAKE AN ASSEMBLY PASS.
                                    791  *
                                    792  *       IF PASS = 1, ASSEMBLE TEXT, CREATE SYMBOL TABLE, PRODUCE
                                    793  *               NO BINARY OR LISTING.
                                    794  *
                                    795  *       IF PASS = 2, ASSEMBLE TEXT, DEFINE NO SYMBOLS, PRODUCE BINARY
                                    796  *               AND LISTING.
                                    797  *
                                    798  *       ENTRY   (A) = PASS NUMBER (1 OR 2)
                                    799  *       EXIT    'END' STATEMENT READ
                                    800  *               ERRCNT = NUMBER OF STATEMENTS WITH ERRORS
                                    801  *               STATNO = NUMBER OF STATEMENTS READ
                                    802
                                    803
043.251  062 164 072               804  ASM     STA     PASS
043.254  021 064 046               805          LXI     D,NULTITL                                       /78.10.GC/
043.257  315 000 046               806          CALL    TITLE.                                          /78.10.GC/
043.262  021 064 046               807          LXI     D,NULTITL                                       /78.10.GC/
043.265  315 053 046               808          CALL    STL.                                            /78.10.GC/
043.270  041 000 000               809          LXI     H,0
043.273  042 165 072               810          SHLD    ERRCNT          CLEAR ERROR COUNT
043.276  042 170 072               811          SHLD    STATNO
043.301  071                       812          DAD     SP              (HL) = (SP)
043.302  042 307 044               813          SHLD    ASMB            SAVE STACK POINTER VALUE
043.305  257                       814          XRA     A
043.306  062 046 073               815          STA     CNDFLG          CLEAR CONDITIONAL ASSEMBLY
043.311  062 175 072               816          STA     ENDFLG          CLEAR END FLAG
043.314  062 051 073               817          STA     PAGNUM          CLEAR PAGE NUMBER               /10.04.77/
043.317  062 203 072               818          STA     GRPFLG          SET IN FIRST GROUP
043.322  062 212 072               819          STA     CODEFLG         CLEAR 'CODE' PSEUDO SEEN FLAG
043.325  062 210 072               820          STA     FTFLAG          CLEAR CODE GENERATION TYPE FLAG
043.330  062 211 072               821          STA     RELFLG          CLEAR 'AM DOING PIC' FLAG       /80.09.BB/
043.333  076 001                   822          MVI     A,LST.L
043.335  107                       823          MOV     B,A             SET XREF LISTING BIT            /WCZ063080/
043.336  072 025 073               824          LDA     TEMPFB+FB.NAM                                   /WCZ063080/
043.341  247                       825          ANA     A                                               /WCZ063080/
043.342  170                       826          MOV     A,B                                             /WCZ063080/
043.343  312 350 043               827          JZ      ASM0.7                                          /WCZ063080/
043.346  366 010                   828          ORI     LST.R             IF TEMP FILE PRESENT          /WCZ063080/
043.350                            829  ASM0.7  EQU     *                                               /WCZ063080/
043.350  062 172 072               830          STA     LSTCTL
043.353  041 200 042               831          LXI     H,USERFWA       DEFAULT ORG
043.356  042 176 072               832          SHLD    ORG
                                    833
                                    834  *       SET INITIAL LISTING CONTROL BITS
                                    835
043.361  041 173 072               836          LXI     H,LSTCTLS
043.364  072 172 072               837          LDA     LSTCTL
043.367  266                       838          ORA     M               SET FORCED ON BITS
043.370  043                       839          INX     H
000.000                            840          ERRNZ   LSTCTLC-LSTCTLS-1       (HL) = #LSTCTLC
043.371  246                       841          ANA     M               CLEAR FORCED OFF BITS
043.372  062 172 072               842          STA     LSTCTL
                                    843
                                    844  *       ASSEMBLE ANOTHER LINE OF PROGRAM.
                                    845
```

```
043.375  052 176 072   846  ASM1    LHLD    ORG
044.000  042 200 072   847           SHLD    SORG            SAVE COPY OF ORG
044.003  315 320 061   848           CALL    PDL             PREPARE DISPLAY LINE
044.006  052 170 072   849           LHLD    STATNO
044.011  043           850           INX     H
044.012  042 170 072   851           SHLD    STATNO          INCREMENT STATEMENT NUMBER
044.015  315 170 062   852           CALL    UNL             UNPACK NEXT LINE
044.020  312 153 044   853           JZ      ASM4            IS COMMENT
                       854
                       855  *        CRACK OPCODE.
                       856
044.023  041 217 067   857           LXI     H,OPCTAB
044.026  021 166 073   858           LXI     D,OPCODE
044.031  257           859           XRA     A
044.032  315 313 060   860           CALL    LVT             LOCATE VALUE IN TABLE
044.035  332 053 044   861           JC      ASM2            FOUND
044.040  315 026 062   862           CALL    SEF             *O* ERROR
044.043  100           863           DB      ERR.O
044.044  001 000 000   864           LXI     B,0
044.047  120           865           MOV     D,B             GENERATE 3 00 BYTES
044.050  303 321 044   866           JMP     ASM7
                       867
044.053  176           868  ASM2    MOV     A,M             (A) = OPCODE INDEX
000.000                869           ERRNZ   OF.CE-200Q      CODE ASSUMES = 200Q
044.054  027           870           RAL                     CHECK FOR CONDITIONAL ELIGIBILITY
044.055  332 067 044   871           JC      ASM3            WILL PROCESS REGARDLESS OF *IF*
044.060  072 046 073   872           LDA     CNDFLG
044.063  017           873           RRC
044.064  332 162 044   874           JC      ASM5            ASM TO SKIP ASSEMBLING
044.067  176           875  ASM3    MOV     A,M             (A) = OPCODE INDEX
044.070  346 100       876           ANI     OF.LD           SEE IF TO DEFINE LABEL
044.072  345           877           PUSH    H
044.073  314 013 056   878           CZ      DLH             IF TO DEFINE LABEL
044.076  341           879           POP     H
044.077  176           880           MOV     A,M             (A) = OPCODE INDEX
044.100  346 077       881           ANI     77Q             CLEAR FLAG BITS
044.102  043           882           INX     H
044.103  106           883           MOV     B,M             (B) = OPCODE
044.104  345           884           PUSH    H               SAVE ADDRESS IN OPCTAB
044.105  365           885           PUSH    PSW             SAVE INDEX
044.106  376 015       886           CPI     PSIND
044.110  076 001       887           MVI     A,1             ASSUME IS MACHINE CODE, WHICH IS GROUP 2
044.112  332 116 044   888           JC      ASM3.1          IS MACHINE CODE
044.115  170           889           MOV     A,B             IS PSEUDO, USE IT'S GROUP FLAG
044.116  041 203 072   890  ASM3.1  LXI     H,GRPFLG
044.121  266           891           ORA     M               <>0 IF THIS OPERATION IN MAIN GROUP
044.122  167           892           MOV     M,A
044.123  304 033 045   893           CNZ     GCP             GENERATE 'CODE' PSEUDO, IF NECESSARY
044.126  361           894           POP     PSW             (A) = OPERATION INDEX
044.127  007           895           RLC
044.130  041 175 044   896           LXI     H,ASMA
044.133  315 101 030   897           CALL    $DADA.          (HL) = ADDRESS OF ADDRESS
044.136  315 211 030   898           CALL    $HLIHL
044.141  072 164 072   899           LDA     PASS
044.144  017           900           RRC                     (C) SET IF PASS = 1
044.145  170           901           MOV     A,B             (A) = OPCODE
```

```
044.146  343              902           XTHL                    (HL) = ADDRESS OF OPCTAB ENTRY
044.147  021 173 073      903           LXI     D,EXPWRK        (DE) = EXPRESSION POINTER
044.152  311              904           RET                     ENTER PROCESSOR
                          905
                          906  *        HAVE COMMENT
                          907
044.153  072 046 073      908  ASM4     LDA     CNDFLG
044.156  017              909           RRC
044.157  322 001 045      910           JNC     ASM13           LIST IF NOT CONDITIONAL ASSEMBLY
                          911
                          912  *        AM SKIPPING LINE. SEE IF TO LIST.
                          913
044.162  072 172 072      914  ASM5     LDA     LSTCTL
044.165  346 002          915           ANI     LST.I
044.167  312 010 045      916           JZ      ASM14           IGNORE
044.172  303 001 045      917           JMP     ASM13           LIST BY ITSELF
                          918
044.175                   919  ASMA     EQU     *               PROCESSOR JUMP TABLE
                          920
                          921  *        MACHINE OPCODES.
                          922
044.175  336 044          923           DW      SNG             SINGLE BYTE - NO OPERAND
044.177  226 051          924           DW      IMM             IMMEDIATE ARITHEMETIC
044.201  235 051          925           DW      THR             THREE-BYTE OPCODES
044.203  244 051          926           DW      RAO             REG ARITH - TYPE 1
044.205  254 051          927           DW      RAT             REG ARIT - TYPE 2
044.207  267 051          928           DW      RPO             REG PAIR GP 1
044.211  277 051          929           DW      RPT             REG PAIR GP 2
044.213  307 051          930           DW      INX             INX INSTRUCTION
044.215  331 051          931           DW      MVI             MVI INSTRUCTION
044.217  355 051          932           DW      INDX            LDAX, STAX INSTRUCTIONS
044.221  002 052          933           DW      RST             RST INSTRUCTION
044.223  027 052          934           DW      LXI             LXI INSTRUCTION
044.225  046 052          935           DW      MOV             MOV INSTRUCTION
                          936
                          937  *        PSEUDO OPCODES.
                          938
000.015                   939  PSIND    EQU     *-ASMA/2        INDEX OF 1ST PSEUDO OP
044.227  053 045          940           DW      DB              DB
044.231  155 045          941           DW      DS              DS
044.233  207 045          942           DW      DW              DW
044.235  242 045          943           DW      EJECT           EJECT
044.237  120 046          944           DW      ELSE            ELSE
044.241  001 047          945           DW      END             END
044.243  136 046          946           DW      ENDIF           ENDIF
044.245  104 047          947           DW      EQU             EQU
044.247  173 046          948           DW      ERRXX           * OBSOLETE *                    /80.09.BB/
044.251  067 046          949           DW      IF              IF
044.253  261 046          950           DW      LOF             LOF
044.255  242 046          951           DW      LON             LON
044.257  346 046          952           DW      PORG            ORG
044.261  157 047          953           DW      SET             SET
044.263  260 045          954           DW      SPACE           SPACE
044.265  042 046          955           DW      STL             STL
044.267  367 045          956           DW      TITLE           TITLE
044.271  200 047          957           DW      CODE            CODE
```

```
044.273  121 050        958          DW      XTEXT           XTEXT
044.275  051 051        959          DW      NOREF           NOREF                   /WCZ062680/
044.277  153 046        960          DW      ERRZR.          ERRZR                   /80.09.BB/
044.301  157 046        961          DW      ERRNZ.          ERRNZ                   /80.09.BB/
044.303  164 046        962          DW      ERRPL.          ERRPL                   /80.09.BB/
044.305  171 046        963          DW      ERRMI.          ERRMI                   /80.09.BB/
                        964
044.307  000 000        965  ASMB    DW      0               SAVED STACKPOINTER
```

```
                                967  **      MACHINE AND PSEUDO OPCODE PROCESSORS EXIT TO
                                968  *
                                969  *       THESE POINTS:


                                971  **      ASM6 - EXIT WITH 2 BYTES OF DATA
                                972  *
                                973  *       PLACE 2 BYTES IN LINE AND LIST IT.
                                974  *
                                975  *       ENTRY   (B) = 1ST BYTE, (C) = 2ND
                                976
                                977
044.311  170                   978  ASM6    MOV     A,B
044.312  315 354 060           979          CALL    OBB             OUTPUT 1ST BYTE
044.315  171                   980          MOV     A,C
044.316  303 336 044           981          JMP     ASM8


                                983  **      ASM7 - EXIT WITH 3 BYTES OF DATA.
                                984  *
                                985  *       PLACE 3 BYTES IN LINE AND LIST IT
                                986  *
                                987  *       ENTRY   (D) = 1ST BUTE
                                988  *               (C) = 2ND BYTE
                                989  *               (B) = 3RD BYTE
                                990
                                991
044.321  172                   992  ASM7    MOV     A,D
044.322  315 354 060           993          CALL    OBB             1ST BYTE
044.325  315 346 061           994          CALL    RRI             RECORD RELOCATION INFORMATION
044.330  171                   995          MOV     A,C
044.331  315 354 060           996          CALL    OBB             2ND BYTE
044.334  257                   997          XRA     A


                                999  **      ASM8 - EXIT WITH ONE BYTE OF CODE.
                               1000  *
                               1001  *       PLACE 1 BYTE IN LINE AND LIST IT.
                               1002  *
                               1003  *       ENTRY   (A) = VALUE
                               1004
                               1005
044.335  200                   1006 ASM8.    ADD     B               ENTRY TO OUTPUT A+B
044.336  315 354 060           1007 ASM8     CALL    OBB             OUTPUT BYTE
044.341  303 367 044           1008         JMP     ASM11           LIST WITH ORG
```

```
                                1010  **      ASM10 - REQUIRE STATEMENT END, THEN LIST STATEMENT
                                1011  *       WITHOUT ORG,
                                1012  *
                                1013  *       USED BY DW AND DB
                                1014
                                1015
044.344  033                   1016  ASM10   DCX     D
044.345  032                   1017          LDAX    D               (A) = LAST CHARACTER
044.346  315 164 055           1018          CALL    CEF             CHECK FOR END OF FIELD CHARACTER
044.351  312 001 045           1019          JZ      ASM13           GOT A LEGAL TERMINATOR
044.354  315 026 062           1020  FLGERA  CALL    SEF             *A* ERROR
044.357  010                   1021          DB      ERR.A
044.360  303 001 045           1022          JMP     ASM13


                                1024  **      ERR.O, - FLAG *O* ERROR, LIST LINE WITH ORG
                                1025  *
                                1026  *       USED WHEN THE OPCODE IS SYNTATICALLY VALID, JUST ILLEGAL
                                1027  *       IN THAT CONTEXT
                                1028
                                1029
044.363  315 026 062           1030  ERR.O,  CALL    SEF
044.366  100                   1031          DB      ERR.O
                                1032  *       JMP     ASM11           LIST LINE WITH ORG


                                1034  **      ASM11 - LIST LINE WITH ORG,
                                1035  *
                                1036
                                1037
044.367  052 200 072           1038  ASM11   LHLD    SORG            (HL) = SAVED ORG VALUE
044.372  104                   1039          MOV     B,H
044.373  115                   1040          MOV     C,L


                                1042  **      ASM11, - LIST LINE WITH (BC) = ORG
                                1043  *
                                1044
                                1045
044.374  140                   1046  ASM11,  MOV     H,B
044.375  151                   1047          MOV     L,C
```

```
                                1049  **      ASM12 - LIST LINE WITH (HL) AS ORG
                                1050  *
                                1051
                                1052
          044.376  315 144 062  1053  ASM12   CALL    UOL.            UNPACK (HL) TO LINE


                                1055  **      ASM13 - LIST WITHOUT ORG.
                                1056  *
                                1057
                                1058
          045.001  315 074 056  1059  ASM13   CALL    DLL             DISPLAY LISTING LINE
          045.004  257          1060          XRA     A
          045.005  062 202 072  1061          STA     ERRFLG          CLEAR ERROR


                                1063  **      ASM14 - NO LIST.
                                1064  *
                                1065
                                1066
          045.010  072 202 072  1067  ASM14   LDA     ERRFLG
          045.013  247          1068          ANA     A
          045.014  302 001 045  1069          JNZ     ASM13           ERROR - MUST LIST
          045.017  052 307 044  1070          LHLD    ASMB
          045.022  371          1071          SPHL                    RESET STACK POINTER
          045.023  072 175 072  1072          LDA     ENDFLG
          045.026  247          1073          ANA     A
          045.027  300          1074          RNZ                     EXIT IF *END* READ
          045.030  303 375 043  1075          JMP     ASM1            PROCESS ANOTHER CARD


                                1077  **      GCP - GENERATE 'CODE' PSEUDO.
                                1078  *
                                1079  *       GCP IS CALLED AFTER THE GROUP CLASSIFICATION OF EVERY STATEMENT
                                1080  *       HAS BEEN DETERMINED. IF THIS TO-BE-ASSEMBLED STATEMENT HAS
                                1081  *       PUT US INTO GROUP 2, AND NO 'CODE' PSEUDO HAS BEEN ENCOUNTERED,
                                1082  *       THEN WE MUST FAKE UP A
                                1083  *
                                1084  *       CODE    ABS
                                1085  *
                                1086  *       STATEMENT.
                                1087  *
                                1088  *       ENTRY   (A) = (GRPFLG)
                                1089  *       EXIT    NONE
                                1090  *       USES    A,F
                                1091
                                1092
          045.033  072 212 072  1093  GCP     LDA     CODEFLG
          045.036  247          1094          ANA     A
          045.037  300          1095          RNZ                     CODE PSEUDO ENCOUNTERED
```

```
045.040  315 054 031  1096          CALL    $SAVALL         SAVE REGS
045.043  006 000      1097          MVI     B,FT.ABS        DO ABS
045.045  315 260 047  1098          CALL    CODE2           PROCESS CODE PSEUDO
045.050  303 047 031  1099          JMP     $RSTALL         RESTORE AND RETURN
```

```
                                1102  **      DB - DEFINE BYTE.
                                1103  *
                                1104  *       DB      VAL,...,VAL
                                1105
                                1106
045.053                         1107  DB      EQU     *
045.053  315 141 062           1108          CALL    UOL             UNPACK ORG INTO LINE
045.056  325                   1109  DB1     PUSH    D
                                1110
                                1111  *       EXAMINE NEXT ELEMENT.
                                1112
045.057  032                   1113          LDAX    D
045.060  376 047               1114          CPI     QUOTE
045.062  302 114 045           1115          JNE     DB3             NOT QUOTE
                                1116
                                1117  *       HAVE QUOTED STRING. SEE IF IS PART OF EXPRESSION, OR JUST
                                1118  *       A STAND-ALONE.
                                1119
045.065  023                   1120          INX     D
045.066  315 255 060           1121  DB2     CALL    GSC             GET STRING CHARACTER
045.071  302 066 045           1122          JNZ     DB2
045.074  032                   1123          LDAX    D
045.075  247                   1124          ANA     A                                       /80.02.GC/
045.076  372 127 045           1125          JM      DB4             MARKED CHARACTER        /80.02.GC/
045.101  315 164 055           1126          CALL    CEF             CHECK FOR END OF FIELD  /80.02.GC/
045.104  312 127 045           1127          JZ      DB4                                     /80.02.GC/
045.107  376 054               1128          CPI     ','             CHECK FOR EXPRESSION    /80.02.GC/
045.111  312 127 045           1129          JZ      DB4                                     /80.02.GC/
                                1130
                                1131  *       HAVE BYTE EXPRESSION
                                1132
045.114  321                   1133  DB3     POP     D
045.115  315 054 057           1134          CALL    E8B             EVALUATE TO 8 BITS
045.120  171                   1135          MOV     A,C             (A) = VALUE
045.121  315 354 060           1136          CALL    OBB             OUTPUT BINARY BYTE
045.124  303 143 045           1137          JMP     DB6
                                1138
                                1139  *       HAVE QUOTED STRING
                                1140
045.127  321                   1141  DB4     POP     D
045.130  023                   1142          INX     D
045.131  257                   1143          XRA     A
045.132  304 354 060           1144  DB5     CNZ     OBB             OUTPUT BINARY BYTE (SKIP 1ST TIME)
045.135  315 255 060           1145          CALL    GSC             GET SGRING CHARACTER
045.140  302 132 045           1146          JNZ     DB5             IF MORE
                                1147
                                1148  *       END OF BYTE VALUE. SEE IF MORE FOLLOW
                                1149
045.143  032                   1150  DB6     LDAX    D
045.144  023                   1151          INX     D
045.145  376 054               1152          CPI     ','
045.147  312 056 045           1153          JE      DB1             IF MORE
045.152  303 344 044           1154          JMP     ASM10           REQUIRE END AND LIST LINE
```

```
                                1158  **      DS - DEFINE STORAGE.
                                1159  *
                                1160  *       DS      EXPR
                                1161
                                1162
          045.155               1163  DS      EQU     *
          045.155  315 105 057  1164          CALL    EPO             EVALUATE FOR PASS 1
          045.160  302 367 044  1165          JNZ     ASM11           EXIT - ERROR
          045.163  052 176 072  1166          LHLD    ORG
          045.166  011          1167          DAD     B
          045.167  332 200 045  1168          JC      DS1             IF WRAP-AROUND                  /80.09.BB/
          045.172  042 176 072  1169          SHLD    ORG
          045.175  303 367 044  1170          JMP     ASM11           LIST WITH ORG
          045.200  315 026 062  1171  DS1     CALL    SEF             SET ERROR                       /80.09.BB/
          045.203  020          1172          DB      ERR.V           VALUE ERROR                     /80.09.BB/
          045.204  303 374 044  1173          JMP     ASM11.          LIST ORG = DS VALUE             /80.09.BB/
```

```
                                1176  **      DW - DEFINE WORD.
                                1177  *
                                1178  *       DW      EXP1,...,EXPN
                                1179
                                1180
045.207                         1181  DW      EQU     *
045.207  315 141 062           1182          CALL    UOL             UNPACK ORG INTO LINE
                                1183
                                1184  *       DECODE NEXT ELEMENT
                                1185
045.212  315 355 053           1186  DW1     CALL    EVL
045.215  315 346 061           1187          CALL    RRI             RECORD RELOCATION INFORMATION
045.220  171                   1188          MOV     A,C
045.221  315 354 060           1189          CALL    OBB             OUTPUT BINARY BUTE
045.224  170                   1190          MOV     A,B
045.225  315 354 060           1191          CALL    OBB             OUTPUT 2ND HALF
045.230  032                   1192          LDAX    D
045.231  023                   1193          INX     D
045.232  376 054               1194          CPI     ','
045.234  312 212 045           1195          JE      DW1             IF MORE TO GO
045.237  303 344 044           1196          JMP     ASM10           ENSURE END AND LIST
```

```
                                1199  **      EJECT - SET PAGE EJECT.
                                1200  *
                                1201
                                1202
045.242  315 174 055            1203  EJECT   CALL    CLE             CHECK LISTING ELIGIBILITY
045.245  312 010 045            1204          JZ      ASM14           NOT TO LIST
045.250  076 001                1205          MVI     A,1
045.252  062 047 073            1206          STA     EJEFLG          FORCE PAGE EJECT
045.255  303 010 045            1207          JMP     ASM14           NO LIST


                                1209  **      SPACE   N,M
                                1210  *
                                1211  *       SPACE   N LINES IF < M LINES REMAIN ON THE PAGE, OTHERWISE, EHECT.
                                1212
                                1213
045.260  332 010 045            1214  SPACE   JC      ASM14           IGNORE IF PASS 1
045.263  315 174 055            1215          CALL    CLE             CHECK LISTING ELIGIBILITY
045.266  312 010 045            1216          JZ      ASM14           NO LISTING
045.271  315 054 057            1217          CALL    E8B             EVALUATE 8 BIT EXPRESSION
045.274  305                    1218          PUSH    B               SAVE N
045.275  032                    1219          LDAX    D
045.276  376 054                1220          CPI     ','
045.300  302 316 045            1221          JNE     SPC1            NO M
045.303  023                    1222          INX     D
045.304  315 054 057            1223          CALL    E8B             EVALUATE M
045.307  072 050 073            1224          LDA     LINCNT
045.312  271                    1225          CMP     C
045.313  332 242 045            1226          JC      EJECT           IF TO FORCE EJECT
                                1227
045.316  301                    1228  SPC1    POP     B               (C) = N
045.317  072 260 072            1229          LDA     PAGEDP
045.322  127                    1230          MOV     D,A             (D) = PAGSIZ
045.323  072 050 073            1231          LDA     LINCNT
045.326  272                    1232          CMP     D
045.327  312 010 045            1233          JE      ASM14           AT TOP OF PAGE, DONT SPACE
045.332  221                    1234          SUB     C               (A) = PROPOSED NEW LINE NUMBER
045.333  332 242 045            1235          JC      EJECT           WILL BRING NEW PAGE
045.336  315 233 055            1236  SPC2    CALL    COL             COUNT OUTPUT LINE
045.341  305                    1237          PUSH    B               SAVE COUNT
045.342  041 272 072            1238          LXI     H,LISTFB
045.345  021 366 045            1239          LXI     D,SPCA
045.350  001 001 000            1240          LXI     B,1
045.353  315 072 066            1241          CALL    $FWRIB          WRITE NEW LINE
045.356  301                    1242          POP     B               (C) = COUNT
045.357  015                    1243          DCR     C
045.360  302 336 045            1244          JNZ     SPC2            IF NOT DONE
045.363  303 010 045            1245          JMP     ASM14           EXIT WITH NO LIST
                                1246
045.366  012                    1247  SPCA    DB      NL              SPACE LINE
```

```
                        1250  **      TITLE - SETUP PAGE TITLE.
                        1251  *
                        1252  *       TITLE 'NEW TITLE'
                        1253
                        1254
045.367 315 000 046     1255  TITLE   CALL    TITLE.
045.372 332 354 044     1256          JC      FLGERA
045.375 303 010 045     1257          JMP     ASM1A
                        1258
046.000 041 276 071     1259  TITLE.  LXI     H,TTLTXT-1      (HL) = ADDRESS FOR TEXT
046.003 006 062         1260          MVI     B,TTXTL         (B) = MAX LENGTH
046.005 032             1261  TTL1    LDAX    D
046.006 376 047         1262          CPI     QUOTE
046.010 302 040 046     1263          JNE     TTL4            NO TITLE
046.013 023             1264          INX     D
046.014 005             1265  TTL2    DCR     B
046.015 312 040 046     1266          JZ      TTL4            TOO MAY CHARACTERS
046.020 043             1267          INX     H
046.021 315 255 060     1268          CALL    GSC             GET STRING CHARACTER
046.024 167             1269          MOV     M,A
046.025 302 014 046     1270          JNZ     TTL2            IF MORE TO GO
                        1271
                        1272  *       FILL REMAINDER OF LINE WITH BLANKS.
                        1273
046.030 066 040         1274  TTL3    MVI     M,' '
046.032 043             1275          INX     H
046.033 005             1276          DCR     B
046.034 302 030 046     1277          JNZ     TTL3
046.037 311             1278          RET
                        1279
046.040 067             1280  TTL4    STC                     FLAG ERROR
046.041 311             1281          RET


                        1283  **      STL - SUBTITLE LINE
                        1284  *
                        1285  *       STL     'NEW SUB-TITLE'
                        1286
                        1287
046.042 315 053 046     1288  STL     CALL    STL.
046.045 332 354 044     1289          JC      FLGERA
046.050 303 010 045     1290          JMP     ASM1A
                        1291
046.053 006 062         1292  STL.    MVI     B,STXTL
046.055 041 006 072     1293          LXI     H,STLTXT-1
046.060 315 005 046     1294          CALL    TTL1
046.063 311             1295          RET
                        1296
046.064 047 040 047     1297  NULTITL DB      047Q,' ',047Q   "' '"
```

```
                                1300  **      IF - INITIATE CONDITIONAL ASSEMBLY.
                                1301  *
                                1302  *       IF      EXPR
                                1303  *
                                1304  *       ASSEMBLE IF EXPR = 0
                                1305
                                1306
046.067  315 105 057            1307  IF      CALL    EPO             EVALUATE PASS 1
046.072  302 354 044            1308          JNZ     FLGERA          ERROR
046.075  041 046 073            1309          LXI     H,CNDFLG
046.100  176                    1310          MOV     A,M
046.101  247                    1311          ANA     A
046.102  302 354 044            1312          JNZ     FLGERA          CONDITIONAL ASSEMBLY ALREADY IN EFFECT
046.105  170                    1313          MOV     A,B
046.106  261                    1314          ORA     C
046.107  066 200                1315          MVI     M,200Q
046.111  312 115 046            1316          JZ      IF1             AM TO ASSEMBLE
046.114  064                    1317          INR     M               AM TO SKIP
046.115  303 374 044            1318  IF1     JMP     ASM11.          LIST WITH (BC) = ORG


                                1320  **      ELSE - TOGGLE CONDITIONAL ASSEMBLY.
                                1321  *
                                1322  *       ELSE
                                1323
                                1324
046.120  041 046 073            1325  ELSE    LXI     H,CNDFLG
046.123  176                    1326          MOV     A,M
046.124  247                    1327          ANA     A
046.125  362 354 044            1328          JP      FLGERA          CONDITIONAL ASSEMBLY NOT IN EFFECT
046.130  356 001                1329          XRI     1
046.132  167                    1330          MOV     M,A
046.133  303 001 045            1331          JMP     ASM13           PRINT NO INFORMATION


                                1333  **      ENDIF - COMPLETE CONDITIONAL PROCESSING.
                                1334  *
                                1335  *       ENDIF
                                1336
                                1337
046.136  041 046 073            1338  ENDIF   LXI     H,CNDFLG
046.141  176                    1339          MOV     A,M
046.142  066 000                1340          MVI     M,0
046.144  247                    1341          ANA     A
046.145  312 354 044            1342          JZ      FLGERA          CONDITIONAL ASSEMBLY NOT IN EFFECT
046.150  303 001 045            1343          JMP     ASM13           LIST WITH NO INFO
```

```
                                1346  **      ERRXX - CONDITIONAL ERRORS.
                                1347  *
                                1348  *       ERRZR EXPR
                                1349  *       ERRNZ  EXPR
                                1350  *       ERRPL EXPR
                                1351  *       ERRMI EXPR
                                1352  *
                                1353  *       FLAG A *P* ERROR IF EXPR MATCHES CONDITION.
                                1354
046.153  257                   1355  ERRZR.  XRA     A               SET INDEX               /80.09.BB/
046.154  303 173 046           1356          JMP     ERRXX           ENTER COMMON CODE       /80.09.BB/
                                1357
046.157  076 001               1358  ERRNZ.  MVI     A,1             INDEX = 1               /80.09.BB/
046.161  303 173 046           1359          JMP     ERRXX                                   /80.09.BB/
                                1360
046.164  076 002               1361  ERRPL.  MVI     A,2             INDEX = 2               /80.09.BB/
046.166  303 173 046           1362          JMP     ERRXX                                   /80.09.BB/
                                1363
046.171  076 003               1364  ERRMI.  MVI     A,3             INDEX = 3               /80.09.BB/
                                1365  *       JMP     ERRXX                                   /80.09.BB/
000.000                         1366          ERRNZ   *-ERRXX         INSURE FALL THROUGH     /80.09.BB/
                                1367
046.173                         1368  ERRXX   EQU     *                                       /80.09.BB/
046.173  365                   1369          PUSH    PSW             SAVE INDEX CODE         /80.09.BB/
046.174  315 355 053           1370          CALL    EVL
046.177  361                   1371          POP     PSW             (A) = TYPE INDEX        /80.09.BB/
046.200  041 374 044           1372          LXI     H,ASM11.        LIST WITH (BC) = ORG
046.203  315 213 046           1373          CALL    ERR1
046.206  315 026 062           1374          CALL    SEF             *P* ERROR
046.211  200                   1375          DB      ERR.P
046.212  351                   1376          PCHL                    GO TO ASM12
                                1377
046.213  315 076 031           1378  ERR1    CALL    $TBRA
046.216  004                   1379          DB      ERRZR-*
046.217  007                   1380          DB      ERRNZ-*
046.220  012                   1381          DB      ERRPL-*
046.221  015                   1382          DB      ERRMI-*

046.222  170                   1384  ERRZR   MOV     A,B
046.223  261                   1385          ORA     C
046.224  310                   1386          RZ      ERR1            *P* ERROR
046.225  351                   1387          PCHL                    LIST WITH (HL) = ORG

046.226  170                   1389  ERRNZ   MOV     A,B
046.227  261                   1390          ORA     C
046.230  300                   1391          RNZ     ERR1            *P* ERROR
046.231  351                   1392          PCHL
```

```
046.232  170          1394  ERRPL   MOV     A,B
046.233  247          1395          ANA     A
046.234  360          1396          RP      ERR1            *P* ERROR
046.235  351          1397          PCHL

046.236  170          1399  ERRMI   MOV     A,B
046.237  247          1400          ANA     A
046.240  370          1401          RM      ERR1            *P* ERROR
046.241  351          1402          PCHL
```

```
                                1405  **      LON - LISTING ON.
                                1406  *
                                1407  *       LON     CCC
                                1408  *
                                1409  *       TURN OPTIONS ON. OPTIONS =
                                1410  *
                                1411  *       L       MASTER LISTING
                                1412  *       I       IF-SKIPPED LINES
                                1413  *       C       INCLUDED CODE
                                1414  *       R       X-REF                                   /80.03.sc/
                                1415  *       G       GENERATED CODE
                                1416
                                1417
046.242  315 304 046           1418  LON     CALL    LST
046.245  312 001 045           1419          JZ      ASM13           ALL DONE
046.250  266                   1420          ORA     M
046.251  041 174 072           1421          LXI     H,LSTCTLC
046.254  246                   1422          ANA     M               CLEAR BITS MENTIONED IN /N: SWITCH
046.255  002                   1423          STAX    B
046.256  303 242 046           1424          JMP     LON             PROCESSES NEXT


                                1426  **      LOF - LISTING OFF.
                                1427  *
                                1428  *       LOF     CCC
                                1429  *
                                1430  *       TURN LON OPTIONS BACK OFF.
                                1431
                                1432
046.261  315 304 046           1433  LOF     CALL    LST
046.264  312 001 045           1434          JZ      ASM13           DONE
046.267  365                   1435          PUSH    PSW             SAVE OLD VALUE
046.270  176                   1436          MOV     A,M
046.271  057                   1437          CMA
046.272  341                   1438          POP     H               (H) = OLD (A)
046.273  244                   1439          ANA     H               (A) = (.NOT.BIT).AND.LSTCTL
046.274  041 173 072           1440          LXI     H,LSTCTLS
046.277  266                   1441          ORA     M               SET BITS MENTIONED IN /L: SWITCH
046.300  002                   1442          STAX    B
046.301  303 261 046           1443          JMP     LOF


                                1445  **      LST - PERFORM LON AND LOF PRESET.
                                1446  *
                                1447  *       LST PERFORMS SOME FIXED TASKS FOR LON AND LOF.
                                1448  *
                                1449  *       ENTRY   (DE) = NEXT EXPRESSION CHARACTER
                                1450  *       EXIT    'Z' SET IF END OF LIST
                                1451  *               'Z' CLEAR IF VALID CHARACTER
                                1452  *               IF NOT AT END:
                                1453  *               (DE) UPDATED
                                1454  *               (BC) = #LSTCTL
                                1455  *               (A) = (LSTCTL)
```

```
                        1456  *               (HL) = ADDRESS OF OPTION BIT
                        1457
                        1458
046.304  032           1459  LST     LDAX    D
046.305  315 164 055     1460          CALL    CEF             CHECK FOR END OF FIELD CHARACTER
046.310  310           1461          RE                      END OF FIELD
046.311  023           1462          INX     D
046.312  041 332 046     1463          LXI     H,LSTA
046.315  315 304 064     1464          CALL    $TBLS           TABLE SEARCH
046.320  302 354 044     1465          JNZ     FLGERA          NOT GOOD OPTION
046.323  366 001         1466          ORI     1               CLEAR 'Z'
046.325  001 172 072     1467          LXI     B,LSTCTL
046.330  012           1468          LDAX    B
046.331  311           1469          RET
                        1470
                        1471
046.332                 1472  LSTA    EQU     *               OPTION TABLE
046.332  114 001         1473          DB      'L',LST.L
046.334  107 200         1474          DB      'G',LST.G
046.336  111 002         1475          DB      'I',LST.I
046.340  122 010         1476          DB      'R',LST.R                               /80.03.GC/
046.342  103 004         1477          DB      'C',LST.C
046.344  000 000         1478          DB      0,0
```

```
                                1481  **      ORG - SET ORIGIN COUNTER.
                                1482  *
                                1483  *       ORG     EXPR
                                1484  *
                                1485  *       EXPRESSION MUST EVALUATE PASS 1
                                1486
                                1487
046.346  072 210 072           1488  PORG    LDA     FTFLAG
046.351  247                   1489          ANA     A
000.000                        1490          ERRNZ   FT.ABS
046.352  302 363 044           1491          JNZ     ERR.O.          OPCODE ERROR
046.355  315 105 057           1492          CALL    EPO             EVALUATE PASS 1
046.360  302 354 044           1493          JNZ     FLGERA          BAD VALUE
046.363  140                   1494          MOV     H,B
046.364  151                   1495          MOV     L,C
046.365  042 176 072           1496          SHLD    ORG             SET NEW ORG
046.370  042 200 072           1497          SHLD    SORG            SET TO COME OUT ON LISTING
046.373  315 013 056           1498          CALL    DLH             DEFINE LABEL HERE
046.376  303 367 044           1499          JMP     ASM11           LIST WITH ORG
```

```
                                1502  **      END - END OF PROGRAM.
                                1503  *
                                1504  *       END
                                1505
                                1506
047.001  315 141 062            1507  END     CALL    UOL             UNPACK ORG INTO LINE
047.004  072 206 072            1508          LDA     XTXFLG
047.007  247                    1509          ANA     A
047.010  302 363 044            1510          JNZ     ERR.O.          AM IN XTEXT
047.013  072 210 072            1511          LDA     FTFLAG
000.000                         1512          ERRNZ   FT.ABS
047.016  247                    1513          ANA     A
047.017  302 041 047            1514          JNZ     END1            IS PIC, CANNOT TAKE ENTRY POINT
047.022  315 105 057            1515          CALL    EPO             EVALUATE PASS 1
047.025  151                    1516          MOV     L,C
047.026  140                    1517          MOV     H,B
047.027  042 245 072            1518          SHLD    ABSENT          SET PROGRAM ENTRY POINT ADDRESS
047.032  257                    1519          XRA     A
047.033  315 354 060            1520          CALL    OBB             ADD 00 BYTE TO END
047.036  303 074 047            1521          JMP     END3            FLAG END AND EXIT
                                1522
                                1523  *       IS PIC, DO PASS-DEPENDANT STUFF..
                                1524
047.041  072 164 072            1525  END1    LDA     PASS
047.044  075                    1526          DCR     A
047.045  302 071 047            1527          JNZ     END2            PASS 2
047.050  052 176 072            1528          LHLD    ORG
047.053  353                    1529          XCHG                    DE = ADDR OF PIC TABLE          /80.09.BB/
047.054  052 241 072            1530          LHLD    ABSFWA          HL = FWA OF CODE                /80.09.BB/
047.057  315 224 030            1531          CALL    $CHL            - HL                            /80.09.BB/
047.062  031                    1532          DAD     D               DE = OFFSET FROM 0 TO PICTAB    /80.09.BB/
047.063  042 255 072            1533          SHLD    PICPTR          SET ADDRESS  OF RELOCATION TABLE
047.066  303 074 047            1534          JMP     END3            FLAG END AND EXIT
                                1535
047.071  315 205 060            1536  END2    CALL    GRT             PIC AND PASS2 - GENERATE RELOC TABLE
                                1537
                                1538
                                1539  **      ENTER HERE TO FORCE END OF PASS
                                1540
047.074                         1541  END.    EQU     *
                                1542
047.074  076 001                1543  END3    MVI     A,1
047.076  062 175 072            1544          STA     ENDFLG
047.101  303 001 045            1545          JMP     ASM13           LIST, ORG ALREADY DECODED
```

```
                                1548  **      EQU - EQUIVALENCE SYMBOL.
                                1549  *
                                1550  * LAB   EQU     EXPR
                                1551  *
                                1552  *       ASSIGN VALUE OF *EXPR* TO LABEL.
                                1553  *
                                1554  *       EXPRESSION MUST EVALUATE PASS 1
                                1555
                                1556
047.104  365                   1557  EQU     PUSH    PSW             SAVE PASS FLAG                  /80.03.GC/
047.105  076 002                1558          MVI     A,XT.EQU                                        /80.03.GC/
047.107  315 213 057            1559          CALL    ESR             ENTER SYMBOLIC REFERENCE        /80.03.GC/
                                1560
047.112  315 105 057            1561          CALL    EPQ             EVALUATE PASS ONE
047.115  302 001 045            1562          JNZ     ASM13           ERROR
047.120  361                   1563          POP     PSW             RESTORE PASS FLAG
047.121  322 135 047            1564          JNC     EQU1            PASS 2
                                1565
                                1566  *       PASS 1
                                1567
047.124  021 000 002            1568          LXI     D,ST.EQU*256+ST.UND
047.127  315 333 055            1569          CALL    DEF             DEFINE SYMBOL
047.132  303 001 045            1570          JMP     ASM13           EXIT
                                1571
                                1572  *       PASS 2
                                1573
047.135  021 155 073            1574  EQU1    LXI     D,LABEL
047.140  315 041 062            1575          CALL    SST
047.143  176                   1576          MOV     A,M
000.000                         1577          ERRNZ   ST.DBL-200Q     ASSUMES .= 200Q
047.144  027                   1578          RAL
047.145  322 374 044            1579          JNC     ASM11.          OK, LIST WITH (BC) = ORG
047.150  315 026 062            1580          CALL    SEF             *D* ERROR
047.153  004                   1581          DB      ERR.D
047.154  303 374 044            1582          JMP     ASM11.


                                1584  **      SET - SET VALUE.
                                1585  *
                                1586  * LAB   SET     EXPR
                                1587  *
                                1588  *       SET PERFORMS THE SAME FUNCTION AS EQU, BUT THE ASSIGNMENT IS MAD
                                1589  *       PASS 2, AND A VALUE MAY BE RE-SET.
                                1590
                                1591
047.157                         1592  SET     EQU     *
047.157  076 003                1593          MVI     A,XT.SET                                        /80.03.GC/
047.161  315 213 057            1594          CALL    ESR                                             /80.03.GC/
                                1595
047.164  315 355 053            1596          CALL    EVL             EVALUATE
047.167  021 003 003            1597          LXI     D,ST.SET*256+ST.SET
047.172  315 333 055            1598          CALL    DEF
047.175  303 374 044            1599          JMP     ASM11.          LIST WITH (BC) = ORG
```

```
                                1603  **      CODE - PROCESS CODE PSEUDO.
                                1604  *
                                1605  *       CODE    ABS             GENERATE ABS CODE
                                1606  *       CODE    PIC             GENERATE PIC CODE
                                1607  *       CODE    +R              RELOCATE THIS CODE              /80.09.BB/
                                1608  *       CODE    -R              DO NOT RELOCATE FOLLOWING       /80.09.BB/
                                1609
047.200                         1610  CODE    EQU     *
047.200  315 206 047           1611          CALL    CODE0           PROCESS PSEUDO
047.203  303 367 044           1612          JMP     ASM11           LIST WITH ORG
                                1613
047.206                         1614  CODE0   EQU     *                                               /80.09.BB/
047.206  032                   1615          LDAX    D               SEE IF + OR -                   /80.09.BB/
047.207  376 053               1616          CPI     '+'             IS PLUS                         /80.09.BB/
047.211  312 070 050           1617          JZ      CODER                                           /80.09.BB/
047.214  376 055               1618          CPI     '-'             IF MINUS                        /80.09.BB/
047.216  312 070 050           1619          JZ      CODER           IF SO.                          /80.09.BB/
047.221  041 203 072           1620          LXI     H,GRPFLG        NOT +/-, CHECK GROUP            /80.09.BB/
047.224  176                   1621          MOV     A,M
047.225  064                   1622          INR     M
047.226  247                   1623          ANA     A
047.227  312 235 047           1624          JZ      CODE1           IN GROUP 1
047.232  303 363 044           1625          JMP     ERR.O.          *O* ERROR, NOT IN 1ST GROUP
                                1626
                                1627  *       AM IN 1ST GROUP.
                                1628
047.235  032                   1629  CODE1   LDAX    D
047.236  376 101               1630          CPI     'A'
047.240  006 000               1631          MVI     B,FT.ABS
047.242  312 260 047           1632          JE      CODE2           GOT TYPE
047.245  004                   1633          INR     B               (B) = FT.PIC
047.246  376 120               1634          CPI     'P'
000.000                         1635          ERRNZ   FT.PIC-FT.ABS-1
047.250  312 260 047           1636          JE      CODE2           GOT IT
047.253  315 026 062           1637          CALL    SEF
047.256  010                   1638          DB      ERR.A           CANT UNDERSTAND OPERAND
047.257  311                   1639          RET
                                1640
                                1641  *       GOT A TYPE SPECIFIED
                                1642  *
                                1643  *       (B) = FT.XXX
                                1644
047.260  170                   1645  CODE2   MOV     A,B
047.261  062 210 072           1646          STA     FTFLAG          SET TYPE
000.000                         1647          ERRNZ   FT.PIC*64-ST.REL
047.264  017                   1648          RRC
047.265  017                   1649          RRC
047.266  062 211 072           1650          STA     RELFLG          RELFLG = ST.REL IF FT.PIC
047.271  076 001               1651          MVI     A,1
047.273  062 212 072           1652          STA     CODEFLG         SET CODE PSEUDO ENCOUNTERED
047.276  170                   1653          MOV     A,B
000.000                         1654          ERRNZ   FT.ABS
047.277  247                   1655          ANA     A
047.300  312 001 050           1656          JZ      CODE2.5         IS ABS
047.303  041 000 000           1657          LXI     H,0
047.306  042 241 072           1658          SHLD    ABSFWA          SET CODE DISPLACEMENT =0
```

```
047.311  041 006 000  1659          LXI   H,PIC.COD
047.314  042 176 072  1660          SHLD  ORG          SET DEFAULT ORG = 0 (PIC.COD FOR 1ST USER GENERATED BYTE)
047.317  042 200 072  1661          SHLD  SORG
047.322  023          1662          INX   D            POINT TO CHAR AFTER 'P'          /80.09.BB/
047.323  032          1663          LDAX  D            GET CHARACTER                    /80.09.BB/
047.324  376 054      1664          CPI   ','          WAS P,                           /80.09.BB/
047.326  302 001 050  1665          JNZ   CODE2.5      NO, GO ON                        /80.09.BB/
                      1666
047.331  023          1667          INX   D            POINT TO EXPRESSION              /80.09.BB/
047.332  315 105 057  1668          CALL  EPO          EVALUATE PASS ONE                /80.09.BB/
047.335  302 354 044  1669          JNZ   FLGERA       IF A PROBLEM                     /80.09.BB/
047.340  170          1670          MOV   A,B          CHECK TO SEE                     /80.09.BB/
047.341  247          1671          ANA   A             IF HE LEFT                    /80.09.BB/
047.342  302 362 047  1672          JNZ   CODE2.2       ROOM FOR 6                   /80.09.BB/
047.345  171          1673          MOV   A,C           BYTE HEADER                  /80.09.BB/
047.346  376 006      1674          CPI   PIC.COD       OR NOT.                      /80.09.BB/
047.350  322 362 047  1675          JNC   CODE2.2       IF OK.                       /80.09.BB/
047.353  315 026 062  1676          CALL  SEF          ANNOUNCE ERROR                   /80.09.BB/
047.356  020          1677          DB    ERR.V        VALUE ERROR                      /80.09.BB/
047.357  303 001 050  1678          JMP   CODE2.5      ASSUME NO ORG GIVEN              /80.09.BB/
047.362  151          1679  CODE2.2 MOV   L,C          GET VALUES FROM BC               /80.09.BB/
047.363  140          1680          MOV   H,B                    AND SET IN HL         /80.09.BB/
047.364  042 176 072  1681          SHLD  ORG          SET ORG VALUE                    /80.09.BB/
047.367  042 200 072  1682          SHLD  SORG         SET SORG VALUE                   /80.09.BB/
047.372  021 372 377  1683          LXI   D,-PIC.COD    ALLOW ROOM FOR HEADER            /80.09.BB/
047.375  031          1684          DAD   D            SUBTRACT FROM REQ. FWA           /80.09.BB/
047.376  042 241 072  1685          SHLD  ABSFWA       SET FWA OF CODE GENERATOR        /80.09.BB/
050.001  072 164 072  1686  CODE2.5 LDA   PASS
050.004  075          1687          DCR   A
050.005  310          1688          RZ                 PASS 1
                      1689
                      1690  *       IS PASS 2. GENERATE BINARY HEADER
050.006  072 210 072  1691          LDA   FTFLAG       GET FILE TYPE FLAG               /80.09.BB/
050.011  075          1692          DCR   A
000.000               1693          ERRNZ FT.PIC-1
050.012  312 052 050  1694          JZ    CODE3        IS PIC
                      1695
                      1696  *       IS ABSOLUTE. GENERATE
                      1697  *
                      1698  *       3770,FT.ABS,FWA,LWA,ENTRY
                      1699
050.015  052 247 072  1700          LHLD  ABSLWA
050.020  353          1701          XCHG
050.021  052 241 072  1702          LHLD  ABSFWA       (HL) = FWA GENED CODE
050.024  315 224 030  1703          CALL  $CHL
050.027  031          1704          DAD   D            (HL) = LENGTH
050.030  042 243 072  1705          SHLD  ABSLEN       SET LENGTH
050.033  076 010      1706          MVI   A,ABS.COD
050.035  062 236 072  1707          STA   BINSKW       SET BINARY SKEW IN FILE
050.040  315 164 064  1708          CALL  $MOVEL
050.043  010 000 237  1709          DW    ABS.COD,ABSHDR,BINBFR   SET HEADER IN BUFFER
050.051  311          1710          RET
                      1711
                      1712  *       IS PIC, GENERATE
                      1713  *
                      1714  *       3770,FT.PIC,LEN,POINTER
```

```
                          1715
050.052  257              1716  CODE3   XRA     A
050.053  062 236 072       1717          STA     BINSKW          NO BINARY SKEW DUE TO HEADER
050.056  315 164 064       1718          CALL    $MOVEL          SET HEADER IN BUFFER
050.061  006 000 251       1719          DW      PIC.COD,PICHDR,BINBFR
050.067  311               1720          RET
                          1721
050.070                   1722  CODER   EQU     *                                       /80.09.BB/
050.070  117               1723          MOV     C,A             SAVE +/-                /80.09.BB/
050.071  023               1724          INX     D               GET 'R'                 /80.09.BB/
050.072  032               1725          LDAX    D               A = 'R'                 /80.09.BB/
050.073  376 122           1726          CPI     'R'             IS IT?                  /80.09.BB/
050.075  312 105 050       1727          JZ      CODER1          OK                      /80.09.BB/
050.100  315 026 062       1728          CALL    SEF             ANNOUNCE ERROR          /80.09.BB/
050.103  010               1729          DB      ERR.A                                   /80.09.BB/
050.104  311               1730          RET                     RETURN TO MAIN CODE     /80.09.BB/
050.105  171               1731  CODER1  MOV     A,C             A = +/-                 /80.09.BB/
050.106  041 211 072       1732          LXI     H,RELFLG        HL = FLAG ADDRESS       /80.09.BB/
050.111  066 100           1733          MVI     M,ST.REL        ASSUME +                /80.09.BB/
050.113  376 053           1734          CPI     '+'             WAS IT?                 /80.09.BB/
050.115  310               1735          RZ                      YES                     /80.09.BB/
050.116  066 000           1736          MVI     M,0             NO, MUST BE -           /80.09.BB/
050.120  311               1737          RET                                             /80.09.BB/
```

```
                                1741  **      XTEXT - PROCESS XTEXT PSEUDO.                          /WCZ062680/
                                1742  *
                                1743  *       XTEXT   NAME
                                1744  *
                                1745
050.121  072 206 072           1746  XTEXT   LDA     XTXFLG
050.124  247                   1747          ANA     A
050.125  302 363 044           1748          JNZ     ERR,0,          ALREADY IN XTEXT
                                1749
050.130  041 372 072           1750          LXI     H,XTXFB+FB.NAM                    → doesn't check for error ('C'
050.133  315 207 064           1751          CALL    $CPF            COPY FILE NAME
                                1752
050.136  315 164 064           1753          CALL    $MOVEL          SET DEFAULT EXTENTION
050.141  003 000 020           1754          DW      XTEXTG,XTEXTD,XTEXTA+3
                                1755
                                1756  *       IF A DEVICE NAME IS SPECIFIED, THEN USE IT AND ONLY IT.
                                1757  *       OTHERWISE USE DEFAULT DEVICES.
                                1758
050.147  001 173 073           1759          LXI     B,XTEXTB
050.152  021 043 051           1760          LXI     D,XTEXTI
050.155  041 372 072           1761          LXI     H,XTXFB+FB.NAM
050.160  377 053               1762          DB      SYSCALL,.DECODE USE DECODE TO GET DEVICE NAME
050.162  322 201 050           1763          JNC     XTEXTOD         NO ERROR -- DEVICE MUST HAVE BEEN SPECIFIED
050.165  376 015               1764          CPI     EC.UND          CHECK THAT ERROR WAS UNKNOWN DEVICE
050.167  312 231 050           1765          JZ      XTEXT1          BR IF YES -- NO DEVICE WAS SPECIFIED
050.172  315 026 062           1766          CALL    SEF             BAD NEWS -- SOMETHINGELSE IS WRONG
050.175  010                   1767          DB      ERR.A
050.176  303 367 044           1768          JMP     ASM11
                                1769
050.201                        1770  XTEXTOD EQU     *
050.201  315 164 064           1771          CALL    $MOVEL          MOVE NO DEVICE NAME IN TO DEFAULT BLOCK
050.204  003 000 043           1772          DW      XTEXTG,XTEXTI,XTEXTA
050.212  021 007 051           1773          LXI     D,XTEXTA
050.215  041 360 072           1774          LXI     H,XTXFB
050.220  315 362 064           1775          CALL    $FOPER,         TRY TO OPEN FILE
050.223  322 377 050           1776          JNC     XTEXT8          GOT IT
050.226  303 370 050           1777          JMP     XTEXT7          DIDN'T GET IT
                                1778
                                1779  *       USE DEFAULT DEVICES.
                                1780  *       STEP 1 -- IF DEFAULT DEVICES WERE SPECIFIED ON THE COMMAND
                                1781  *                 LINE, THEN SEARCH THOSE DEVICES FOR THE FILE.
                                1782  *       STEP 2 -- IF NOT FOUND, THEN USE SOURCE DEVICE AS DEFAULT
                                1783  *                 DEVICE.
                                1784  *       STEP 3 -- IF STILL NOT FOUND, THEN TRY USING SY0: AS
                                1785  *                 DEFAULT DEVICE.
                                1786  *       STEP 4 -- GIVE UP.
                                1787
                                1788  *       TRY DEFAULT DEVICES GIVEN ON COMMAND LINE.
                                1789
050.231                        1790  XTEXT1  EQU     *
050.231  072 023 051           1791          LDA     XTEXTE
050.234  247                   1792          ANA     A
050.235  312 306 050           1793          JZ      XTEXT3          NO DEVICES GIVEN ON COMMAND LINE
                                1794
050.240  041 024 051           1795          LXI     H,XTEXTH        BEGINNING OF DEFAULT LIST
                                1796
```

```
050.243                    1797  XTEXT1D EQU     *
050.243  365               1798          PUSH    PSW             SAVE COUNT
050.244  345               1799          PUSH    H               SAVE ADDR OF CURRENT DEFAULT
050.245  021 007 051       1800          LXI     D,XTEXTA
050.250  353               1801          XCHG
050.251  001 003 000       1802          LXI     B,XTEXTG
050.254  315 252 030       1803          CALL    $MOVE           MOVE DEFAULT TO BLOCK
050.257  021 007 051       1804          LXI     D,XTEXTA
050.262  041 360 072       1805          LXI     H,XTXFB
050.265  315 362 064       1806          CALL    $FOPER.         TRY TO OPEN
050.270  341               1807          POP     H
050.271  301               1808          POP     B
050.272  322 377 050       1809          JNC     XTEXT8          GOT IT
050.275  021 003 000       1810          LXI     D,XTEXTG
050.300  031               1811          DAD     D               BUMP DEFAULT TABLE POINTER
050.301  170               1812          MOV     A,B
050.302  075               1813          DCR     A
050.303  302 243 050       1814          JNZ     XTEXT1D         GO THROUGH TABLE
                           1815
                           1816  *       GET CURRENT DEVICE
                           1817
050.306                    1818  XTEXT3  EQU     *
050.306  021 007 051       1819          LXI     D,XTEXTA
050.311  041 173 073       1820          LXI     H,XTEXTB
050.314  076 002           1821          MVI     A,CN.SOU
050.316  377 054           1822          DB      SYSCALL,.NAME   GET NAME OF INPUT FILE
050.320  322 332 050       1823          JNC     XTEXT3D         NO ERROR
050.323  315 026 062       1824          CALL    SEF
050.326  010               1825          DB      ERR.A
050.327  303 367 044       1826          JMP     ASM11
                           1827
050.332  315 164 064       1828  XTEXT3D CALL    $MOVEL          RESET DEFAULT EXTENSION
050.335  003 000 020       1829          DW      XTEXTG,XTEXTD,XTEXTA+3
050.343  021 007 051       1830          LXI     D,XTEXTA        (DE) = DEFAULT BLOCK ADDRESS
050.346  041 360 072       1831          LXI     H,XTXFB
050.351  315 362 064       1832          CALL    $FOPER.         OPEN WITH DEVICE AS DEFAULT
050.354  322 377 050       1833          JNC     XTEXT8          GOT IT
                           1834
                           1835  *       CANT OPEN ON THAT DEVICE, TRY SY0:
                           1836
050.357  021 015 051       1837          LXI     D,XTEXTC
050.362  315 362 064       1838          CALL    $FOPER.
050.365  322 377 050       1839          JNC     XTEXT8          GOT IT
                           1840
                           1841  *       CANT FIND IT ANYWHERE !
                           1842
050.370                    1843  XTEXT7  EQU     *
050.370  315 026 062       1844          CALL    SEF
050.373  001               1845          DB      ERR.U
050.374  303 367 044       1846          JMP     ASM11           LIST WITH ORG
                           1847
                           1848  *       GOT IT OPEN
                           1849
050.377  076 001           1850  XTEXT8  MVI     A,1
051.001  062 206 072       1851          STA     XTXFLG
051.004  303 367 044       1852          JMP     ASM11           LIST WITH ORG
```

```
                        1853
051.007                 1854  XTEXTA  DS      6               DEFAULT BLOCK FOR FIRST TRY TO OPEN
051.015  123 131 060    1855  XTEXTC  DB      'SY0'           DEFAULT BLOCK FOR 2ND TRY
051.020  101 103 115    1856  XTEXTD  DB      'ACM'           EXTENSION FOR ANY FETCH
051.023  000            1857  XTEXTE  DB      0       NUMBER OF DEVICES FROM COMMAND LINE
000.005                 1858  XTEXTF  EQU     5               MAXIMUM NUMBER OF DEVICES
000.003                 1859  XTEXTG  EQU     3               LENGTH OF DEVICE NAME
051.024                 1860  XTEXTH  DS      XTEXTF*XTEXTG   DEVICE TABLE
051.043  000 000 000    1861  XTEXTI  DB      0,0,0,0,0,0     NO DEFAULTS                     /WCZ062680/
```

```
                                1865  **      NOREF - PROCESS NOREF PSEUDO.                                   /WCZ0062680/
                                1866  *
                                1867  *       NOREF   SYMBOL1,...,SYMBOLN
                                1868  *
                                1869  *       SET NOREF BIT IN SYMBOL TABLE FOR THE REQUESTED SYMBOLS
                                1870  *       AND XREF ENTRY TO INDICATE NOREF OF SYMBOL IS GENERATED
                                1871  *
                                1872
051.051                         1873  NOREF   EQU     *
051.051  072 164 072            1874          LDA     PASS
051.054  075                   1875          DCR     A
051.055  312 010 045            1876          JZ      ASM14           SKIP IF PASS 1
                                1877
051.060                         1878  NOREF1  EQU     *
051.060  315 164 055            1879          CALL    CEF             Q, EOL
051.063  312 205 051            1880          JZ      NOREF8            BR IF YES
051.066  315 167 053            1881          CALL    LCT             LOOKUP CHARACTER
051.071  007                   1882          RLC
051.072  322 210 051            1883          JNC     NOREF9          BR IF NOT A LETTER
                                1884
                                1885  *       HAVE SYMBOL.  BUILD IT UP.
                                1886
051.075  041 217 051            1887          LXI     H,NOREFA        (HL)=WORKAREA
051.100  006 007                1888          MVI     B,7             MAX # OF CHARACTERS
051.102  345                   1889          PUSH    H               SAVE HL
051.103  053                   1890          DCX     H
051.104                         1891  NOREF4  EQU     *
051.104  315 167 053            1892          CALL    LCT             LOOKUP CHARACTER TYPE
051.107  346 300                1893          ANI     300Q
051.111  312 124 051            1894          JZ      NOREF5          BR IF NOT ALPHANUMERIC
051.114  032                   1895          LDAX    D
051.115  043                   1896          INX     H
051.116  167                   1897          MOV     M,A
051.117  023                   1898          INX     D
051.120  005                   1899          DCR     B
051.121  302 104 051            1900          JNZ     NOREF4
                                1901
                                1902  *       HAVE SYMBOL.  SEE IF IT IS IN THE SYMBOL TABLE AND IS DEFINED.
                                1903  *       IF IT IS, THEN SET NOREF BIT AND GENERATE XREF ENTRY.
                                1904
051.124                         1905  NOREF5  EQU     *
051.124  176                   1906          MOV     A,M
051.125  366 200                1907          ORI     200Q            SET SIGN ON LAST CHARACTER
051.127  167                   1908          MOV     M,A
                                1909
051.130  353                   1910          XCHG
051.131  343                   1911          XTHL                    SAVE DE, (HL)=NOREFA
051.132  353                   1912          XCHG
051.133  315 041 062            1913          CALL    SST             SEARCH SYMBOL TABLE
051.136  321                   1914          POP     D
051.137  176                   1915          MOV     A,M
051.140  247                   1916          ANA     A
000.000                         1917          ERRNZ   ST.UND
051.141  312 162 051            1918          JZ      NOREF6          BR IF SYMBOL UNDEFINED
051.144  366 010                1919          ORI     ST.NRF          SET NO XREF BIT
051.146  167                   1920          MOV     M,A
```

```
                                1921
051.147  041 217 051           1922            LXI     H,NOREFA
051.152  076 004               1923            MVI     A,XT.NRF
051.154  315 221 057           1924            CALL    ESR.            GENERATE XREF ENTRY
                                1925
051.157  303 176 051           1926            JMP     NOREF7
                                1927
                                1928  *         FLAG UNDEFINED SYMBOL ERROR AND SHOW SYMBOL WAS REFERENCED
                                1929  *         FOR 'XREF' ENTRY.
                                1930
051.162                         1931  NOREF6    EQU     *
051.162  315 026 062           1932            CALL    SEF
051.165  001                   1933            DB      ERR.U
                                1934
051.166  041 217 051           1935            LXI     H,NOREFA
051.171  076 000               1936            MVI     A,XT.REF
051.173  315 221 057           1937            CALL    ESR.            GENERATE XREF ENTRY
000.000                         1938            ERRNZ   NOREF7-*
                                1939
                                1940  *         CHECK IF MORE POSSIBLE SYMBOLS.
                                1941
051.176                         1942  NOREF7    EQU     *
051.176  032                   1943            LDAX    D
051.177  023                   1944            INX     D
051.200  376 054               1945            CPI     ','             0. DELIMITER MUST BE ','
051.202  312 060 051           1946            JZ      NOREF1          BR IF IT IS
                                1947
                                1948  *         ALL DONE.
                                1949
051.205                         1950  NOREF8    EQU     *
051.205  303 344 044           1951            JMP     ASM10           REQUIRE EOL AND LIST W/O ORG
                                1952
                                1953  *         ERROR.
                                1954
051.210                         1955  NOREF9    EQU     *
051.210  315 026 062           1956            CALL    SEF             FLAG EXPRESSION ERROR
051.213  010                   1957            DB      ERR.A
051.214  303 001 045           1958            JMP     ASM13
                                1959
051.217                         1960  NOREFA    DS      7               WORKAREA FOR BUILDING SYMBOL    /WCZ062480/
```

```
                                1964  **      SNG - SINGLE BYTE, NO OPERAND.
                                1965  *
                                1966
                                1967
        044.336                 1968  SNG     EQU     ASM8            GENERATE 1 BYTE


                                1970  **      IMM - IMMEDIATE ARITHMETIC.
                                1971  *
                                1972  *       OPC     VAL
                                1973
                                1974
        051.226 315 054 057     1975  IMM     CALL    E8B             EVALUATE TO 8 BITS
        051.231 106             1976          MOV     B,M
        051.232 303 311 044     1977          JMP     ASM6            GENERATE 2 BYTES


                                1979  **      THR - THREE BYTE OPCODES.
                                1980  *
                                1981  *       OPC     EXPR
                                1982  *
                                1983  *       JMP, CALL, LHLD, SHLD, LDA, STA
                                1984
                                1985
        051.235 315 355 053     1986  THR     CALL    EVL
        051.240 126             1987          MOV     D,M
        051.241 303 321 044     1988          JMP     ASM7            GENERATE 3 BYTES


                                1990  **      RA0 - REGISTER ARITHEMETIC, TYPE 1.
                                1991  *
                                1992  *       REGISTER SPECIFIED IN LOW 3 BITS.
                                1993
                                1994
        051.244 315 331 056     1995  RA0     CALL    DRS             DECODE REGISTER SPECIFICATIOIN
        051.247 011 057         1996          DW      DRSA            GROUP 1                         /80.09.BB/
        051.251 303 335 044     1997          JMP     ASM8.           GENERATE 1 BYTE


                                1999  **      RAT - REGISTER ARITHEMETIC, TYPE 2
                                2000  *
                                2001  *       REGISTER SPECIFICATIOIN IN MID 3 BITS
                                2002
                                2003
        051.254 315 331 056     2004  RAT     CALL    DRS             DECODE REGISTER SPECIFICATIOIN
        051.257 011 057         2005          DW      DRSA            GROUP 1                         /80.09.BB/
        051.261 007             2006          RLC
        051.262 007             2007          RLC
```

```
051.263  007           2008          RLC
051.264  303 335 044   2009          JMP     ASM8.           GENERATE 1 BYTE


                       2011  **      RPO - REGISTER PAIR, GROUP 1
                       2012  *
                       2013  *       B =1, D=0, H=5, S=7
                       2014
                       2015
051.267  315 331 056   2016  RPO     CALL    DRS             DECODE REGISTER SPECIFICATION
051.272  032 057       2017          DW      DRSB            GROUP 2                         /80.09.BB/
051.274  303 335 044   2018          JMP     ASM8.           GENERATE 1 BYTE


                       2020  **      RPT - REGISTER PAIR GROUP 2
                       2021  *
                       2022  *       PUSH, POP
                       2023  *
                       2024  *       B=0, D=2, H=4, P=6
                       2025
                       2026
051.277  315 331 056   2027  RPT     CALL    DRS             DECODE REGISTER SPECIFICATION
051.302  043 057       2028          DW      DRSC            GROUP 3                         /80.09.BB/
051.304  303 335 044   2029          JMP     ASM8.           GENERATE 1 BYTE


                       2031  **      INX - PROCESS INX INSTRUCTION.
                       2032  *
                       2033
                       2034
051.307  315 315 051   2035  INX     CALL    INX1            DECODE REGISTER SPECIFICATION
051.312  303 335 044   2036          JMP     ASM8.           RETURN WITH CODE
                       2037
                       2038
                       2039  **      INX1 - B=00, D=20, H=40, S=60
                       2040
051.315  315 331 056   2041  INX1    CALL    DRS
051.320  032 057       2042          DW      DRSB            GROUP 2                         /80.09.BB/
051.322  075           2043          DCR     A
051.323  027           2044          RAL
051.324  027           2045          RAL
051.325  027           2046          RAL
051.326  346 070       2047          ANI     070Q
051.330  311           2048          RET
```

```
                        2050  **      MVI - PROCESS MVI INSTRUCTIIN.
                        2051  *
                        2052  *       MVI     REG,VAL
                        2053
                        2054
051.331  315 331 056   2055  MVI     CALL    DRS             DECODE REG SPEC
051.334  011 057       2056          DW      DRSA            GROUP 1                         /80.09.BB/
051.336  007           2057          RLC
051.337  007           2058          RLC
051.340  007           2059          RLC
051.341  200           2060          ADD     B
051.342  147           2061          MOV     H,A             (H) = OPCODE BYTE
051.343  315 221 055   2062          CALL    CMA             REQUIRE COMA
051.346  315 054 057   2063          CALL    E8B             EVALUTE TO 8 BITS
051.351  104           2064          MOV     B,H
051.352  303 311 044   2065          JMP     ASM6            OUTPUT 2 BYTES


                        2067  **      INDX - PROCESS LDAX, STAX INSTRUCTIONS
                        2068  *
                        2069  *       LDAX    B
                        2070  *       LDAX    D
                        2071  *       STAX    B
                        2072  *       STAX    D
                        2073
                        2074
051.355  032           2075  INDX    LDAX    D
051.356  376 102       2076          CPI     'B'
051.360  312 376 051   2077          JE      INDX1           IS 'B'
051.363  376 104       2078          CPI     'D'
051.365  076 020       2079          MVI     A,20Q
051.367  312 335 044   2080          JE      ASM8,           OUTPUT 1 BYTE
051.372  315 026 062   2081          CALL    SEF             ERROR
051.375  002           2082          DB      ERR.R           BAD RESIGER SPECIFIED
051.376  170           2083  INDX1   MOV     A,B
051.377  303 336 044   2084          JMP     ASM8            OUTPUT 1 BYTE


                        2086  **      RST - RESTART INSTRUCTIOIN
                        2087  *
                        2088  *       RST     EXPR
                        2089  *
                        2090  *       EXPR MUST BE 0-7
                        2091
                        2092
052.002  315 054 057   2093  RST     CALL    E8B             EVALUATE TO 8 BITS
052.005  171           2094          MOV     A,C
052.006  346 370       2095          ANI     370Q
052.010  312 017 052   2096          JZ      RST1            IF OK
052.013  315 026 062   2097          CALL    SEF
052.016  020           2098          DB      ERR.V
052.017  171           2099  RST1    MOV     A,C
```

```
052.020  007            2100           RLC
052.021  007            2101           RLC
052.022  007            2102           RLC
052.023  206            2103           ADD     M
052.024  303 336 044    2104           JMP     ASM8            OUTPUT 1


                        2106  **       LXI - PROCESS LXI INSTRUCTION.
                        2107  *
                        2108  *        LXI     REG,EXPR
                        2109
                        2110
052.027  315 315 051    2111  LXI      CALL    INX1            DECODE SPECIFICATION
052.032  200            2112           ADD     B
052.033  147            2113           MOV     H,A             (H) = OPCODE
052.034  315 221 055    2114           CALL    CMA             GOBBLE COMMA
052.037  315 355 053    2115           CALL    EVL             EVALUATE EXPRESSION
052.042  124            2116           MOV     D,H             (D) = 3RD BYTE
052.043  303 321 044    2117           JMP     ASM7            OUTPUT 3 BYTES


                        2119  **       MOV - PROCESS MOV INSTRUCTION.
                        2120  *
                        2121  *        MOV     REG,REG
                        2122
                        2123
052.046  315 331 056    2124  MOV      CALL    DRS
052.051  011 057        2125           DW      DRSA            GROUP 1                         /80.09.BB/
052.053  007            2126           RLC
052.054  007            2127           RLC
052.055  007            2128           RLC
052.056  260            2129           ORA     B
052.057  107            2130           MOV     B,A             (B) = OPCODE AND 1ST REG
052.060  315 221 055    2131           CALL    CMA             READ ,
052.063  315 331 056    2132           CALL    DRS
052.066  011 057        2133           DW      DRSA            GROUP 1                         /80.09.BB/
052.070  200            2134           ADD     B
052.071  303 336 044    2135           JMP     ASM8            SINGLE BYTE
```

```
                                2139  **      DNT - DECODE NEXT TOKEN.
                                2140  *
                                2141  *       DNT IS CALLED TO DECODE THE NEXT TOKEN.
                                2142  *
                                2143  *       IF TOKEN = OPERATOR, (L) = INDEX
                                2144  *               =0 +
                                2145  *               =1 -
                                2146  *               =2 *
                                2147  *               =3 /
                                2148  *
                                2149  *       IF TOKEN = SYMBOL, (BC) = VALUE
                                2150  *
                                2151  *       DNT EXITS THROUGH A BRANCH TABLE.
                                2152  *
                                2153  *       CALL    DNT
                                2154  *       DB      ADRA-*          IF +
                                2155  *       DB      ADRB-*          IF -
                                2156  *       DB      ADRC-*          IF *
                                2157  *       DB      ADRD-*          IF /
                                2158  *       DB      ADRE-*          IF SYMBOL
                                2159  *       DB      ADRF-*          IF END OF EXPR
                                2160  *
                                2161  *       ENTRY   (DE) = EXPRESSION POINTER
                                2162  *       EXIT    (BC) = VALUE IF SYMBOL
                                2163  *               (L) = INDEX IF OPERATOR
                                2164  *               TOKREL = ST.REL IF RELOCATABLE VALUE
                                2165  *       USES    ALL
                                2166
                                2167
052.074                         2168  DNT     EQU     *
052.074  041 076 031           2169          LXI     H,$TBRA
052.077  345                   2170          PUSH    H               SET $TBRA EXIT VIA *RET*
052.100  257                   2171          XRA     A
052.101  062 053 073           2172          STA     TOKREL          CLEAR RELOCATION FLAG
052.104  032                   2173          LDAX    D
052.105  376 047               2174          CPI     QUOTE
052.107  302 145 052           2175          JNE     DNT2            NOT QUOTE
                                2176
                                2177  *       HAVE 'C' OR 'CC'
                                2178
052.112  023                   2179          INX     D
052.113  315 255 060           2180          CALL    GSC             GET STRING CHARACTER
052.116  312 113 053           2181          JZ      DNT13           NULL STRING ILLEGAL
052.121  117                   2182          MOV     C,A
052.122  006 000               2183          MVI     B,0             ASSUME ONE CHARACTER
052.124  315 255 060           2184          CALL    GSC             GET STRING CHARACTER
052.127  312 142 052           2185          JZ      DNT1            ONLY 1 CHARACTER
052.132  101                   2186          MOV     B,C
052.133  117                   2187          MOV     C,A
052.134  315 255 060           2188          CALL    GSC             GET STRING CHARACTER
052.137  302 113 053           2189          JNZ     DNT13           TOO MANY CHARACTERS
052.142  076 004               2190  DNT1    MVI     A,4
052.144  311                   2191          RET                     RETURN VIA $TBRA
                                2192
                                2193  *       HAVE OPERATOR OR SYMBOL OR NULL
                                2194
```

```
052.145  315 164 055  2195  DNT2    CALL    CEF             CHECK FOR END OF FIELD
052.150  076 005      2196          MVI     A,5
052.152  310          2197          RZ                      IF END OF FIELD
052.153  032          2198          LDAX    D
052.154  376 054      2199          CPI     ','
052.156  076 005      2200          MVI     A,5
052.160  310          2201          RE                      IF ',', FLAG AS END OF EXPRESSION
052.161  315 167 053  2202          CALL    LCT             LOCKUP CHARACTER
052.164  157          2203          MOV     L,A
052.165  007          2204          RLC
052.166  332 207 052  2205          JC      DNT3            IS SYMBOL
052.171  007          2206          RLC
052.172  332 323 052  2207          JC      DNT6            IS NUMBER
052.175  007          2208          RLC
052.176  322 113 053  2209          JNC     DNT13           ERROR
                      2210
                      2211  *       HAVE OPERATOR
                      2212
052.201  175          2213          MOV     A,L
052.202  346 003      2214          ANI     3
052.204  157          2215          MOV     L,A
052.205  023          2216          INX     D
052.206  311          2217          RET                     EXIT VIA $TBRA
                      2218
                      2219  *       HAVE SYMBOL. BUILD IT UP
                      2220
052.207  041 144 053  2221  DNT3    LXI     H,DNTA          (HL) = WORKAREA POINTER
052.212  006 007      2222          MVI     B,7             7 CHAR MAX
052.214  345          2223          PUSH    H
052.215  053          2224          DCX     H
052.216  315 167 053  2225  DNT4    CALL    LCT             LOOKUP CHARACTER TYPE
052.221  346 300      2226          ANI     300Q
052.223  312 236 052  2227          JZ      DNT5            NOT ALPHANUMERIC
052.226  032          2228          LDAX    D
052.227  043          2229          INX     H
052.230  167          2230          MOV     M,A
052.231  023          2231          INX     D
052.232  005          2232          DCR     B
052.233  302 216 052  2233          JNZ     DNT4            IF MORE TO COPY
                      2234
                      2235  *       HAVE SYMBOL. LOOKUP VALUE
                      2236
052.236  176          2237  DNT5    MOV     A,M             SET SIGN ON LAST CHARACTER
052.237  366 200      2238          ORI     200Q
052.241  167          2239          MOV     M,A
                      2240
052.242  345          2241          PUSH    H                                       /80.03.GC/
052.243  041 144 053  2242          LXI     H,DNTA          HL = ADDRESS OF SYMBOL  /80.03.GC/
052.246  076 000      2243          MVI     A,XT.REF                                /80.03.GC/
052.250  315 221 057  2244          CALL    ESR.                                    /80.03.GC/
052.253  341          2245          POP     H                                       /80.03.GC/
                      2246
052.254  353          2247          XCHG
052.255  343          2248          XTHL                    SAVE DE, (HL) = DNTA
052.256  353          2249          XCHG
052.257  315 041 062  2250          CALL    SST             SEARCH SYMBOL TABLE
```

```
052.262  176           2251           MOV     A,M             (A) = TYPE
052.263  043           2252           INX     H
000.000                2253           ERRNZ   ST.UND          CODE ASSUMES = 0
052.264  247           2254           ANA     A
052.265  302 275 052    2255           JNZ     DNT5.5          DEFINED
052.270  315 026 062    2256           CALL    SEF             *U* ERROR
052.273  001           2257           DB      ERR.U
052.274  257           2258           XRA     A               CLEAR FLAG
052.275  365           2259  DNT5.5   PUSH    PSW             SAVE CODE
052.276  346 100       2260           ANI     ST.REL
052.300  062 053 073    2261           STA     TOKREL          SET RELOCATABLE FLAG
052.303  361           2262           POP     PSW             (A) = FLAG BITS
052.304  027           2263           RAL
052.305  322 314 052    2264           JNC     DNT5.7          NOT REFERENCE TO DOUBLE DEFINED
052.310  315 026 062    2265           CALL    SEF             FLAG *P* FOR DOUBLE REFERENCE
052.313  200           2266           DB      ERR.P
052.314                2267  DNT5.7   EQU     *
052.314  116           2268           MOV     C,M
052.315  043           2269           INX     H
052.316  106           2270           MOV     B,M             (BC) = VALUE
052.317  321           2271           POP     D               RESTORE (DE)
052.320  076 004       2272           MVI     A,4
052.322  311           2273           RET     $TBRA           EXIT VAI $TBRA
                       2274
                       2275  *        HAVE NUMBER
                       2276
052.323  041 143 053    2277  DNT6     LXI     H,DNTA-1
052.326  006 022       2278           MVI     B,18            18 DIGITS MAX
052.330  315 167 053    2279  DNT7     CALL    LCT             LOOKUP TYPE
052.333  346 120       2280           ANI     120Q            SEE IF NUMBER OR POSTRADIX
052.335  312 350 052    2281           JZ      DNT8            OUT OF NUMBER
052.340  032           2282           LDAX    D
052.341  043           2283           INX     H
052.342  167           2284           MOV     M,A             COPY TO WORK AREA
052.343  023           2285           INX     D
052.344  005           2286           DCR     B
052.345  302 330 052    2287           JNZ     DNT7
                       2288
                       2289  *        HAVE ACCUMULATED NUMBER, SEE IF HAS POSTRADIX.
                       2290
052.350                2291  DNT8     EQU     *
052.350  257           2292           XRA     A
052.351  062 072 053    2293           STA     DNTD            FLAG NO OVERFLOW
052.354  176           2294           MOV     A,M
052.355  062 073 053    2295           STA     DNTC            SAVE POSTRADIX
052.360  315 174 053    2296           CALL    LCT.            LOOKUP CHARACTER TYPE
052.363  346 020       2297           ANI     20Q
052.365  302 373 052    2298           JNZ     DNT9            HAS POSTRADIX
052.370  043           2299           INX     H
052.371  066 104       2300           MVI     M,'D'
052.372                2301  DNTB     EQU     *-1             DEFAULT POSTRADIX
                       2302
                       2303  *        COMPUTE BASE
                       2304
052.373  176           2305  DNT9     MOV     A,M             (A) = POSTRADIX
052.374  066 200       2306           MVI     M,200Q          FLAG END OF NUMBER
```

```
052.376  315 174 053  2307           CALL    LCT.            LOOPUP CHARACTER TYPE
053.001  346 017      2308           ANI     17Q
053.003  074          2309           INR     A               (A) = POSTRADIX
053.004  325          2310           PUSH    D               SAVE EXPRESSION POINTER
053.005  137          2311           MOV     E,A
053.006  026 000      2312           MVI     D,0             (DE) = BASE
                      2313
                      2314  *        DECODE NUMBER.
                      2315
053.010  041 144 053  2316           LXI     H,DNTA
053.013  001 000 000  2317           LXI     B,0             PRESET ACCUMULATOR TO 0
053.016  176          2318  DNT10    MOV     A,M
053.017  247          2319           ANA     A
053.020  372 070 053  2320           JM      DNT11           ALL DONE
053.023  315 310 055  2321           CALL    DHD             DECODE HEX DIGITS
053.026  332 112 053  2322           JC      DNT12           ERROR
053.031  043          2323           INX     H
053.032  345          2324           PUSH    H
053.033  325          2325           PUSH    D
053.034  365          2326           PUSH    PSW
053.035  315 337 030  2327           CALL    $MU66           ACCUM = ACCUM*BASE
053.040  345          2328           PUSH    H
053.041  041 072 053  2329           LXI     H,DNTD          ACCUMULATE OVERFLOW FLAGS
053.044  206          2330           ADD     M
053.045  167          2331           MOV     M,A
053.046  341          2332           POP     H
053.047  361          2333           POP     PSW
053.050  117          2334           MOV     C,A
053.051  006 000      2335           MVI     B,0             (BC) = DIGIT VALUE
053.053  011          2336           DAD     B               (HL) = ACCUM*BASE + DIGIT
053.054  321          2337           POP     D
053.055  171          2338           MOV     A,C
053.056  273          2339           CMP     E               COMPARE DIGIT TO BASE
053.057  104          2340           MOV     B,H
053.060  115          2341           MOV     C,L
053.061  341          2342           POP     H
053.062  322 112 053  2343           JNC     DNT12           ERROR
053.065  303 016 053  2344           JMP     DNT10
                      2345
053.070  321          2346  DNT11    POP     D               NUMBER ACCUMULATED OK
053.071  041 000 000  2347           LXI     H,0             (H) = POSTRADIX, (L) = OVERFLOW
053.072               2348  DNTD     EQU     $-2             OVERFLOW FLAG
053.073               2349  DNTC     EQU     $-1             POSTRADIX
053.074  076 101      2350           MVI     A,'A'
053.076  274          2351           CMP     H
053.077  314 122 053  2352           CE      DNT14           IS 'A' POSTRADIX
053.102  175          2353           MOV     A,L
053.103  247          2354           ANA     A
053.104  304 137 053  2355           CNZ     DNT15           IS OVERFLOW
                      2356
053.107  076 004      2357           MVI     A,4
053.111  311          2358           RET     $IBRA           EXIT VIA $IBRA
                      2359
053.112  321          2360  DNT12    POP     D               ERROR WHILE CRACKING NUMBER
                      2361
                      2362  *        ERROR DETECTED
```

```
                        2363
053.113 315 026 062     2364  DNT13   CALL    SEF             *A* ERROR
053.116 010             2365          DB      ERR.A
053.117 076 005         2366          MVI     A,5             SET TYPE = NULL
053.121 311             2367          RET     $TBRA           EXIT THROUGH $TBRA
                        2368
053.122 175             2369  DNT14   MOV     A,L
053.123 037             2370          RAR
053.124 170             2371          MOV     A,B             SHIFT HIGH BYTE RIGHT WITH CARRY
053.125 037             2372          RAR
053.126 107             2373          MOV     B,A
053.127 334 137 053     2374          CC      DNT15           BAD DIGIT
053.132 175             2375          MOV     A,L             (A) = OVERFLOW REGISTER
053.133 346 376         2376          ANI     376Q            CLEAR ALLOWED OVERFLOW
053.135 157             2377          MOV     L,A             CLEAR SINGLE CARRY
053.136 311             2378          RET
                        2379
053.137 315 026 062     2380  DNT15   CALL    SEF             FLAG OVERFLOW
053.142 020             2381          DB      ERR.V
053.143 311             2382          RET
                        2383
                        2384
053.144                 2385  DNTA    DS      19              WORK AREA


                        2387  **      LCT - LOOKUP CHARACTER TYPE.
                        2388  *
                        2389  *       LCT LOOKS UP THE CHARACTER TYPE INDEX FOR A CHARACTER.
                        2390  *
                        2391  *       ENTRY   (DE) = STRING POINTER
                        2392  *       EXIT    (A) = INDEX
                        2393  *                       1000    VALID ALPHA
                        2394  *                       0100    VALID NUMBER
                        2395  *                       0010    VALID OPERATOR
                        2396  *                       0001    VALID POSTRADIX
                        2397  *                           NNNN OPCODE INDEX IF OPERATOR, BASE IF POSTRADIX
                        2398  *       USES    A,F
                        2399  *
                        2400
000.000                 2401          ERRNZ   CT.ALPH-200Q                                    /80.02.GC/
                        2402
053.167 032             2403  LCT     LDAX    D
053.170 247             2404          ANA     A
053.171 372 213 053     2405          JM      LCT1
053.174                 2406  LCT.    EQU     *               ENTRY WITH (A) = CHARACTER
053.174 326 040         2407          SUI     ' '
053.176 332 213 053     2408          JC      LCT1            TOO SMALL
053.201 345             2409          PUSH    H               SAVE (HL)
053.202 041 215 053     2410          LXI     H,LCTA
053.205 315 101 030     2411          CALL    $DADA.
053.210 176             2412          MOV     A,M             (A) = FLAG BYTE
053.211 341             2413          POP     H
053.212 311             2414          RET
                        2415
```

```
053.213  257          2416  LCT1    XRA     A               END OF LINE
053.214  311          2417          RET
                      2418
053.215               2419  LCTA    EQU     *               CHARACTER TABLE
                      2420
053.215  000          2421          DB      00000000B       BLANK
053.216  000          2422          DB      00000000B       !
053.217  000          2423          DB      00000000B       "
053.220  000          2424          DB      00000000B       #
053.221  200          2425          DB      10000000B       $
053.222  000          2426          DB      00000000B       PERCENT
053.223  000          2427          DB      00000000B       &
053.224  000          2428          DB      00000000B       '
053.225  000          2429          DB      00000000B       (
053.226  000          2430          DB      00000000B       )
053.227  042          2431          DB      00100010B       *
053.230  040          2432          DB      00100000B       +
053.231  000          2433          DB      00000000B       ,
053.232  041          2434          DB      00100001B       -
053.233  200          2435          DB      10000000B       .
053.234  043          2436          DB      00100011B       /
053.235  100          2437          DB      01000000B       0
053.236  100          2438          DB      01000000B       1
053.237  100          2439          DB      01000000B       2
053.240  100          2440          DB      01000000B       3
053.241  100          2441          DB      01000000B       4
053.242  100          2442          DB      01000000B       5
053.243  100          2443          DB      01000000B       6
053.244  100          2444          DB      01000000B       7
053.245  100          2445          DB      01000000B       8
053.246  100          2446          DB      01000000B       9
053.247  000          2447          DB      00000000B       :
053.250  000          2448          DB      00000000B       ;
053.251  000          2449          DB      00000000B       <
053.252  000          2450          DB      00000000B       =
053.253  000          2451          DB      00000000B       >
053.254  000          2452          DB      00000000B       ?
053.255  000          2453          DB      00000000B       @
053.256  327          2454          DB      11010111B       A
053.257  321          2455          DB      11010001B       B
053.260  300          2456          DB      11000000B       C
053.261  331          2457          DB      11011001B       D
053.262  300          2458          DB      11000000B       E
053.263  300          2459          DB      11000000B       F
053.264  200          2460          DB      10000000B       G
053.265  237          2461          DB      10011111B       H
053.266  200          2462          DB      10000000B       I
053.267  200          2463          DB      10000000B       J
053.270  200          2464          DB      10000000B       K
053.271  200          2465          DB      10000000B       L
053.272  200          2466          DB      10000000B       M
053.273  200          2467          DB      10000000B       N
053.274  227          2468          DB      10010111B       O
053.275  200          2469          DB      10000000B       P
053.276  227          2470          DB      10010111B       Q
053.277  200          2471          DB      10000000B       R
```

```
053.300  200          2472          DB      10000000B       S
053.301  200          2473          DB      10000000B       T
053.302  200          2474          DB      10000000B       U
053.303  200          2475          DB      10000000B       V
053.304  200          2476          DB      10000000B       W
053.305  200          2477          DB      10000000B       X
053.306  200          2478          DB      10000000B       Y
053.307  200          2479          DB      10000000B       Z
053.310  000          2480          DB      00000000B       [
053.311  000          2481          DB      00000000B       \
053.312  000          2482          DB      00000000B       ]
053.313  000          2483          DB      00000000B       ^
053.314  000          2484          DB      00000000B       _
053.315  000          2485          DB      00000000B       `
053.316  327          2486          DB      11010111B       a
053.317  321          2487          DB      11010001B       b
053.320  300          2488          DB      11000000B       c
053.321  331          2489          DB      11011001B       d
053.322  300          2490          DB      11000000B       e
053.323  300          2491          DB      11000000B       f
053.324  200          2492          DB      10000000B       g
053.325  237          2493          DB      10011111B       h
053.326  200          2494          DB      10000000B       i
053.327  200          2495          DB      10000000B       j
053.330  200          2496          DB      10000000B       k
053.331  200          2497          DB      10000000B       l
053.332  200          2498          DB      10000000B       m
053.333  200          2499          DB      10000000B       n
053.334  227          2500          DB      10010111B       o
053.335  200          2501          DB      10000000B       p
053.336  227          2502          DB      10010111B       q
053.337  200          2503          DB      10000000B       r
053.340  200          2504          DB      10000000B       s
053.341  200          2505          DB      10000000B       t
053.342  200          2506          DB      10000000B       u
053.343  200          2507          DB      10000000B       v
053.344  200          2508          DB      10000000B       w
053.345  200          2509          DB      10000000B       x
053.346  200          2510          DB      10000000B       y
053.347  200          2511          DB      10000000B       z
053.350  000          2512          DB      00000000B       {
053.351  000          2513          DB      00000000B       |
053.352  000          2514          DB      00000000B
053.353  000          2515          DB      00000000B
053.354  000          2516          DB      00000000B       DEL


                      2518  **      EVL - EVALUATE OPERAND EXPRESSION.
                      2519  *
                      2520  *       EVL EVALUATES AN OPERAND EXPRESSION. IT IS PROCESSED
                      2521  *       LEFT TO RIGHT, WITH NO OPERATOR PRECIDENCE, AND NO PARANETHESIS.
                      2522  *
                      2523  *       VALID OPERATORS
                      2524  *               +
```

```
                                2525  *               -
                                2526  *               *
                                2527  *               /
                                2528  *
                                2529  *       VALID SYMBOLS
                                2530  *
                                2531  *       LABEL
                                2532  *       *       LOCATION COUNTER
                                2533  *       'C'     8 BIT ASCII
                                2534  *       'CC'    16 BIT ASCII
                                2535  *       NNN     NUMBER, POSTRADIX =
                                2536  *               Q OCTAL
                                2537  *               D DECIMAL
                                2538  *               B BINARY
                                2539  *               H HEX
                                2540  *
                                2541  *       IF PASS1, UNDEFINED ERROR FLAGS WILL BE IGNORED.
                                2542  *
                                2543  *       ENTRY   (DE) = STRING POINTER
                                2544  *       EXIT    (BC) = VALUE
                                2545  *               (DE) UPDATED
                                2546  *               EXPREL = ST.REL IF RELOCATABLE
                                2547  *               'C' SET IF ERROR
                                2548  *       USES    A,F,B,C,D,E
                                2549
                                2550
053.355  345                   2551  EVL     PUSH    H               SAVE (HL)
053.356  001 000 000           2552          LXI     B,0
053.361  257                   2553          XRA     A
053.362  062 052 073           2554          STA     EXPREL          CLEAR RELOCATABLE FLAG
053.365  305                   2555          PUSH    B               SAVE ACCUMULATOR ON STACK
053.366  032                   2556          LDAX    D
053.367  376 043               2557          CPI     '#'
053.371  076 377               2558          MVI     A,377Q          ASSUME NO #
053.373  302 000 054           2559          JNE     EVL1            NO #
053.376  074                   2560          INR     A               (A) = 0
053.377  023                   2561          INX     D               SKIP #
054.000  062 102 054           2562  EVL1    STA     EVLA            SET MASK FOR RESULT
                                2563
                                2564  *       HAVE NULL
                                2565
054.003  315 074 052           2566          CALL    DNT             DECODE NEXT TOKEN
054.006  032                   2567          DB      EVL5-*          + - UNARY +
054.007  031                   2568          DB      EVL5-*          - - UNARY -
054.010  004                   2569          DB      EVL2-*          * - ORG
054.011  061                   2570          DB      EVL8-*          / - ERROR
054.012  005                   2571          DB      EVL3-*          VAL - VALUE
054.013  057                   2572          DB      EVL8-*          NUL - ERROR
                                2573
054.014  315 216 054           2574  EVL2    CALL    EVL20           (BC) = (ORG)
054.017  341                   2575  EVL3    POP     H               DISCARD INITIAL VALUE
054.020  305                   2576          PUSH    B               SET INITIAL VALUE = ORG
054.021  072 053 073           2577          LDA     TOKREL
054.024  062 052 073           2578          STA     EXPREL          SET RELOCATABILITY OF EXPRESSION
                                2579
                                2580  *       HAVE VALUE.
```

```
                                2581
054.027  315 074 052  2582  EVL4    CALL    DNT             DECODE NEXT TOKEN
054.032  006          2583          DB      EVL5-*          +
054.033  005          2584          DB      EVL5-*          -
054.034  004          2585          DB      EVL5-*          *
054.035  003          2586          DB      EVL5-*          /
054.036  034          2587          DB      EVL8-*          VAL - ERROR
054.037  037          2588          DB      EVL9-*          NUL - DONE
                      2589
                      2590  *       HAVE OPERATOR
                      2591
054.040  345          2592  EVL5    PUSH    H               SAVE OPERATOR INDEX
054.041  315 074 052  2593          CALL    DNT             DECODE NEXT TOKEN
054.044  025          2594          DB      EVL7.5-*        + - ERROR
054.045  024          2595          DB      EVL7.5-*        - - ERROR
054.046  004          2596          DB      EVL6-*          * - ORG
054.047  022          2597          DB      EVL7.5-*        / - ERROR
054.050  005          2598          DB      EVL7-*          VAL - DO OPEARTION
054.051  020          2599          DB      EVL7.5-*        NUL - ERROR
                      2600
054.052  315 216 054  2601  EVL6    CALL    EVL20           (BC) = (ORG)
054.055  341          2602  EVL7    POP     H
054.056  175          2603          MOV     A,L             (A) = OPERATOR INDEX
054.057  341          2604          POP     H               (HL) = OLD VALUE, (BC) = NEW
054.060  325          2605          PUSH    D
054.061  315 105 054  2606          CALL    EVL10           PERFORM OPERATION
054.064  321          2607          POP     D
054.065  345          2608          PUSH    H               SAVE RESULT
054.066  303 027 054  2609          JMP     EVL4
                      2610
                      2611  *       ERROR
                      2612
054.071  361          2613  EVL7.5  POP     PSW             CLEAN STACK
054.072  315 026 062  2614  EVL8    CALL    SEF             SET ERROR FLAG
054.075  010          2615          DB      ERR.A
                      2616
                      2617  *       DONE
                      2618
054.076  301          2619  EVL9    POP     B               (BC) = VALUE
054.077  341          2620          POP     H               RESTORE HL
054.100  170          2621          MOV     A,B
054.101  346 000      2622          ANI     0               MASK OFF IF *
054.102               2623  EVLA    EQU     *-1
054.103  107          2624          MOV     B,A
054.104  311          2625          RET
                      2626
                      2627
                      2628  *       PERFORM ARITHEMETIC.
                      2629  *
                      2630  *       ENTRY   (L) = OPERATOR INDEX
                      2631  *               (BC) = Y
                      2632  *               (HL) = X
                      2633  *
                      2634  *       EXIT    (HL) = X OP Y
                      2635
054.105               2636  EVL10   EQU     *
```

```
054.105  315 076 031  2637          CALL    $TBRA
054.110  004          2638          DB      EVL11-*         +
054.111  022          2639          DB      EVL12-*         -
054.112  047          2640          DB      EVL13-*         *
054.113  055          2641          DB      EVL14-*         /
                      2642
054.114  011          2643  EVL11   DAD     B               +
054.115  345          2644          PUSH    H               SAVE SUM
054.116  041 052 073  2645          LXI     H,EXPREL
054.121  072 053 073  2646          LDA     TOKREL
054.124  206          2647          ADD     M               SUM RELOCATION FLAGS
000.000               2648          ERRNZ   ST.REL-1000
054.125  167          2649          MOV     M,A
054.126  341          2650          POP     H               RESTORE RESULT
054.127  372 211 054  2651          JM      EVL16           REL+REL IS ILLEGAL
054.132  311          2652          RET
                      2653
054.133  175          2654  EVL12   MOV     A,L             -
054.134  221          2655          SUB     C
054.135  157          2656          MOV     L,A
054.136  174          2657          MOV     A,H
054.137  230          2658          SBB     B
054.140  147          2659          MOV     H,A
054.141  345          2660          PUSH    H               SAVE RESULT
054.142  041 053 073  2661          LXI     H,TOKREL                                /80.09.BB/
054.145  072 052 073  2662          LDA     EXPREL                                  /80.09.BB/
054.150  226          2663          SUB     M
054.151  062 052 073  2664          STA     EXPREL          STORE RESULT            /80.09.BB/
054.154  341          2665          POP     H               RESTORE RESULT
054.155  332 211 054  2666          JC      EVL16           ABS-REL ILLEGAL
054.160  311          2667          RET
                      2668
054.161  353          2669  EVL13   XCHG                    *
054.162  315 337 030  2670          CALL    $MU66
054.165  303 177 054  2671          JMP     EVL15           CHECK FOR RELOCATION ERROR
                      2672
054.170  120          2673  EVL14   MOV     D,B             /
054.171  131          2674          MOV     E,C
054.172  104          2675          MOV     B,H
054.173  115          2676          MOV     C,L
054.174  315 106 030  2677          CALL    $DU66
054.177  345          2678  EVL15   PUSH    H               SAVE RESULT
054.200  041 052 073  2679          LXI     H,EXPREL
054.203  072 053 073  2680          LDA     TOKREL
054.206  267          2681          ORA     A
054.207  341          2682          POP     H               RESTORE RESULT
054.210  310          2683          RZ                      ABS 'OP' ABS IS OK
                      2684
                      2685  *       RELOCATION ERROR
                      2686
054.211  315 026 062  2687  EVL16   CALL    SEF
054.214  002          2688          DB      ERR.R
054.215  311          2689          RET
```

```
                                2691  **      EVL20 - USE ORG AS TOKEN VALUE
                                2692  *
                                2693  *       ENTRY   NONE
                                2694  *       EXIT    (BC) = ORG
                                2695  *               TOKREL SET PROPERLY
                                2696  *       USES    A,F,B,C,H,L
                                2697
                                2698
054.216  072 211 072           2699  EVL20   LDA     RELFLG
054.221  062 053 073           2700          STA     TOKREL          SET FLAG PROPERLY
054.224  052 176 072           2701          LHLD    ORG
054.227  104                   2702          MOV     B,H
054.230  115                   2703          MOV     C,L             (BC) = (ORG)
054.231  311                   2704          RET
```

```
                                2708  **      ABV - ACCUMULATE BYTE VALUE.
                                2709  *
                                2710  *       ABV ADDS A BYTE TO THE BINARY BUFFER.
                                2711  *
                                2712  *       THE ORG UPON ENTRY IS THE ADDRESS+1 OF THE BYTE
                                2713  *
                                2714  *       ENTRY   (A) = VALUE
                                2715  *       EXIT    NONE
                                2716  *       USES    NONE
                                2717
                                2718
054.232  315 054 031           2719  ABV     CALL    $SAVALL         SAVE REGS
054.235  107                   2720          MOV     B,A             (B) = VALUE
054.236  072 214 072           2721          LDA     BINFNAM
054.241  247                   2722          ANA     A
054.242  312 047 031           2723          JZ      $RSTALL         NO BINARY FILE
054.245  072 164 072           2724          LDA     PASS
054.250  376 002               2725          CPI     2
054.252  302 006 055           2726          JNE     ABV2            NOT PASS 2
054.255  305                   2727          PUSH    B               SAVE VALUE
054.256  052 241 072           2728          LHLD    ABSFWA
054.261  353                   2729          XCHG                    (DE) = ORG OF FIRST BINARY BYTE
054.262  052 176 072           2730          LHLD    ORG
054.265  053                   2731          DCX     H               (HL) = REAL ORG
054.266  175                   2732          MOV     A,L
054.267  223                   2733          SUB     E
054.270  157                   2734          MOV     L,A
054.271  174                   2735          MOV     A,H
054.272  232                   2736          SBB     D
054.273  147                   2737          MOV     H,A             (HL) = INDEX OF BYTE IN BINARY FILE
054.274  072 236 072           2738          LDA     BINSKW
054.277  315 101 030           2739          CALL    $DADA           (HL) = NUMBER OF BYTE IN BINARY FILE
054.302  072 235 072           2740          LDA     BINCSN
054.305  274                   2741          CMP     H
054.306  312 373 054           2742          JE      ABV1            CAN GO IN THIS SECTOR
                                2743
                                2744  *       WILL NOT GO IN THIS SECTOR, PUT THIS SECTOR BACK, GET
                                2745  *       THE PROPER ONE
                                2746
054.311  315 146 063           2747          CALL    WBB             WRITE BINARY BUFFER
054.314  174                   2748          MOV     A,H
054.315  062 235 072           2749          STA     BINCSN          SET CURRENT SECTOR NUMBER
                                2750
                                2751  *       NOW READ THE NEW SECTOR INTO THE BUFFER AREA.  IF IT DOES NOT
                                2752  *       YET EXIST, WRITE ENOUGH GARBAGE UNTIL IT DOES.
                                2753
054.320  345                   2754          PUSH    H
054.321  114                   2755          MOV     C,H             (C) = SECTOR NUMBER
054.322  006 000               2756          MVI     B,0
054.324  076 000               2757          MVI     A,CN.BIN
054.326  377 047               2758          DB      SYSCALL+.POSIT  POSITION TO WHERE WE WANT
054.330  322 350 054           2759          JNC     ABV0            GOT THERE
054.333  101                   2760          MOV     B,C             (B) = SECTORS TO WRITE
054.334  016 000               2761          MVI     C,0
054.336  021 000 020           2762          LXI     D,4096          POINT TO GARBAGE (MOSTLY 0, I THINK..)
054.341  076 000               2763          MVI     A,CN.BIN
```

```
054.343  377 005       2764          DB      SYSCALL,.WRITE  WRITE IT
054.345  332 357 063   2765          JC      BINERR          ERROR
054.350  001 000 001   2766  ABV0    LXI     B,256
054.353  021 336 073   2767          LXI     D,BINBFR
054.356  076 000       2768          MVI     A,CN.BIN
054.360  377 004       2769          DB      SYSCALL,.READ   READ IN NEW SECTOR
054.362  322 372 054   2770          JNC     ABV00           OK
054.365  376 001       2771          CPI     EC.EOF          OK IF SIMPLE EOF
054.367  302 357 063   2772          JNE     BINERR          ERROR
054.372  341           2773  ABV00   POP     H               (L) = INDEX FOR BYTE
                       2774
                       2775  *       BYTE WILL GO IN THIS SECTOR
                       2776
054.373  021 336 073   2777  ABV1    LXI     D,BINBFR
054.376  046 000       2778          MVI     H,0
055.000  031           2779          DAD     D               (HL) = ADDRESS IN BINBFR
055.001  361           2780          POP     PSW             (A) = VALUE
055.002  167           2781          MOV     M,A             SET
055.003  303 047 031   2782          JMP     $RSTALL         RESTORE AND EXIT
                       2783
                       2784  *       IS PASS 1, IF AN ABS FILE, KEEP TRACK OF THE SMALLEST
                       2785  *       AND LARGEST ADDRESS WHICH GOT DATA...
                       2786
055.006                2787  ABV2    EQU     *                                       /80.09.BB/
                       2788  *       LDA     FTFLAG                                  /80.09.BB/
                       2789  *       ANA     A                                       /80.09.BB/
                       2790  **      JNZ     $RSTALL         NOT ABS                 /80.09.BB/
055.006  052 241 072   2791          LHLD    ABSFWA
055.011  353           2792          XCHG
055.012  052 176 072   2793          LHLD    ORG             COMPARE OLD LOWEST TO NOW
055.015  053           2794          DCX     H               (HL) = ORG FOR THIS BYTE
055.016  175           2795          MOV     A,L
055.017  223           2796          SUB     E
055.020  174           2797          MOV     A,H
055.021  232           2798          SBB     D
055.022  322 030 055   2799          JNC     ABV3            NOT NEW LOW
055.025  042 241 072   2800          SHLD    ABSFWA          NEW LOW
055.030  353           2801  ABV3    XCHG
055.031  052 247 072   2802          LHLD    ABSLWA
055.034  353           2803          XCHG
055.035  173           2804          MOV     A,E             SEE IF NEW HIGH
055.036  225           2805          SUB     L
055.037  172           2806          MOV     A,D
055.040  234           2807          SBB     H
055.041  322 047 031   2808          JNC     $RSTALL         NOT NEW HIGH
055.044  042 247 072   2809          SHLD    ABSLWA          SET IT
055.047  303 047 031   2810          JMP     $RSTALL         RESTORE AND RETURN
```

```
                                2812  **      BDT - BUILD DYNAMIC TABLES.
                                2813  *
                                2814  *       BDT INITIALIZES THE SYMBOL TABLE AND THE RELOCTION TABLE.
                                2815  *
                                2816  *       THE SYMBOL TABLE STARTS AT THE FIRST AVAILABLE ADDRESS,
                                2817  *       AND CONTINUES UP TO THE END OF THE RELOCATION TABLE.
                                2818  *
                                2819  *       THE RELOCATION TABLE STARTS IN HIGH MEMORY, AND GOES DOWN.
                                2820  *
                                2821  *       THIS ROUTINE IS USED ONE TIME ONLY, BUT CANNOT BE INCLUDED
                                2822  *       WITH THE OVERLAID CODE, IN THAT THIS ROUTINE ZAPS THAT OVERLAID AREA.
                                2823  *
                                2824  *       ENTRY   NONE
                                2825  *       EXIT    'C' CLEAR IF OK
                                2826  *               'C' SET IF ERROR, ERROR MESSAGE PRINTED
                                2827  *       USES    ALL
                                2828
                                2829
055.052  052 320 040           2830  BDT     LHLD    S.SYSM          (HL) = FWA SYSTEM
055.055  072 213 072           2831          LDA     LARGE
055.060  247                   2832          ANA     A
055.061  302 074 055           2833          JNZ     BDT1            WILL USEE ALL WE CAN
055.064  353                   2834          XCHG
055.065  052 324 040           2835          LHLD    S.OMAX
055.070  315 224 030           2836          CALL    $CHL
055.073  031                   2837          DAD     D               (HL) = AMOUNT WHICH WILL NOT CAUSE OVERLAY SWAPING
055.074  021 364 377           2838  BDT1    LXI     D,-12
055.077  031                   2839          DAD     D
055.100  353                   2840          XCHG                    (DE) = LIMIT FOR REL TABLE
055.101  041 224 274           2841          LXI     H,-SYMTAB-256
055.104  031                   2842          DAD     D
055.105  322 154 055           2843          JNC     BDT4            NOT AT LEAST 256 BYTES
055.110  353                   2844          XCHG                    (HL) = REL LIMIT
055.111  042 266 072           2845          SHLD    RELLWA          SET LIMIT FOR REL TABLE
055.114  042 270 072           2846          SHLD    RELPTR          SET REL TABLE EMPTY
055.117  377 052               2847          DB      SYSCALL,.SETTF  REQUEST IT
055.121  322 132 055           2848          JNC     BDT2            OK
055.124  046 007               2849          MVI     H,BELL
055.126  377 057               2850          DB      SYSCALL,.ERROR          PROBLEMS
055.130  067                   2851          STC                     FLAG ERROR
055.131  311                   2852          RET
                                2853
055.132  052 262 072           2854  BDT2    LHLD    SYMFWA
055.135  353                   2855          XCHG
055.136  052 266 072           2856          LHLD    RELLWA
055.141  053                   2857  BDT3    DCX     H
055.142  066 000               2858          MVI     M,0             CLEAR TABLE AREA
055.144  315 216 030           2859          CALL    $CDEHL
055.147  302 141 055           2860          JNE     BDT3            MORE TO GO
055.152  247                   2861          ANA     A
055.153  311                   2862          RET                     RETURN WITH 'C' CLEAR
                                2863
                                2864  *       NOT AT LEAST 256 BYTES IN SYMTAB. FORCE /LARGE
                                2865
055.154  076 001               2866  BDT4    MVI     A,1
055.156  062 213 072           2867          STA     LARGE           SET LARGE
```

```
055.161  303 052 055  2868          JMP     BDT             TRY AGAIN


                      2870  **      CEF - CHECK FOR END OF FIELD CHARACTER.
                      2871  *
                      2872  *       CEF CHECKS A CHARACTER TO SEE IF IT IS A
                      2873  *
                      2874  *       00, BLANK, OR TAB
                      2875  *
                      2876  *       ENTRY   (A) = CHARACTER
                      2877  *       EXIT    'Z' SET IF 00, BLANK OR TAB
                      2878  *               'Z' CLEAR OTHERWISE
                      2879  *       USES    F
                      2880
                      2881
055.164  247         2882  CEF     ANA     A
055.165  310         2883          RZ                      00
055.166  376 011     2884          CPI     TAB
055.170  310         2885          RE                      TAB
055.171  376 040     2886          CPI     ' '
055.173  311         2887          RET                     RETURN WITH CODE


                      2889  **      CLE - CHECK LISTING ELIGIBILITY.
                      2890  *
                      2891  *       CLE IS CALLED TO SEE IF THE CURRENT LINE SHOULD BE LISTED TO
                      2892  *       THE OUTPUT FILE.
                      2893  *
                      2894  *       IF LST.L = FALSE, DONT LIST
                      2895  *       IF XTEXT LINE AND LST.C = FALSE, DONT LIST
                      2896  *
                      2897  *       ENTRY   NONE
                      2898  *       EXIT    'Z' CLEAR IFF TO LIST
                      2899  *       USES    A,F,H,L
                      2900
                      2901
055.174  041 172 072  2902  CLE     LXI     H,LSTCTL
055.177  176         2903          MOV     A,M
055.200  346 001     2904          ANI     LST.L
055.202  310         2905          RZ                      DONT LIST
055.203  072 207 072  2906          LDA     XTXLINE
055.206  247         2907          ANA     A
055.207  302 215 055  2908          JNZ     CLE1            IS XTEXT
055.212  366 001     2909          ORI     1               LIST
055.214  311         2910          RET
                      2911
055.215  176         2912  CLE1    MOV     A,M
055.216  346 004     2913          ANI     LST.C
055.220  311         2914          RET
```

```
                                2916  **      CMA - READ COMMA.
                                2917  *
                                2918  *       CMA IS CALLED WHEN A COMMA IS EXPECTED TO APPEAR IN THE
                                2919  *       EXPRESSION. IT IS CHECKED, AND ADVANCED OVER.
                                2920  *
                                2921  *       ENTRY   (DE) = LINE POINTER
                                2922  *       EXIT    (DE) ADVANCED
                                2923  *       USES    A;F;D;E
                                2924
                                2925
055.221  032                   2926  CMA     LDAX    D
055.222  023                   2927          INX     D
055.223  376 054               2928          CPI     ','
055.225  310                   2929          RE                      OK
055.226  315 026 062           2930          CALL    SEF             *A* ERROR
055.231  010                   2931          DB      ERR.A
055.232  311                   2932          RET


                                2934  **      COL - COUNT OUTPUT LINES.
                                2935  *
                                2936  *       COL IS CALLED TO COUNT AN OUTPUT LINE BEFORE IT IS WRITTEN.
                                2937  *
                                2938  *       ENTRY   NONE
                                2939  *       EXIT    NONE
                                2940  *       USES    A;F
                                2941
                                2942
055.233  345                   2943  COL     PUSH    H               SAVE (HL)
055.234  325                   2944          PUSH    D
055.235  305                   2945          PUSH    B               SAVE REGISTERS
                                2946
055.236  041 047 073           2947          LXI     H,EJEFLG
055.241  176                   2948          MOV     A,M             (A) = EJEFLG
055.242  247                   2949          ANA     A
055.243  066 000               2950          MVI     M,0             CLEAR FLAG
000.000                        2951          ERRNZ   LINCNT-EJEFLG-1
055.245  043                   2952          INX     H
055.246  176                   2953          MOV     A,M             (A) = LINCNT
055.247  302 256 055           2954          JNZ     COL1            EJEFLG <> 0
055.252  247                   2955          ANA     A                                       /78.10.GC/
055.253  302 263 055           2956          JNZ     COL3            NOT TIME YET            /78.10.GC/
                                2957
                                2958  *       FORCE NEW PAGE.
                                2959
055.256  345                   2960  COL1    PUSH    H                                       /78.10.GC/
055.257  315 357 057           2961          CALL    FNP             FORCE NEW PAGE          /78.10.GC/
055.262  341                   2962          POP     H
                                2963
055.263  065                   2964  COL3    DCR     M               COUNT LINE              /78.10.GC/
                                2965
055.264  301                   2966          POP     B
055.265  321                   2967          POP     D
055.266  341                   2968          POP     H
```

```
055.267  311            2969            RET


                        2971  **        CUS - COMPUTE UNUSED SPACE.
                        2972  *
                        2973  *         CUS COMPUTES THE FREE SPACE LEFT TO THE ASSEMBLER.
                        2974  *
                        2975  *         IF NOT ENOUGH IS FREE TO CONTINUE, CUS RETURNS A NEGATIVE VALUE.
                        2976  *
                        2977  *         ENTRY   NONE
                        2978  *         EXIT    (HL) = BYTES FREE
                        2979  *                 'C' SET IF NOT ENOUGH TO CONTINUE
                        2980  *         USES    A,F,H,L
                        2981
                        2982
055.270  325            2983  CUS       PUSH    D
055.271  052 264 072    2984            LHLD    SYMPTR
055.274  353            2985            XCHG
055.275  052 270 072    2986            LHLD    RELPTR
055.300  175            2987            MOV     A,L
055.301  223            2988            SUB     E
055.302  157            2989            MOV     L,A
055.303  174            2990            MOV     A,H
055.304  232            2991            SBB     D
055.305  147            2992            MOV     H,A             COMPUTE DIFFERENCE
055.306  321            2993            POP     D               RESTORE (DE)
055.307  311            2994            RET


                        2996  **        DHD - DECODE HEX DIGIT.
                        2997  *
                        2998  *         DHD DECODES AN ASCII CHARACTER INTO A 4 BIT VALUE.
                        2999  *
                        3000  *         ENTRY   (A) = CHARACTER
                        3001  *         EXIT    (A) = VALUE
                        3002  *                 'C' SET IF ERROR
                        3003  *         USES    A,F
                        3004
                        3005
055.310  326 060        3006  DHD       SUI     '0'
055.312  330            3007            RC                      ERROR
055.313  376 012        3008            CPI     10
055.315  332 331 055    3009            JC      DHD1            IS 0-9
055.320  326 021        3010            SUI     'A'-'0'
055.322  330            3011            RC                      ERROR
055.323  376 006        3012            CPI     6
055.325  077            3013            CMC
055.326  330            3014            RC                      NOT A-F
055.327  306 012        3015            ADI     10
055.331  247            3016  DHD1      ANA     A               CLEAR CARRY
055.332  311            3017            RET                     EXIT WITH VALUE
```

```
                                3019  **      DEF - DEFINE SYMBOL.
                                3020  *
                                3021  *       DEF IS CALLED TO DEFINE THE SYMBOL IN *SYMBOL*.
                                3022  *       THE SYMBOL IS DEFINED ABSOLUTE OR RELOCATABLE, ACCORDING TO
                                3023  *       (EXPREL)
                                3024  *
                                3025  *       ENTRY   (D) = NEW SYMBOL TYPE
                                3026  *               (E) = OLD TYPE, IF SYMBOL IS PRESNET, AND ITS TYPE
                                3027  *               IS NOT ST.UND OR (E), FLAG 'D' ERROR.
                                3028  *               (BC) = VALUE
                                3029  *               LABEL = LINE LABEL
                                3030  *       EXIT    TO RET
                                3031  *       USES    A,F,D,E,H,L
                                3032
                                3033
     055.333  072 155 073       3034  DEF     LDA     LABEL
     055.336  247               3035          ANA     A
     055.337  302 347 055       3036          JNZ     DEF0            IF LABEL EXISTS
     055.342  315 026 062       3037          CALL    SEF             MUST HAVE LABEL
     055.345  040               3038          DB      ERR.F           *F* ERROR
     055.346  311               3039          RET
                                3040
     055.347  072 052 073       3041  DEF0    LDA     EXPREL
     055.352  262               3042          ORA     D               SET RELOCATION FLAG, IF RELOCATABLE
     055.353  127               3043          MOV     D,A
     055.354  325               3044          PUSH    D
     055.355  021 155 073       3045          LXI     D,LABEL
     055.360  315 041 062       3046          CALL    SST             SEARCH SYMBOL TABLE
     055.363  321               3047          POP     D
     055.364  176               3048          MOV     A,M             (A) = SYMBOL TYPE
     055.365  247               3049          ANA     A
     000.000                    3050          ERRNZ   ST.UND          CODE ASSUMES = 0
     055.366  312 005 056       3051          JZ      DEF1            UNDEFINED
     055.371  273               3052          CMP     E
     055.372  312 005 056       3053          JE      DEF1            IS PROPER OLD TYPE
     055.375  366 200           3054          ORI     ST.DBL
     055.377  167               3055          MOV     M,A             FLAG DOUBLE DEFINITION
     056.000  315 026 062       3056          CALL    SEF
     056.003  004               3057          DB      ERR.D           *D* ERROR
     056.004  311               3058          RET                     DONT RE-DEFINE
                                3059
     056.005  162               3060  DEF1    MOV     M,D             SET TYPE
     056.006  043               3061          INX     H
     056.007  161               3062          MOV     M,C
     056.010  043               3063          INX     H
     056.011  160               3064          MOV     M,B             SET VALUE
     056.012  311               3065          RET
```

```
                                3067  **      DLH - DEFINE LABEL HERE.
                                3068  *
                                3069  *       DLH IS CALLED TO DEFINE A LABEL (IF ONE EXISTS) AT THE
                                3070  *       CURRENT ORG.
                                3071  *
                                3072  *       ENTRY   (LABEL) = LABEL STRING
                                3073  *       EXIT    (HL) = ORG VALUE
                                3074  *               'D' SET ON LABEL IF DOUBLY DEFINED
                                3075  *       USES    ALL
                                3076
                                3077
        056.013  072 155 073    3078  DLH     LDA     LABEL
        056.016  247            3079          ANA     A
        056.017  310            3080          RZ                      NO LABEL EXISTS
                                3081
        056.020  076 001        3082          MVI     A,XT.LAB                                        /80.03.GC/
        056.022  315 213 057    3083          CALL    ESR                                             /80.03.GC/
                                3084
        056.025  072 164 072    3085          LDA     PASS
        056.030  075            3086          DCR     A
        056.031  302 055 056    3087          JNZ     DLH1            NOT PASS 1                      /80.03.GC/
                                3088
        056.034  052 176 072    3089          LHLD    ORG             (HL) = LABEL'S VALUE
        056.037  021 000 001    3090          LXI     D,ST.LAB*256+0
        056.042  072 211 072    3091          LDA     RELFLG
        056.045  062 052 073    3092          STA     EXPREL          DEFINE SYMBOL REL IF GENERATING REL CODE
        056.050  104            3093          MOV     B,H
        056.051  115            3094          MOV     C,L             (BC) = VALUE
        056.052  303 333 055    3095          JMP     DEF             DEFINE SYMBOL HERE
                                3096
                                3097  *       Pass 2                                                  /80.03.GC/
                                3098
        056.055  021 155 073    3099  DLH1    LXI     D,LABEL                                         /80.03.GC/
        056.060  315 041 062    3100          CALL    SST                                             /80.03.GC/
        056.063  176            3101          MOV     A,M             A = SYMBOL TYPE                 /80.03.GC/
        056.064  346 200        3102          ANI     ST.DBL          CHECK FOR DOUBLE DEFINITION     /80.03.GC/
        056.066  310            3103          RZ                      NOT DOUBLY DEFINED              /80.03.GC/
                                3104
        056.067  315 026 062    3105          CALL    SEF                                             /80.03.GC/
        056.072  004            3106          DB      ERR.D                                           /80.03.GC/
        056.073  311            3107          RET                                                     /80.03.GC/


                                3109  **      DLL - DISPLAY LISTING LINE.
                                3110  *
                                3111  *       DLL TYPES THE LISTING LINE IF THE 'L' LIST OPTION IS SET,
                                3112  *       OR IF AN ERROR IS PRESENT.
                                3113  *
                                3114  *       ENTRY   NONE
                                3115  *       EXIT    NONE
                                3116  *       USES    ALL
                                3117
                                3118
        056.074  072 164 072    3119  DLL     LDA     PASS
```

```
056.077  037              3120            RAR
056.100  330              3121            RC                      DO NOTHING PASS 1
                           3122
                           3123  *         SET XTEXT FLAG
                           3124
056.101  072 207 072       3125            LDA     XTXLINE                                 /80.02.GC/
056.104  247               3126            ANA     A                                       /80.02.GC/
056.105  312 115 056       3127            JZ      DLL0            IS NOT CURRENTLY AN *XTEXT*     /80.02.GC/
                           3128
056.110  076 130           3129            MVI     A,'X'                                   /80.02.GC/
056.112  062 155 072       3130            STA     DSPLNE          FLAG THE XTEXT          /80.02.GC/
                           3131
056.115  072 202 072       3132  DLL0      LDA     ERRFLG
056.120  107               3133            MOV     B,A
056.121  247               3134            ANA     A
056.122  302 134 056       3135            JNZ     DLL1            HAVE ERROR
056.125  315 174 055       3136            CALL    CLE             CHECK LISTING ELIGIBILITY
056.130  310               3137            RZ                      NOT TO LIST
056.131  303 234 056       3138            JMP     DLL3            DONT TRY TO INSERT ERROR MESSAGES
                           3139
                           3140  *         GENERATE ERROR CHARACTERS FOR FLAGGED ERRORS.
                           3141
056.134  315 026 064       3142  DLL1      CALL    $CCO            CLEAR CONTROL-O
056.137  016 003           3143            MVI     C,3             (C) = MAX NUMBER OF ERROR MESSAGES
056.141  052 165 072       3144            LHLD    ERRCNT
056.144  043               3145            INX     H
056.145  042 165 072       3146            SHLD    ERRCNT          COUNT LINES IN ERROR
056.150  041 120 072       3147            LXI     H,DSPLIN
056.153  021 307 056       3148            LXI     D,DLLB-1        (DE) = TABLE POINTER
                           3149
056.156  023               3150  DLL2      INX     D               LOOK UP ERROR CHARACTERS
056.157  032               3151            LDAX    D
056.160  023               3152            INX     D
056.161  247               3153            ANA     A
056.162  312 200 056       3154            JZ      DLL2.5          ALL MESSAGES TYPED
056.165  240               3155            ANA     B
056.166  312 156 056       3156            JZ      DLL2            NO ERROR OF THIS TYPE
056.171  032               3157            LDAX    D               (A) = CHARACTER
056.172  167               3158            MOV     M,A             STORE IN LINE
056.173  043               3159            INX     H
056.174  015               3160            DCR     C
056.175  302 156 056       3161            JNZ     DLL2            MORE ROOM FOR ERRORS
                           3162
                           3163  *         HAVE JUST FORMATTED ERROR LINE, SEE IF TO GO TO CONSOLE
                           3164
056.200  072 273 072       3165  DLL2.5    LDA     LISTFB+FB.FLG
056.203  247               3166            ANA     A
056.204  312 216 056       3167            JZ      DLL2.7          SEND TO CONSOLEE, FOR SURE
056.207  072 167 072       3168            LDA     ERRSHO
056.212  247               3169            ANA     A
056.213  312 234 056       3170            JZ      DLL3            JUST WRITE TO FILE
                           3171
                           3172
                           3173  *         TYPE LINE (WITH ERROR) ON CONSOLE
                           3174
056.216  041 120 072       3175  DLL2.7    LXI     H,DSPLIN
```

```
056.221  377 003      3176          DB      SYSCALL,.PRINT  PRINT HEADER
056.223  041 010 102  3177          LXI     H,LINE
056.226  315 074 064  3178          CALL    $TYPLZ          TYPE LINE TO 00
056.231  315 251 064  3179          CALL    $CRLF           END OF LINE AFTER IT
                      3180
                      3181  *       TYPE OUT LISTING LINE.
                      3182
056.234               3183  DLL3    EQU     *
056.234  315 233 055  3184          CALL    COL             COUNT OUTPUT LINE
056.237  041 010 102  3185          LXI     H,LINE
056.242  315 365 063  3186          CALL    $DTB            DELETE TRAILING BLANKS
056.245  075          3187          DCR     A
056.246  312 273 056  3188          JZ      DLL4            NO LINE TO LIST, MAYBE JUST HEADER
                      3189
                      3190  *       PRINT LINE HEADER AND BODY
                      3191
056.251  001 040 000  3192          LXI     B,DSPLEN
056.254  021 120 072  3193          LXI     D,DSPLIN
056.257  041 272 072  3194          LXI     H,LISTFB
056.262  315 072 066  3195          CALL    $FWRIB          WRITE LINE
056.265  021 010 102  3196          LXI     D,LINE
056.270  303 025 066  3197          JMP     $FWRIL          WRITE LINE AND RETURN
                      3198
                      3199  *       HAVE NO LINE BODY, SEND JUST HEADER
                      3200
056.273  041 120 072  3201  DLL4    LXI     H,DSPLIN
056.276  315 365 063  3202          CALL    $DTB            DELETE TRAILING BLANKS
056.301  353          3203          XCHG
056.302  041 272 072  3204          LXI     H,LISTFR
056.305  303 025 066  3205          JMP     $FWRIL          WRITE LINE AND RETURN
                      3206
056.310               3207  DLLB    EQU     *
056.310  001 125      3208          DB      ERR.U,'U'
056.312  002 122      3209          DB      ERR.R,'R'
056.314  004 104      3210          DB      ERR.D,'D'
056.316  010 101      3211          DB      ERR.A,'A'
056.320  020 126      3212          DB      ERR.V,'V'
056.322  040 106      3213          DB      ERR.F,'F'
056.324  100 117      3214          DB      ERR.O,'O'
056.326  200 120      3215          DB      ERR.P,'P'
056.330  000          3216          DB      0
```

```
                      3218  **      DRS - DECODE REGISTER SPECIFICATIOIN
                      3219  *
                      3220  *       DRS DECODES A REGISTER SPECIFICATION.
                      3221  *
                      3222  *       CALL    DRS
                      3223  *       DB      CODE
                      3224  *
                      3225  *       CODE =
                      3226  *       1       2               3
                      3227  *       B'0     B'1             B'00
                      3228  *       C 1     D 3             D 20
```

```
                                3229  *       D 2     H 5             H 40
                                3230  *       E 3     S 7             P 60
                                3231  *       H 4
                                3232  *       L 5
                                3233  *       M 6
                                3234  *       A 7
                                3235  *
                                3236  *       ENTRY   (DE) = OPERAND POINTER
                                3237  *       EXIT    (DE) UPDATED
                                3238  *               (A) = REGISTER INDEX
                                3239  *       ERR.R SET IF ERROR
                                3240  *       USES    A,F,D,E
                                3241
        056,331  343            3242  DRS     XTHL                                    /80.09.BB/
        056,332  305            3243          PUSH    B                               /80.09.BB/
        056,333  116            3244          MOV     C,M                             /80.09.BB/
        056,334  043            3245          INX     H                               /80.09.BB/
        056,335  106            3246          MOV     B,M             BC = TABLE FWA  /80.09.BB/
        056,336  043            3247          INX     H                               /80.09.BB/
        056,337  345            3248          PUSH    H                               /80.09.BB/
        056,340  151            3249          MOV     L,C                             /80.09.BB/
        056,341  140            3250          MOV     H,B             HL = TABLE FWA  /80.09.BB/
        056,342  315 351 056     3251          CALL    DRS.                            /80.09.BB/
        056,345  341            3252          POP     H               RESTORE HL      /80.09.BB/
        056,346  301            3253          POP     B               RESTORE BC      /80.09.BB/
        056,347  343            3254          XTHL                                    /80.09.BB/
        056,350  311            3255          RET                                     /80.09.BB/
                                3256
        056,351  032            3257  DRS.    LDAX    D
        056,352  023            3258          INX     D
        056,353  315 304 064     3259          CALL    $TBLS
        056,356  176            3260          MOV     A,M             (A) = REGISTER SPECIFICATION
        056,357  037            3261          RAR
        056,360  302 003 057     3262          JNZ     DRS3            NO GOOD
                                3263
                                3264  *       HAVE VALID REGISTER.  DISCARD EXTRA CHARACTERS
                                3265
        056,363  365            3266          PUSH    PSW             SAVE REGISTER CODE
        056,364  033            3267          DCX     D
        056,365  023            3268  DRS1    INX     D
        056,366  032            3269          LDAX    D
        056,367  376 101        3270          CPI     'A'
        056,371  332 001 057     3271          JC      DRS2            NOT ALPHA
        056,374  376 133        3272          CPI     'Z'+1
        056,376  332 365 056     3273          JC      DRS1            IS ALPHA
        057,001  361            3274  DRS2    POP     PSW             (A) = CODE
        057,002  311            3275          RET
                                3276
                                3277  *       ILLEGAL REGISTER SPECIFICATION
                                3278
        057,003  315 026 062     3279  DRS3    CALL    SEF
        057,006  002            3280          DB      ERR.R           *R* ERROR
        057,007  257            3281          XRA     A
        057,010  311            3282          RET
```

```
                        3284  **      REGISTER VALUE TABLES.
                        3285
057.011                 3286  DRSA    EQU     *               GROUP 1
057.011  101 017        3287          DB      'A',7*2+1
057.013  102 001        3288          DB      'B',0*2+1
057.015  103 003        3289          DB      'C',1*2+1
057.017  104 005        3290          DB      'D',2*2+1
057.021  105 007        3291          DB      'E',3*2+1
057.023  110 011        3292          DB      'H',4*2+1
057.025  114 013        3293          DB      'L',5*2+1
057.027  115 015        3294          DB      'M',6*2+1
057.031  000            3295          DB      0
                        3296
057.032                 3297  DRSB    EQU     *               GROUP 2
057.032  102 003        3298          DB      'B',1*2+1
057.034  104 007        3299          DB      'D',3*2+1
057.036  110 013        3300          DB      'H',5*2+1
057.040  123 017        3301          DB      'S',7*2+1
057.042  000            3302          DB      0
                        3303
057.043                 3304  DRSC    EQU     *               GROUP 3
057.043  102 001        3305          DB      'B',00Q*2+1
057.045  104 041        3306          DB      'D',20Q*2+1
057.047  110 101        3307          DB      'H',40Q*2+1
057.051  120 141        3308          DB      'P',60Q*2+1
057.053  000            3309          DB      0


                        3312  **      E8B - EVALUATE 8 BIT EXPRESSION
                        3313  *
                        3314  *       E8B IS CALLED TO EVALUATE AN EXPRESSION AND TO INSURE THAT
                        3315  *       IS EVALUATES TO 8 BITS OR LESS. IF NOT, THE *V* ERROR
                        3316  *       IS FLAGGED.
                        3317  *
                        3318  *       ENTRY   (DE) = OPERAND POINTER
                        3319  *       EXIT    (C) = VALUE
                        3320  *               (DE) UPDATED
                        3321  *       USES    A,B,C,D,E,F
                        3322
                        3323
057.054  315 355 053    3324  E8B     CALL    EVL             EVALUATE EXPRESSION
057.057  072 052 073    3325          LDA     EXPREL
057.062  247            3326          ANA     A
057.063  304 100 057    3327          CNZ     E8B1            RELOCATION ERROR
057.066  170            3328          MOV     A,B
057.067  247            3329          ANA     A
057.070  310            3330          RZ                      IF 0
057.071  074            3331          INR     A
057.072  310            3332          RZ                      IF =0
057.073  315 026 062    3333          CALL    SEF             *V* ERROR
057.076  020            3334          DB      ERR.V
057.077  311            3335          RET
```

```
                        3336
                        3337  *       RELOCATION ERROR
                        3338
057.100 315 026 062     3339  E8B1    CALL    SEF
057.103 002             3340          DB      ERR.R           FLAG ERROR
057.104 311             3341          RET



                        3343  **      EPO - EVALUATE FOR PASS 1.
                        3344  *
                        3345  *       EPO IS CALLED WHEN AN EVALUATION IS REQUIRED DURING PASS 1.
                        3346  *
                        3347  *       IF PASS = 1, EVALUATE THE EXPRESSION. IF IT CONTAINS UNDEFINED
                        3348  *               SYMBOLS, DEFINE A SYMBOL NNNNNN, WHERE NNNNNN = THE
                        3349  *               STATEMENT NUMBER (IN OCTAL).
                        3350  *
                        3351  *       IF PASS = 2, SEE IF NNNNNN IS DEFINED. IF SO, WAS AN ERROR
                        3352  *               PASS 1, FLAG 'U' THIS PASS.
                        3353  *
                        3354  *       The leading zero flass this as not a normal symbol for XREF
                        3355  *                                               G. Chandler 80.06.09
                        3356  *
                        3357  *       ENTRY   (DE) = EXPRESSION POINTER
                        3358  *       EXIT    (DE) UPDATED
                        3359  *               (BC) = VALUE
                        3360  *       'Z' SET IF NO ERROR
                        3361  *       USES    A,B,C,D,E,F
                        3362  *
                        3363
057.105 345             3364  EPO     PUSH    H
                        3365
057.106 325             3366          PUSH    D                                       /80.06.sc/
057.107 315 164 064     3367          CALL    $MOVEL          Initialize the statement number /80.06.sc/
057.112 005 000 150     3368          DW      5,DSPLND,EPOA+1                         /80.06.sc/
057.120 321             3369          POP     D                                       /80.06.sc/
                        3370
057.121 315 355 053     3371          CALL    EVL             EVALUATE EXPRESSION
057.124 072 164 072     3372          LDA     PASS
057.127 017             3373          RRC
057.130 322 154 057     3374          JNC     EPO2            PASS = 2
                        3375
                        3376  *       PASS = 1
                        3377
057.133 072 202 072     3378          LDA     ERRFLG
000.000                 3379          ERRNZ   ERR.U-1         COSE ASSUMES = 1
057.136 037             3380          RAR
057.137 322 176 057     3381          JNC     EPO4            OK
                        3382
                        3383  *       HAVE 'U' ERROR, DEFINE NNNNNN
                        3384
057.142 325             3385          PUSH    D
057.143 021 204 057     3386          LXI     D,EPOA          (DE) = ADDRESS OF SYMBOL
057.146 315 041 062     3387          CALL    SST             SEARCH SYMBOL TABLE
057.151 303 175 057     3388          JMP     EPO3            RETURN WITH ERROR
```

```
                          3389
                          3390  *       PASS = 2
                          3391
057.154  325             3392  EPO2    PUSH    D
057.155  021 204 057      3393          LXI     D,EPOA
057.160  052 262 072      3394          LHLD    SYMFWA
057.163  315 311 060      3395          CALL    LVT.            LOCATE VALUE IN TABLE
057.166  322 175 057      3396          JNC     EPO3            NOT FOUND, OK
057.171  315 026 062      3397          CALL    SEF             *U* AND *A* ERRORS
057.174  011              3398          DB      ERR.U+ERR.A
057.175  321              3399  EPO3    POP     D
057.176  341              3400  EPO4    POP     H
057.177  072 202 072      3401          LDA     ERRFLG
057.202  247              3402          ANA     A               SET ERROR CODE
057.203  311              3403          RET                     RETURN
                          3404
057.204  060 060 060      3405  EPOA    DB      '000000','X'+80H        Leading '0' for XREF    /80.06.sc/
```

```
                          3407  **      ESR     -  Enter Symbolic Reference                     /80.03.sc/
                          3408  *
                          3409  *       ESR is called to enter a symbolic reference record
                          3410  *       into the the TEMP file.  The entry is of the form:
                          3411  *
                          3412  *               DB      'SYMBOL1'       7 character name
                          3413  *               DW      STATNO          statement number
                          3414  *               DB      REFTYPE         type of reference
                          3415  *
                          3416  *       Data is collected only during the second pass.
                          3417  *
                          3418  *
                          3419  *       ENTRY:  A       = reference type
                          3420  *
                          3421  *       EXIT:   PSW     = 'C' CLEAR IF NO ERRORS
                          3422  *                         'C' SET   IF    ERRORS
                          3423  *
                          3424  *       USES:   PSW
                          3425  *
                          3426
057.213  041 155 073      3427  ESR     LXI     H,LABEL
057.216  303 221 057      3428          JMP     ESR.
```

```
                          3430  **      ESR.    -  HL = ADDRESS OF LABEL (MINUS TERMINATED)
057.221  062 356 057      3431  ESR.    STA     ESRC            SAVE REFERECE TYPE
057.224  072 164 072      3432          LDA     PASS
057.227  376 002          3433          CPI     2
057.231  300              3434          RNZ                     NOT PASS 2
                          3435
                          3436  *       DON'T GENERATE AN ENTRY IF THERE IS NO TEMP FILE.
                          3437
057.232  072 025 073      3438          LDA     TEMPFB+FB.NAM
```

```
057.235  247              3439            ANA     A
057.236  310              3440            RZ
                          3441
                          3442  *         SAVE REGS.
                          3443
057.237  305              3444            PUSH    B
057.240  325              3445            PUSH    D
057.241  345              3446            PUSH    H
                          3447
                          3448  *         GENERATE ENTRY UNCONDITIONALLY FOR EVERYTHING
                          3449  *         EXCEPT 'REFERENCED IN EXPRESSION'.
                          3450
057.242  072 356 057      3451            LDA     ESRC
057.245  376 000          3452            CPI     XT.REF
057.247  302 275 057      3453            JNZ     ESR0
                          3454
                          3455  *         IF XREF OFF IN LISTING CONTROL, THEN DON'T GENERATE
                          3456  *         AN ENTRY.
                          3457
057.252  072 172 072      3458            LDA     LSTCTL          CHECK LISTING CONTROL
057.255  346 010          3459            ANI     LST.R             FOR XREF OFF
057.257  312 341 057      3460            JZ      ESR4            NOT TO COLLECT DATA
                          3461
                          3462  *         IF ST.NRF BIT IS SET FOR THIS SYMBOL, THEN DON'T
                          3463  *         GENERATE AN ENTRY.
                          3464
057.262  353              3465            XCHG
057.263  315 041 062      3466            CALL    SST             SEARCH TABLE FOR SYMBOL
057.266  176              3467            MOV     A,M
057.267  346 010          3468            ANI     ST.NRF
057.271  302 341 057      3469            JNZ     ESR4            BR IF ST.NRF BIT = 1
057.274  353              3470            XCHG
                          3471
                          3472  *         GENERATE THE ENTRY.
                          3473
057.275  021 345 057      3474  ESR0      LXI     D,ESRA
057.300  176              3475  ESR1      MOV     A,M             MOVE SYMBOL
057.301  022              3476            STAX    D
057.302  023              3477            INX     D
057.303  043              3478            INX     H
057.304  247              3479            ANA     A
057.305  362 300 057      3480            JP      ESR1            MOVE BYTES UNTIL THE LAST MINUS-TERMINATED ONE
                          3481
057.310  052 170 072      3482            LHLD    STATNO
057.313  042 354 057      3483            SHLD    ESRB            SAVE STATEMENT NUMBER
                          3484
057.316  001 012 000      3485            LXI     B,ESRAL         BC = NUMBER OF BYTES
057.321  021 345 057      3486            LXI     D,ESRA          DE = ADDRESS OF DATA
057.324  041 013 073      3487            LXI     H,TEMPFB        HL = FILE BUFFER ADDRESS
057.327  315 072 066      3488            CALL    $FWRIB
                          3489
057.332  052 204 072      3490            LHLD    XREFCNT
057.335  043              3491            INX     H
057.336  042 204 072      3492            SHLD    XREFCNT         COUNT THE REFERENCE
                          3493
                          3494  *         RESTORE REGS.
```

```
                        3495
057.341  341           3496  ESR4    POP     H
057.342  321           3497          POP     D
057.343  301           3498          POP     B
057.344  311           3499          RET
                        3500
057.345  061 062 063   3501  ESRA    DB      '1234567'       SYMBOL
057.354  000 000       3502  ESRB    DW      0               STATEMENT NUMBER
057.356  000           3503  ESRC    DB      0               X-REF HISTORY TYPE
000.012                 3504  ESRAL   EQU     *-ESRA



                        3506  **      FNP - FORCE NEW PAGE.
                        3507  *
                        3508  *       FNP CAUSES A PAGE EJECT, BY FORMFEED OR BY LINE FEED,
                        3509  *       WHICHEVER IS REQUIRED.
                        3510  *       PRINT HEADING IF REQUESTED.                             /WCZ062680/
                        3511  *
                        3512  *       1ST TIME ENTERED, THERE IS NO NEED TO DO SKIP TO NEW
                        3513  *       PAGE, ASSUME AT TOP OF NEW PAGE.                        /WCZ062680/
                        3514  *
                        3515  *       ENTRY   NONE
                        3516  *       EXIT    NONE
                        3517  *       USES    ALL
                        3518
                        3519
057.357                 3520  FNP     EQU     *                                           /WCZ062680/
057.357  076 001       3521          MVI     A,1             INDICATE PRINT
057.361  062 204 060   3522          STA     FNPD              THE HEADING
                        3523
057.364                 3524  FNP.    EQU     *
057.364  072 203 060   3525          LDA     FNPC            0, 1ST TIME HERE
057.367  247           3526          ANA     A
057.370  312 002 060   3527          JZ      FNP0              BR IF NOT
057.373  257           3528          XRA     A               RESET FLAG
057.374  062 203 060   3529          STA     FNPC
057.377  303 065 060   3530          JMP     FNP3
                        3531
060.002                 3532  FNP0    EQU     *                                           /WCZ062680/
060.002  072 261 072   3533          LDA     FORMDP
060.005  247           3534          ANA     A
060.006  302 030 060   3535          JNZ     FNP1            MUST LINE FEED
                        3536
                        3537  *       DEVICE WILL TAKE A FORM FEED
                        3538
060.011  001 001 000   3539          LXI     B,1
060.014  021 201 060   3540          LXI     D,FNPA
060.017  041 272 072   3541          LXI     H,LISTFB
060.022  315 072 066   3542          CALL    $FWRIB          WRITE
060.025  303 065 060   3543          JMP     FNP3            ADJUST LINE COUNT
                        3544
                        3545  *       MUST USE CRLF'S TO GET THERE
                        3546
060.030  041 260 072   3547  FNP1    LXI     H,PAGEDP
```

```
060.033  226            3548           SUB     M               (A) = GAP SPACE
060.034  041 050 073    3549           LXI     H,LINCNT
060.037  206            3550           ADD     M               (A) = AMOUNT NEEDED
060.040  312 065 060    3551           JZ      FNP3            IF NO LINES (IS THE CASE AT START OF ASSEMBLY)
060.043  001 001 000    3552  FNP2     LXI     B,1             (BC) = COUNT
060.046  021 202 060    3553           LXI     D,FNPB
060.051  041 272 072    3554           LXI     H,LISTFB
060.054  365            3555           PUSH    PSW             SAVE COUNT
060.055  315 072 066    3556           CALL    $FWRIB          WRITE BYTE
060.060  361            3557           POP     PSW
060.061  075            3558           DCR     A
060.062  302 043 060    3559           JNZ     FNP2            GO SOME MORE
                        3560
                        3561  *        PRINT HEADING,                                          /WCZ062680/
                        3562
060.065                 3563  FNP3     EQU     *
060.065  072 204 060    3564           LDA     FNPD
060.070  247            3565           ANA     A
060.071  310            3566           RZ                      REQUESTED NOT TO PRINT HEADING
                        3567
060.072  072 051 073    3568           LDA     PAGNUM          INCREMENT PAGE NUMBER
060.075  074            3569           INR     A
060.076  062 051 073    3570           STA     PAGNUM
                        3571
060.101  117            3572           MOV     C,A             UNPACK INTO HEADING LINE
060.102  006 000        3573           MVI     B,0
060.104  041 113 072    3574           LXI     H,HEADA
060.107  076 003        3575           MVI     A,HEADAL
060.111  315 157 031    3576           CALL    $UDD
                        3577
060.114  041 113 072    3578           LXI     H,HEADA         POINT PAGE NO.                  /80.09.BB/
060.117  016 002        3579           MVI     C,HEADAL-1      CHECK THIS MANY LEADING 0       /80.09.BB/
060.121  176            3580  FNP4     MOV     A,M
060.122  376 060        3581           CPI     '0'             IS IT A ZERO?                   /80.09.BB/
060.124  302 136 060    3582           JNZ     FNP5            NO, BAIL OUT                    /80.09.BB/
060.127  066 040        3583           MVI     M,' '           YES, MAKE IT A SPACE            /80.09.BB/
060.131  043            3584           INX     H               BUMP TO NEXT                    /80.09.BB/
060.132  015            3585           DCR     C               CHECKED ALL?                    /80.09.BB/
060.133  302 121 060    3586           JNZ     FNP4.           NO, DO NEXT                     /80.09.BB/
                        3587
000.000                 3588           ERRNZ   *-FNP5          ENSURE FALL THROUGH             /80.09.BB/
060.136                 3589  FNP5     EQU     *                                               /80.09.BB/
060.136  001 221 000    3590           LXI     B,HEADLEN
060.141  021 277 071    3591           LXI     D,HEADING
060.144  041 272 072    3592           LXI     H,LISTFB
060.147  315 072 066    3593           CALL    $FWRIB          WRITE HEADING                   /WCZ062680/
                        3594
                        3595  *        ADJUST PAGE LINE COUNT
                        3596
060.152  072 260 072    3597           LDA     PAGEDP
060.155  326 003        3598           SUI     3               (A) = SPACES ON PAGE -HEADING SIZE
060.157  062 050 073    3599           STA     LINCNT          SET LINES REMAINING
060.162  311            3600           RET                     DONE
                        3601
                        3602  *        FORCE NEWPAGE WITHOUT PAGE HEADING.
                        3603
```

```
060.163                 3604  FNP6    EQU     *
060.163  257            3605          XRA     A               INDICATE DON'T
060.164  062 204 060     3606          STA     FNPD              PRINT HEADING
060.167  315 364 057     3607          CALL    FNP.
060.172  072 260 072     3608          LDA     PAGEDP
060.175  062 050 073     3609          STA     LINCNT
060.200  311            3610          RET
                        3611
060.201  014            3612  FNPA    DB      FF              FORM FEED
060.202  012            3613  FNPB    DB      NL              NEW LINE
060.203  001            3614  FNPC    DB      1       FLAG := <>0 INDICATES 1ST TIME HERE     /WCZ062680/
060.204                 3615  FNPD    DS      1       FLAG := <>0 PRINT PAGE HEADING          /WCZ062680/


                        3617  **      GRT - GENERATE RELOCATION TABLE.
                        3618  *
                        3619  *       GRT IS CALLED AT THE EEND OF PASS 2 TO GENERATE
                        3620  *       ANY RELOCATION TABLES NEEDED.
                        3621  *
                        3622  *       ENTRY   NONE
                        3623  *       EXIT    NONE
                        3624  *       USES    ALL
                        3625
                        3626
060.205  072 210 072     3627  GRT     LDA     FTFLAG
000.000                 3628          ERRNZ   FT.PIC-1
060.210  075            3629          DCR     A
060.211  300            3630          RNZ                     NOT PIC
060.212  052 270 072     3631          LHLD    RELPTR
060.215  353            3632          XCHG
060.216  052 266 072     3633          LHLD    RELLWA          (DE) = LOW TABLE ADDR, (HL) = HIGH TABLE ADDR
                        3634
060.221  315 216 030     3635  GRT1    CALL    $CDEHL
060.224  312 245 060     3636          JE      GRT2            ALL DONE
                        3637
                        3638  *       WRITE ELEMENT TO BINARY
                        3639
060.227  053            3640          DCX     H
060.230  106            3641          MOV     B,M
060.231  053            3642          DCX     H
060.232  176            3643          MOV     A,M
060.233  315 354 060     3644          CALL    OBB             OUTPUT
060.236  170            3645          MOV     A,B
060.237  315 354 060     3646          CALL    OBB             OUTPUT 2ND HALF
060.242  303 221 060     3647          JMP     GRT1            SEE IF MORE
                        3648
060.245  257            3649  GRT2    XRA     A
060.246  315 354 060     3650          CALL    OBB             FINISH TABLE
060.251  257            3651          XRA     A
060.252  303 354 060     3652          JMP     OBB             WRITE 2ND 00 AND EXIT
```

```
                        3654  **      GSC - GET STRING CHARACTER
                        3655  *
                        3656  *       GSC READS A CHARACTER FROM A QUOTED STRING IN THE SOURCE LINE.
                        3657  *
                        3658  *       A DOUBLE QUOTE ('') IS TAKEN AS ONE QUOTE CHARACTER.
                        3659  *
                        3660  *       ENTRY   (DE) = POINTER TO NEXT CHARACTER
                        3661  *       EXIT    (DE) UPDATED
                        3662  *               (A) = CHARACTER
                        3663  *               'Z' SET IF END OF STRING
                        3664  *       USES    A,F,D,E
                        3665
                        3666
060.255  032           3667  GSC     LDAX    D               (A) = NEXT LINE CHARACTER
060.256  023           3668          INX     D
060.257  247           3669          ANA     A               SEE IF END OF LINE
060.260  312 301 060    3670          JZ      GSC4            GONE PAST END
060.263  376 047        3671          CPI     QUOTE
060.265  300           3672          RNE                     NOT END QUOTE
                        3673
                        3674  *       HAVE END-QUOTE
                        3675
060.266  032           3676          LDAX    D
060.267  376 047        3677          CPI     QUOTE
060.271  302 277 060    3678          JNE     GSC3            NOT DOUBLE QUOTE, IS END QUOTE
060.274  023           3679          INX     D
060.275  267           3680          ORA     A               CLEAR 'Z'
060.276  311           3681          RET                     RETURN WITH QUOTE
                        3682
060.277  257           3683  GSC3    XRA     A               SET 'Z'
060.300  311           3684          RET
                        3685
                        3686  *       GONE PAST END OF LINE WITHOUT TRAILING QUOTE
                        3687
060.301  315 026 062    3688  GSC4    CALL    SEF
060.304  010           3689          DB      ERR,A
060.305  033           3690          DCX     D
060.306  033           3691          DCX     D
060.307  257           3692          XRA     A               FLAG END OF STRING
060.310  311           3693          RET



                        3695  **      LVT - LOCATE VALUE IN TABLE.
                        3696  *
                        3697  *       LVT LOOKS UP A VALUE IN A TABLE.
                        3698  *
                        3699  *       ENTRY   (HL) = TBL ADDRESS
                        3700  *               (DE) = VALUE ADDRESS
                        3701  *               (A) = NOP IF 2 DATA BYTES PER ENTRY, = INX H INSTRUCTION
                        3702  *                     IF 3 DATA BYTES PER ENTRY.
                        3703  *       EXIT    'C' SET IF FOUND
                        3704  *               (HL) = ADDRESS
                        3705  *               'C' CLEAR IF NOT FOUND
                        3706  *               (HL) = ADDRESS OF NEXT EMPTY
```

```
                        3707  *       USES    A,F,H,L
                        3708
                        3709
060.311 076 043         3710  LVT.    MVI     A,MI.INXH       SYMTAB SEARCH ENTRY POINT
                        3711
060.313                 3712  LVT     EQU     *
060.313 062 347 060     3713          STA     LVTA            SET INX OR NOP INSTRUCTION
060.316 176             3714  LVT0    MOV     A,M             (A) = TABLE FIRST ENTRY
060.317 247             3715          ANA     A
060.320 310             3716          RZ                      TABLE EXHAUSTED
060.321 325             3717          PUSH    D               SAVE (DE)
                        3718
                        3719  *       COMPARE ENTRYS
                        3720
060.322 032             3721  LVT1    LDAX    D               COMPARE CHARACTERS
060.323 276             3722          CMP     M
060.324 302 337 060     3723          JNE     LVT2            NO MATCH
060.327 023             3724          INX     D
060.330 043             3725          INX     H
060.331 027             3726          RAL
060.332 322 322 060     3727          JNC     LVT1            MORE TO CHECK
060.335 321             3728          POP     D
060.336 311             3729          RET                     FOUND ENTRY
                        3730
                        3731
                        3732  *       NOT FOUND
                        3733
060.337 176             3734  LVT2    MOV     A,M
060.340 043             3735          INX     H
060.341 247             3736          ANA     A
060.342 302 337 060     3737          JP      LVT2            NOT AT END OF ENTRY
060.345 043             3738          INX     H
060.346 043             3739          INX     H
060.347 043             3740  LVTA    INX     H
060.350 321             3741          POP     D
060.351 303 316 060     3742          JMP     LVT0            LOOK AGAIN


                        3744  **      OBB - OUTPUT BINARY BYTE.
                        3745  *
                        3746  *       OBB IS CALLED TO OUTPUT A BINARY BYTE.
                        3747  *       THE BYTE IS ADDED TO THE BINARY FILE, AND IS ADDED TO THE
                        3748  *       LISTING (IF APPROPRIATE).
                        3749  *               ORG IS INCREMENTED.
                        3750  *
                        3751  *       * * NOTE * *    EVERYBODY WHO GENERATES ANY BINARY
                        3752  *       INFORMATION MUST DO IT VIA A CALL TO OBB! THE ONLY EXCEPTION
                        3753  *       IS THE TWO BYTES BACK-GENERATED INTO THE PIC HEADERE
                        3754  *       BY THE MAIN LOOP. OBB CALLS ABV, AND ABV (DURING PASS 1) KEEPS
                        3755  *       TRACK OF THE RANGE OF BINARY GENERATED.
                        3756  *
                        3757  *       IF PASS = 1, DO NOTHING
                        3758  *
                        3759  *       ENTRY   (A) = VALUE
```

```
                                3760  *       EXIT    NONE
                                3761  *       USES    A,F
                                3762
                                3763
060.354  345                   3764  OBB     PUSH    H
060.355  052 176 072            3765          LHLD    ORG
060.360  043                   3766          INX     H
060.361  042 176 072            3767          SHLD    ORG
060.364  062 034 061            3768          STA     OBBA            SAVE VALUE
060.367  072 164 072            3769          LDA     PASS
060.372  017                   3770          RRC
060.373  332 044 061            3771          JC      OBB3            PASS = 1
060.376  052 162 072            3772  OBB1    LHLD    OBBPTR
061.001  076 150                3773          MVI     A,$DSPLIM       COMPARE TO LIMIT
061.003  275                   3774          CMP     L
061.004  302 033 061            3775          JNE     OBB2            NOT 3 VALUES YET
                                3776
                                3777  *       LINE IS FULL. IF *G* SET, PRINT AND ADD ENTRY
                                3778
061.007  072 172 072            3779          LDA     LSTCTL
000.000                         3780          ERRNZ   LST.G-200Q      CODE ASSUMES = 200Q
061.012  027                   3781          RAL
061.013  322 044 061            3782          JNC     OBB3            *G* CLEAR
061.016  325                   3783          PUSH    D
061.017  305                   3784          PUSH    B               PRESERVE REGISTERS
061.020  315 074 056            3785          CALL    DLL             DISPLAY LISTING LINE
061.023  315 320 061            3786          CALL    PDL             PREPARE DISPLAY LINE
061.026  301                   3787          POP     B               RESTORE REGISTERS
061.027  321                   3788          POP     D
061.030  303 376 060            3789          JMP     OBB1
                                3790
                                3791  *       ADD TO LINE
                                3792
061.033  076 000                3793  OBB2    MVI     A,0
061.034                         3794  OBBA    EQU     *-1             VALUE STORED HERE
061.035  315 257 064            3795          CALL    $UOD            UNPACK OCTAL DIGITS
061.040  043                   3796          INX     H
061.041  042 162 072            3797          SHLD    OBBPTR          SET NEW POINTER VALUE
061.044  072 034 061            3798  OBB3    LDA     OBBA            (A) = VALUE
061.047  315 232 054            3799          CALL    ABV             ADD BINARY VALUE
061.052  341                   3800          POP     H
061.053  311                   3801          RET
```

```
                                3803  **      PAS - PRINT ASSEMBLY STATISTICS.
                                3804  *
                                3805  *       PAS PRINTS THE FINAL ASSEMBLY STATISTICS.
                                3806  *
                                3807  *       STATEMENTS = NNN
                                3808  *       NO ERRORS DETECTED          [OR]
                                3809  *       ERRORS = NN
                                3810  *
                                3811  *       ENTRY   ERRCNT = # OF ERRORS
                                3812  *               LINCNT = # OF LINES
```

```
                              3813  *       EXIT    NONE
                              3814  *       USES    ALL
                              3815
                              3816
061.054 052 170 072           3817  PAS     LHLD    STATNO
061.057 104                   3818          MOV     B,H
061.060 115                   3819          MOV     C,L
061.061 041 210 061           3820          LXI     H,PASB
061.064 076 005               3821          MVI     A,5
061.066 315 157 031           3822          CALL    $UDD            UNPACK STATEMENT COUNT
061.071 315 270 055           3823          CALL    CUS             COMPUTE UNUSED SPACE
061.074 104                   3824          MOV     B,H
061.075 115                   3825          MOV     C,L             (BC) = COUNT
061.076 322 104 061           3826          JNC     PAS0            NOT ALL USED UP
061.101 001 000 000           3827          LXI     B,0             ALL USED UP
061.104 041 243 061           3828  PAS0    LXI     H,PASC
061.107 076 005               3829          MVI     A,5
061.111 315 157 031           3830          CALL    $UDD            UNPACK FREE BYTES COUNT
061.114 052 165 072           3831          LHLD    ERRCNT
061.117 104                   3832          MOV     B,H
061.120 115                   3833          MOV     C,L             (DE) = ERROR COUNT
061.121 041 264 061           3834          LXI     H,PASD
061.124 076 005               3835          MVI     A,5
061.126 305                   3836          PUSH    B               SAVE COUNT
061.127 315 157 031           3837          CALL    $UDD            UNPACK COUNT
061.132 301                   3838          POP     B
061.133 170                   3839          MOV     A,B
061.134 261                   3840          ORA     C
061.135 302 151 061           3841          JNZ     PAS2            HAVE ERROR COUNT
061.140 315 164 064           3842          CALL    $MOVEL
061.143 005 000 313           3843          DW      5,PASE,PASD     USE ' NO ' IN PLACE OF COUNT
061.151                       3844  PAS2    EQU     *
061.151 072 050 073           3845          LDA     LINCNT          MUST BE AT LEAST 5 LINES        /WCZ062680/
061.154 376 005               3846          CPI     5                 LEFT ON THIS PAGE             /WCZ062680/
061.156 334 357 057           3847          CC      FNP                 OTHERWISE FNP               /WCZ062680/
061.161 001 104 000           3848          LXI     B,PASAL
061.164 021 206 061           3849          LXI     D,PASA
061.167 041 272 072           3850          LXI     H,LISTFB
061.172 315 072 066           3851          CALL    $FWRIB          WRITE TO LISTING FILE
061.175 072 050 073           3852          LDA     LINCNT          ADJUST LINE COUNTER             /WCZ062680/
061.200 326 005               3853          SUI     5                                               /WCZ062680/
061.202 062 050 073           3854          STA     LINCNT                                          /WCZ062680/
061.205 311                   3855          RET                     EXIT
                              3856
061.206 012 012               3857  PASA    DB      NL,NL
061.210 060 060 060           3858  PASB    DB      '00000 Statements Assembled',NL
061.243 060 060 060           3859  PASC    DB      '00000 Bytes Free',NL
061.264 060 060 060           3860  PASD    DB      '00000 Errors Detected',NL
000.104                       3861  PASAL   EQU     $-PASA
061.312 212                   3862          DB      ENL             END FOR 'PRINT' STATEMENT
061.313 040 116 157           3863  PASE    DB      ' No',0,0
```

```
                                3865  **      PDL - PREPARE DISPLAY LINE
                                3866  *
                                3867  *       PDL PRESETS THE DISPLAY LINE BY BLANKING IT OUT.
                                3868  *
                                3869  *       ENTRY   NONE
                                3870  *       EXIT    LSTLIN = BLANKS
                                3871  *       USES    A,F,H,L
                                3872
                                3873
061.320  041 120 072            3874  PDL     LXI     H,DSPLIN
061.323  076 040               3875          MVI     A,DSPLEN
061.325  066 040               3876  PDL1    MVI     M,' '
061.327  043                  3877          INX     H
061.330  075                  3878          DCR     A
061.331  302 325 061           3879          JNZ     PDL1
061.334  062 010 102           3880          STA     LINE            ZERO LINE
061.337  041 134 072           3881          LXI     H,DSPLNB
061.342  042 162 072           3882          SHLD    OBBPTR          SET NEW POINTER VALUE
061.345  311                  3883          RET


                                3885  **      RRI - RECORD RELOCATION INFORMATION.
                                3886  *
                                3887  *       RRI IS CALLED WHEN AN BINARY ADDRESS VALUE IS
                                3888  *       ABOUT TO BE PRODUCED. IF IT IS RELOCATABLE (EXPREL M<> 0)
                                3889  *       THEN ITS ADDRESS IS ENTERED IN THE RELOCATION TABLE.
                                3890  *
                                3891  *       ENTRY   ORG = ORG FOR VALUEE
                                3892  *               EXPREL <> 0 IF RELOCATABLE
                                3893  *       EXIT    NONE
                                3894  *       USES    A,F
                                3895
                                3896
061.346  072 052 073           3897  RRI     LDA     EXPREL
061.351  247                  3898          ANA     A
061.352  310                  3899          RZ                      NOT RELOCATABLE
061.353  072 164 072           3900          LDA     PASS
061.356  075                  3901          DCR     A
061.357  310                  3902          RZ                      IS PASS 1
061.360  325                  3903          PUSH    D
061.361  345                  3904          PUSH    H               SAVE REGS
061.362  315 270 055           3905          CALL    CUS             COMPUTE UNUSED SPACE
061.365  332 061 062           3906          JC      MEMOVR          OVERFLOW
061.370  052 176 072           3907          LHLD    ORG
061.373  353                  3908          XCHG
061.374  052 270 072           3909          LHLD    RELPTR
061.377  053                  3910          DCX     H               ADD VALUE TO LIST
062.000  162                  3911          MOV     M,D
062.001  053                  3912          DCX     H
062.002  163                  3913          MOV     M,E
062.003  042 270 072           3914          SHLD    RELPTR
062.006  341                  3915          POP     H
062.007  321                  3916          POP     D               RESTORE REGS
062.010  311                  3917          RET
```

```
                                3918
                                3919


                                3921  **      RSF - REWIND SOURCE FILE.
                                3922  *
                                3923  *       RSF IS CALLED TO REWIND THE INPUT SOURCE FILE.
                                3924  *
                                3925  *       ENTRY   RSFA = TEXT NAME
                                3926  *               RSFB = TEXT LENGTH
                                3927  *       EXIT    TAPE POSITIONED
                                3928  *       USES    ALL
                                3929
                                3930
062.011  001 000 000            3931  RSF     LXI     B,0
062.014  076 002                3932          MVI     A,CN.SOU
062.016  377 047                3933          DB      SYSCALL,.POSIT          REWIND SOURCE FILE
062.020  041 325 072            3934          LXI     H,SORCFB
062.023  303 202 065            3935          JMP     $FCLEAR         CLEAR FILE BLOCK AND EXIT


                                3937  **      SEF - SET ERROR FLAGS.
                                3938  *
                                3939  *       SEF SETS THE SPECIFIED ERROR IN *ERRFLG*, THEN RETURSN TO
                                3940  *       RET+1
                                3941  *
                                3942  *       CALL    SEF
                                3943  *       DB      ERRBIT
                                3944  *
                                3945  *       ENTRY   (RET) = ERROR
                                3946  *       EXIT    ERROR SET
                                3947  *               RETURN TO (RET)+1
                                3948  *       USES    A,F
                                3949
                                3950
062.026  343                    3951  SEF     XTHL                    (HL) = RETURN ADDRESS
062.027  072 202 072            3952          LDA     ERRFLG
062.032  266                    3953          ORA     M
062.033  062 202 072            3954          STA     ERRFLG
062.036  043                    3955          INX     H
062.037  343                    3956          XTHL                    ADVANCE EXIT
062.040  311                    3957          RET                     RETURN PAST CODE
```

```
                                3959  **      SST - SEARCH SYMBOL TABLE.
                                3960  *
                                3961  *       SST SCANS THE SYMTAB FOR A GIVEN ENTRY. IF FOUND, RETURN
                                3962  *       IF NOT FOUND, CREATE AS TYPE *U*
                                3963  *
                                3964  *       ENTRY   (DE) = ADDRESS OF SYMBOL
                                3965  *       EXIT    (DE) UNCHANGED
                                3966  *               (HL) = ADDRESS OF START OF VALUE BYTES.
                                3967  *       USES    A,F,H,L
                                3968
                                3969
062.041 052 262 072             3970  SST     LHLD    SYMFWA
062.044 315 311 060             3971          CALL    LVT.            LOCATE VALUE IN TABLE
062.047 330                     3972          RC                      FOUND IT
062.050 325                     3973          PUSH    D               SAVE (DE)
                                3974
                                3975  *       WILL CREATE NEW ENTRY, SEE IF TABLE OVERFLOW.
                                3976
062.051 345                     3977          PUSH    H
062.052 315 270 055             3978          CALL    CUS             COMPUTE UNUSED SPACE
062.055 341                     3979          POP     H
062.056 322 114 062             3980          JNC     SST1            OK
062.061 315 136 031             3981  MEMOVR  CALL    $TYPTX
062.064 012 012 007             3982          DB      NL,NL,BELL,'Symtab Overflow',NL,NL,BELL+200Q
062.111 303 074 047             3983          JMP     END.            FORCE END OF THIS PASS
                                3984
062.114 321                     3985  SST1    POP     D               REFRESH (DE)
062.115 325                     3986          PUSH    D
                                3987
                                3988  *       NOT FOUND, PUT IN TABLE, SET TYPE = ST,UND
                                3989
062.116 032                     3990  SST2    LDAX    D
062.117 167                     3991          MOV     M,A
062.120 023                     3992          INX     D
062.121 043                     3993          INX     H
062.122 007                     3994          RLC
062.123 322 116 062             3995          JNC     SST2
062.126 345                     3996          PUSH    H
062.127 021 020 000             3997          LXI     D,16
062.132 031                     3998          DAD     D
062.133 042 264 072             3999          SHLD    SYMPTR          SET SYMBOL TABLE LIMIT
062.136 341                     4000          POP     H
062.137 321                     4001          POP     D
062.140 311                     4002          RET                     SYMBOL CREATED IN TABLE


                                4004  **      UOL - UNPACK ORG INTO LINE.
                                4005  *
                                4006  *       UOL UNPACKS THE ORIGIN VALUE INTO THE LISTING LINE.
                                4007  *
                                4008  *       ENTRY   NONE
                                4009  *       EXIT    (HL) = SORG
                                4010  *       USES    A,F,H,L
                                4011
```

```
                              4012
062.141  052 200 072  4013  UOL     LHLD    SORG
062.144  345          4014  UOL.    PUSH    H
062.145  325          4015          PUSH    D
062.146  353          4016          XCHG                    (DE) = VALUE
062.147  041 123 072  4017          LXI     H,DSPLNA
062.152  172          4018          MOV     A,D
062.153  315 257 064  4019          CALL    $UOD            UNPACK BANK
062.156  066 056      4020          MVI     M,'.'           SET PERIOD BETWEEN BANKS
062.160  043          4021          INX     H
062.161  173          4022          MOV     A,E
062.162  315 257 064  4023          CALL    $UOD            UNPACK ADDR
062.165  321          4024          POP     D
062.166  341          4025          POP     H
062.167  311          4026          RET


                              4028  **      UNL - UNPACK NEXT LINE.
                              4029  *
                              4030  *       UNL UNPACKS THE NEXT SOURCE LINE FROM THE TEXT BUFFER. IF THE
                              4031  *       IS NO MORE TEXT, AND *END* LINE IS GENERATED.
                              4032  *
                              4033  *       ENTRY   NONE
                              4034  *       EXIT    'Z' SET IF COMMENT
                              4035  *       USES    ALL
                              4036
                              4037
062.170                       4038  UNL     EQU     *                                       /80.02.GC/
                              4039
062.170  052 170 072  4040          LHLD    STATNO                                          /80.02.GC/
062.173  104          4041          MOV     B,H                                             /80.02.GC/
062.174  115          4042          MOV     C,L             BC = STATMENT NUMBER            /80.02.GC/
062.175  041 150 072  4043          LXI     H,DSPLND        HL = ADDRESS TO PAT AT          /80.02.GC/
062.200  076 005      4044          MVI     A,5             A  = DIGIT COUNT                /80.02.GC/
062.202  315 157 031  4045          CALL    $UDD            OUTPUT THE LINE NUMBER          /80.02.GC/
                              4046
062.205  001 144 000  4047          LXI     B,LINEMAX
062.210  021 010 102  4048          LXI     D,LINE
062.213  072 206 072  4049          LDA     XTXFLG
062.216  062 207 072  4050          STA     XTXLINE         XTXLINE <> 0 IFF READING FROM XTEXT
062.221  247          4051          ANA     A
062.222  312 250 062  4052          JZ      UNL00           FROM MAIN SOURCE
062.225  041 360 072  4053          LXI     H,XTXFB
062.230  315 222 065  4054          CALL    $FREAL          READ LINE
062.233  322 273 062  4055          JNC     UNL0            GOT IT
                              4056
                              4057  *       EOF ON XTEXT FILE
                              4058
062.236  315 075 065  4059          CALL    $FCLO           CLOSE
062.241  257          4060          XRA     A
062.242  062 206 072  4061          STA     XTXFLG
062.245  303 170 062  4062          JMP     UNL             TRY AGAIN
                              4063
062.250  041 325 072  4064  UNL00   LXI     H,SORCFB
```

```
062.253  315 222 065  4065           CALL    $FREAL          READ LINE
062.256  322 273 062  4066           JNC     UNL0            OK
                      4067
                      4068  *        EOF ON MAIN PROGRAM, FAKE AN END STATEMENT
                      4069
062.261  315 164 064  4070           CALL    $MOVEL
062.264  027 000 117  4071           DW      UNLAL,UNLA,LINE         USE END STATEMENT
062.272  353          4072           XCHG                    (DE) = LINEE LWA
062.273  072 010 102  4073  UNL0     LDA     LINE
062.276  376 052      4074           CPI     '*'
062.300  310          4075           RE                      IS COMMENT
062.301  041 010 102  4076           LXI     H,LINE          (HL) = LINE POINTER
                      4077
                      4078  *        STRIP OFF LABEL.
                      4079
062.304  021 155 073  4080           LXI     D,LABEL
062.307  006 011      4081           MVI     B,8+1           8 CHARACTER MAX                 /80.02.GC/
062.311  315 040 063  4082           CALL    UNL3            STRIP LABEL
062.314  032          4083           LDAX    D               CHECK FOR ':'
062.315  326 272      4084           SUI     ':'+200Q
062.317  302 330 062  4085           JNZ     UNL0.5          NOT :
062.322  022          4086  UNL0.3   STAX    D               CLEAR IT
062.323  033          4087           DCX     D
062.324  032          4088           LDAX    D
062.325  366 200      4089           ORI     200Q
062.327  022          4090           STAX    D               FLAG LAST
062.330  076 164      4091  UNL0.5   MVI     A,$LABEL+7      MAKE SURE AM NOW 7 OR LESS      /80.02.GC/
062.332  223          4092           SUB     E
062.333  302 346 062  4093           JNE     UNL0.7          IS OK
062.336  315 026 062  4094           CALL    SEF
062.341  040          4095           DB      ERR.F
062.342  257          4096           XRA     A
062.343  303 322 062  4097           JMP     UNL0.3
                      4098
                      4099  *        VERIFY LABEL STARTS WITH ALPHA                          /80.02.GC/
                      4100
062.346  021 155 073  4101  UNL0.7   LXI     D,LABEL                                         /80.02.GC/
062.351  032          4102           LDAX    D                                               /80.02.GC/
062.352  247          4103           ANA     A                                               /80.02.GC/
062.353  312 374 062  4104           JZ      UNL0.9          NO LABEL DEFINED                /80.02.GC/
062.356  346 177      4105           ANI     177Q            STRIP OFF ANY  *END*  BIT      /80.02.GC/
062.360  315 174 053  4106           CALL    LCT,            LOOK-UP CHARACTER TYPE          /80.02.GC/
062.363  346 200      4107           ANI     CT.ALPH                                         /80.02.GC/
062.365  302 374 062  4108           JNZ     UNL0.9          CHARACTER IS ALPHA              /80.02.GC/
062.370  315 026 062  4109           CALL    SEF                                             /80.02.GC/
062.373  040          4110           DB      ERR.F           FLAG FORMAT ERROR               /80.02.GC/
                      4111
                      4112  *        STRIP OFF OPCODE.
                      4113
062.374  021 166 073  4114  UNL0.9   LXI     D,OPCODE
062.377  006 006      4115           MVI     B,5+1
063.001  315 040 063  4116           CALL    UNL3
                      4117
                      4118  *        COPY EXPRESSION TO WORKAREA.
                      4119
063.004  021 173 073  4120           LXI     D,EXPWRK
```

```
063.007  176            4121  UNL1    MOV     A,M
063.010  022            4122          STAX    D
063.011  023            4123          INX     D
063.012  043            4124          INX     H
063.013  247            4125          ANA     A
063.014  302 007 063     4126          JNZ     UNL1            NOT AT END OF LINE
063.017  023            4127          INX     D
063.020  022            4128          STAX    D               SET ZERO BYTE AT END
063.021  033            4129          DCX     D
063.022  076 040        4130          MVI     A,' '
063.024  022            4131          STAX    D               GUARANTEE ' ',000 ENDING
                        4132
                        4133  *       SEE IF COMMENT.
                        4134  *
                        4135  *       IF NO LABEL OR OPCODE, IS COMMENT
                        4136
063.025  072 155 073    4137  UNL2.5  LDA     LABEL
063.030  346 177        4138          ANI     177Q
063.032  300            4139          RNZ                     NOT COMMENT
063.033  072 166 073    4140          LDA     OPCODE
063.036  247            4141          ANA     A
063.037  311            4142          RET


                        4144  **      UNL3 - PARSE LABEL OR OPCODE.
                        4145  *
                        4146  *       COPY A CHARACTER STRING FROM (HL) TO (DE)
                        4147  *       UNTIL TAB, SPACE, OR END OF LINE IS SEEN.
                        4148  *
                        4149  *       IF NONE COPIED, ZERO (DE)
                        4150  *       IF SOME COPIED, SET 80H ON LAST CHARACTER
                        4151  *
                        4152  *       ENTRY   (B) = MAX CHARACTER COUNT
                        4153  *       EXIT    (DE) = ADDRESS OF LAST CHARACTER
                        4154
                        4155
063.040  176            4156  UNL3    MOV     A,M
063.041  022            4157          STAX    D
063.042  247            4158          ANA     A
063.043  312 072 063    4159          JZ      UNL5            IS END OF LINE
063.046  043            4160          INX     H
063.047  376 040        4161          CPI     ' '
063.051  312 072 063    4162          JE      UNL5            IS SPACE
063.054  376 011        4163          CPI     TAB
063.056  312 072 063    4164          JE      UNL5            IS TAB
063.061  023            4165          INX     D
063.062  005            4166          DCR     B
063.063  302 040 063    4167          JNZ     UNL3            MORE TO GO
                        4168
                        4169  *       TOO MANY CHARACTERS, FLAG AN *F* ERROR
                        4170
063.066  315 026 062    4171          CALL    SEF             *F* ERROR
063.071  040            4172          DB      ERR.F
                        4173
```

```
                                4174  *       ITEM COPIED, SET SIGN BIT AND 0 NEXT BYTE
                                4175
063.072  257                    4176  UNL5    XRA     A
063.073  022                    4177          STAX    D               CLEAR FIELD
063.074  033                    4178          DCX     D               SET SIGN OVER LAST CHARACTER
063.075  032                    4179          LDAX    D
063.076  366 200                4180          ORI     80H
063.100  022                    4181          STAX    D
                                4182
                                4183  *       SKIP BLANKS AND TABS
                                4184
063.101  053                    4185          DCX     H
063.102  043                    4186  UNL9    INX     H
063.103  176                    4187  UNL10   MOV     A,M
063.104  376 040                4188          CPI     ' '
063.106  312 102 063            4189          JE      UNL9            IS BLANK
063.111  376 011                4190          CPI     TAB
063.113  312 102 063            4191          JE      UNL9            IS TAB
063.116  311                    4192          RET                     EXIT
                                4193
063.117  011 105 116            4194  UNLA    DB      '       END Statement Missing',0
000.027                         4195  UNLAL   EQU     *-UNLA



                                4197  **      WBB - WRITE BINARY BUFFER.
                                4198  *
                                4199  *       WBB WRITES THE BINARY BUFFER TO THE DISK.
                                4200  *
                                4201  *       IF IT IS A REPLACEMENT FOR AN EXISIGING SECTOR, JUST WRITE IT.
                                4202  *
                                4203  *       IF NECESSARY, EXTEND THE FILE WITH GARBAGE UNTIL THE PROPER SECTOR
                                4204  *       IS REACHED.
                                4205  *
                                4206  *       ENTRY   NONE
                                4207  *       EXIT    NONE
                                4208  *       USES    NONE
                                4209
                                4210
063.146  315 054 031            4211  WBB     CALL    $SAVALL         SAVE REGISTESS
063.151  072 235 072            4212  WBB1    LDA     BINCSN          (A) = CURRENT SECTOR NUMBER
063.154  117                    4213          MOV     C,A
063.155  006 000                4214          MVI     B,0
063.157  076 000                4215          MVI     A,CN.BIN
063.161  377 047                4216          DB      SYSCALL,.POSIT          POSITION
063.163  322 342 063            4217          JNC     WBB2            OK
063.166  376 001                4218          CPI     EC.EOF
063.170  302 357 063            4219          JNE     BINERR          NOT EOF, SERIOUS ERROR
                                4220
                                4221  *       SHOULD NOT OCCOUR
                                4222
063.173  315 136 031            4223          CALL    $TYPTX
063.176  012 007 111            4224          DB      NL,BELL,'Internal Error #1'
063.221  012 124 150            4225          DB      NL,'This should not occur.'
063.250  103 157 156            4226          DB      'Contact HEATH Technical Correspondence for Assistance.'
```

```
063.336  212              4227          DB      ENL
063.337  303 211 043      4228          JMP     EXIT
                          4229
                          4230  *       GOT THERE. WRITE SECTOR
                          4231
063.342  001 000 001      4232  WBB2    LXI     B,256
063.345  021 336 073      4233          LXI     D,BINBFR
063.350  076 000          4234          MVI     A,CN.BIN
063.352  377 005          4235          DB      SYSCALL+.WRITE
063.354  322 047 031      4236          JNC     $RSTALL         EXIT IF OK
                          4237
                          4238  *       ERROR ON BINARY FILE
                          4239
063.357  041 202 072      4240  BINERR  LXI     H,BINFNAM-FB.NAM
063.362  303 135 067      4241          JMP     $FERROR         EXPLAIN ERROR
```

```
063.365                 4244          XTEXT   DTB


                        4246X **      $DTB - DELETE TRAILING BLANKS.
                        4247X *
                        4248X *       $DTB DELETES THE TRAILING BLANKS FROM A CODED LINE.
                        4249X *
                        4250X *       ENTRY   (HL) = LINE FWA
                        4251X *       EXIT    (A) = LENGTH OF RESULT (ENCLUDING 00 TERMINATOR BYTE)
                        4252X *       USES    A,F
                        4253X
                        4254X
063.365  325           4255X $DTB     PUSH    D               SAVE (DE)
063.366  124           4256X          MOV     D,H
063.367  135           4257X          MOV     E,L             (DE) = FWA
063.370  033           4258X          DCX     D               (DE) = FWA-1
063.371  176           4259X $DTB1    MOV     A,M
063.372  043           4260X          INX     H
063.373  247           4261X          ANA     A               FIND END OF LINE
063.374  302 371 063   4262X          JNZ     $DTB1
063.377  053           4263X          DCX     H               (HL) = ADDRESS OF TERMINATING ZERO BYTE
                        4264X
                        4265X *       GOT END OF LINE, DELETE TRAILING BLANKS
                        4266X
064.000  053           4267X $DTB2    DCX     H               BACKUP ONE CHARACTER
064.001  315 216 030   4268X          CALL    $CDEHL
064.004  312 015 064   4269X          JE      $DTB3           GONE PAST FRONT OF LINE, MUST BE ALL BLANKS
064.007  176           4270X          MOV     A,M
064.010  376 040       4271X          CPI     ' '
064.012  312 000 064   4272X          JE      $DTB2           GOT BLANK
                        4273X
                        4274X *       HAVE TRIMED LINE, COMPUTE LENGTH
                        4275X
064.015  043           4276X $DTB3    INX     H
064.016  066 000       4277X          MVI     M,0             TERMINATE LINE
064.020  175           4278X          MOV     A,L
064.021  223           4279X          SUB     E               (A) = LENGTH +1 (FOR 00 BYTE)
064.022  353           4280X          XCHG
064.023  043           4281X          INX     H               (HL) = LINE FWA
064.024  321           4282X          POP     D               RESTORE (DE)
064.025  311           4283X          RET
064.026               4284           XTEXT   CCO


                        4286X **      $CCO - CLEAR CONTROL-O
                        4287X *
                        4288X *       $CCO IS CALLED TO CLEAR THE EFFECT OF THE CTL-O CHARACTER.
                        4289X *
                        4290X *       ENTRY   NONE
                        4291X *       EXIT    NONE
                        4292X *       USES    NONE
                        4293X
```

```
                                4294X
064.026  315 054 031  4295X $CCO    CALL    $SAVALL         SAVE REGISTERS
064.031  076 004      4296X          MVI     A,I.CONFL
064.033  001 001 000  4297X          LXI     B,CO.FLG        CLEAR CO.FLG
064.036  377 006      4298X          DB      SYSCALL+.CONSL
064.040  303 047 031  4299X          JMP     $RSTALL         RESTORE REGISTERS AND RETURN
064.043               4300           XTEXT   MCU


                                4302X **       MCU - MAP LOWER CASE TO UPPER CASE.
                                4303X *
                                4304X *        MCU MAPS A LOWER CASE ALPHABETIC TO UPPER
                                4305X *        CASE.
                                4306X *
                                4307X *        ENTRY   (A) = CHARACTER
                                4308X *        EXIT    (A) = CHARACTER RESULT
                                4309X *        USES    A,F
                                4310X
                                4311X
064.043  376 141      4312X $MCU    CPI     'a'
064.045  330          4313X          RC                      NOT LOWER CASE
064.046  376 173      4314X          CPI     'z'+1
064.050  320          4315X          RNC                     NOT LOWER CASE
064.051  326 040      4316X          SUI     'a'-'A'
064.053  311          4317X          RET
064.054               4318           XTEXT   MLU


                                4320X **       MLU - MAP LOWER CASE LINE TO UPPER CASE.
                                4321X *
                                4322X *        MLU MAPS THE LOWER CASE ALPHABETICS IN A LINE TO UPPER CASE.
                                4323X *
                                4324X *        ENTRY   (HL) = LINE FWA
                                4325X *        EXIT    NONE
                                4326X *        USES    NONE
                                4327X
                                4328X
064.054  365          4329X $MLU    PUSH    PSW             SAVE (PSW)
064.055  345          4330X          PUSH    H               SAVE FWA
064.056  053          4331X          DCX     H               ANTICIPATE INX H
064.057  043          4332X $MLU1   INX     H
064.060  176          4333X          MOV     A,M             (A)= CHARACTER
064.061  315 043 064  4334X          CALL    $MCU            MAP CHAR TO UPPER
064.064  167          4335X          MOV     M,A
064.065  247          4336X          ANA     A
064.066  302 057 064  4337X          JNZ     $MLU1           MORE TO GO
064.071  341          4338X          POP     H               RESTORE (HL)
064.072  361          4339X          POP     PSW             RESTORE (PSW)
064.073  311          4340X          RET
064.074               4341           XTEXT   TYPLZ
```

```
                                4343X **      $TYPLZ - TYPE LINE UNTIL ZERO BYTE ENCOUNTERED.
                                4344X *
                                4345X *       NO NEW-LINE IS SENT.
                                4346X *
                                4347X *       ENTRY   (HL) = FWA OF TEXT
                                4348X *       EXIT    (HL) ADVANCED PAST ZERO BYTE
                                4349X *       USES    A,F,H,L
                                4350X
                                4351X
064.074  176                   4352X $TYPLZ  MOV     A,M
064.075  043                   4353X         INX     H
064.076  247                   4354X         ANA     A
064.077  310                   4355X         RZ                      ALL DONE
064.100  315 161 064           4356X         CALL    $WCHAR          WRITE LINE
064.103  303 074 064           4357X         JMP     $TYPLZ          DO MORE
064.106                        4358          XTEXT   HLIHL


                                4360X **      $HLIHL - LOAD HL INDIRECT THROUGH HL.
                                4361X *
                                4362X *       (HL) = ((HL))
                                4363X *
                                4364X *       ENTRY   NONE
                                4365X *       EXIT    NONE
                                4366X *       USES    A,H,L
                                4367X
030.211                         4368X $HLIHL  EQU     30211A          IN H17 ROM
064.106                         4369          XTEXT   CDEHL


                                4371X **      $CDEHL - COMPARE (DE) TO (HL)
                                4372X *
                                4373X *       $CDEHL COMPARES (DE) TO (HL) FOR EQUALITY.
                                4374X *
                                4375X *       ENTRY   NONE
                                4376X *       EXIT    'Z' SET IF (DE) = (HL)
                                4377X *       USES    A,F
                                4378X
                                4379X
030.216                         4380X $CDEHL  EQU     30216A          IN H17 ROM
064.106                         4381          XTEXT   CHL


                                4383X **      $CHL - COMPLEMENT (HL).
                                4384X *
                                4385X *       (HL) = -(HL)            TWO'S COMPLEMENT
                                4386X *
                                4387X *       ENTRY   NONE
                                4388X *       EXIT    NONE
                                4389X *       USES    A,F,H,L
```

```
                        4390X
                        4391X
030.224                 4392X $CHL    EQU     30224A          IN H17 ROM
064.106                 4393          XTEXT   DADA2


                        4395X **      $DADA. - ADD (0,A) TO (H,L)
                        4396X *
                        4397X *       ENTRY   NONE
                        4398X *       EXIT    (HL) = (HL) + (0A)
                        4399X *       USES    A,F,H,L
                        4400X
                        4401X
030.101                 4402X $DADA.  EQU     30101A          IN H17 ROM
064.106                 4403          XTEXT   SAVALL


                        4405X **      $RSTALL - RESTORE ALL REGISTERS.
                        4406X *
                        4407X *       $RSTALL RESTORES ALL THE REGISTERS OFF THE STACK, AND
                        4408X *       RETURNS TO THE PREVIOUS CALLER.
                        4409X *
                        4410X *       ENTRY   (SP) = PSW
                        4411X *               (SP+2) = BC
                        4412X *               (SP+4) = DE
                        4413X *               (SP+6) = HL
                        4414X *               (SP+8) = RET
                        4415X *       EXIT    TO *RET*, REGISTERS RESTORED
                        4416X *       USES    ALL
                        4417X
                        4418X
031.047                 4419X $RSTALL EQU     31047A          IN H17 ROM


                        4421X **      $SAVALL - SAVE ALL REGISTERS ON STACK.
                        4422X *
                        4423X *       $SAVALL SAVES ALL THE REGISTERS ON THE STACK.
                        4424X *
                        4425X *       ENTRY   NONE
                        4426X *       EXIT    (SP) = PSW
                        4427X *               (SP+2) = BC
                        4428X *               (SP+4) = DE
                        4429X *               (SP+6) = HL
                        4430X *       USES    H,L
                        4431X
                        4432X
031.054                 4433X $SAVALL EQU     31054A          IN H17 ROM
064.106                 4434          XTEXT   RTL
```

```
                                4436X **      $RTL - READ TEXT LINE.
                                4437X *
                                4438X *       $RTL READS A LINE FROM THE TERMINAL.
                                4439X *
                                4440X *       CHARACTER ARE ACCEPTED FROM THE TERMINAL, RUBOUT AND BACKSPACE
                                4441X *       CHARACTERS ARE PROCESSED. WHEN A CARRIAGE RETURN IS ENTERED,
                                4442X *       $RTL RETURNS.
                                4443X *
                                4444X *       ENTRY   (HL) = BUFFER FWA
                                4445X *       EXIT    'C' CLEAR IF OK
                                4446X *                DATA IN BUFFER
                                4447X *                (A) = TEXT LENGTH
                                4448X *                'C' SET IF CTL-D STRUCK
                                4449X *       USES    A,F
                                4450X
                                4451X
064.106  315 115 064           4452X $RTL.   CALL    $RTL            $RTL IN UPPER CASE
064.111  330                   4453X         RC                      CTL-D
064.112  303 054 064           4454X         JMP     $MLU            MAP LINE TO UPPER CASE
                                4455X
064.115                        4456X $RTL    EQU     *
064.115  345                   4457X         PUSH    H               SAVE FWA
064.116  315 153 064           4458X $RTL1   CALL    $RCHAR
064.121  376 004               4459X         CPI     CTLD
064.123  312 150 064           4460X         JE      $RTL2           CTL-D STRUCK
064.126  167                   4461X         MOV     M,A
064.127  043                   4462X         INX     H
064.130  376 012               4463X         CPI     NL
064.132  302 116 064           4464X         JNE     $RTL1
064.135  053                   4465X         DCX     H
064.136  066 000               4466X         MVI     M,0
064.140  043                   4467X         INX     H
                                4468X
                                4469X *       ALL DONE, COMPUTE LENGTH
                                4470X
064.141  353                   4471X         XCHG                    (DE) = LWA+1
064.142  343                   4472X         XTHL                    (HL) = FWA
064.143  173                   4473X         MOV     A,E
064.144  225                   4474X         SUB     L               (A) = LENGTH
064.145  247                   4475X         ANA     A               CLEAR CARRY
064.146  321                   4476X         POP     D               RESTORE (DE)
064.147  311                   4477X         RET
                                4478X
                                4479X *       CTL-D STRUCK
                                4480X
064.150  341                   4481X $RTL2   POP     H               (HL) = FWA
064.151  067                   4482X         STC
064.152  311                   4483X         RET
064.153                        4484          XTEXT   RCHAR
```

```
                                4486X **      $RCHAR - READ SINGLE CHARACTER FROM CONSOLE.
                                4487X *
                                4488X *       ENTRY   NONE
                                4489X *       EXIT    (A) = CHARACTER
                                4490X *       USES    A,F
                                4491X
                                4492X
064.153  377 001               4493X $RCHAR  DB      SYSCALL,.SCIN
064.155  332 153 064           4494X         JC      $RCHAR          NOT READY
064.160  311                   4495X         RET
                                4496X
064.161  377 002               4497X $WCHAR  DB      SYSCALL,.SCOUT
064.163  311                   4498X         RET
064.164                        4499          XTEXT   UDD



                                4501X **      $UDD - UNPACK DECIMAL DIGITS.
                                4502X *
                                4503X *       UDD CONVERTS A 16 BIT VALUE INTO A SPECIFIED NUMBER OF
                                4504X *       DECIMAL DIGITS. THE RESULT IS ZERO FILLED.
                                4505X *
                                4506X *       ENTRY   (B,C) = ADDRESS VALUE
                                4507X *               (A) = DIGIT COUNT
                                4508X *               (H,L) = MEMORY ADDRESS
                                4509X *       EXIT    (HL) = (HL) + (A)
                                4510X *       USES    ALL
                                4511X
                                4512X
031.157                         4513X $UDD    EQU     31157A          IN H17 ROM
064.164                         4514          XTEXT   MOVEL



                                4516X **      $MOVEL - MOVE DATA
                                4517X *
                                4518X *       $MOVEL MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
                                4519X *       IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
                                4520X *       FIRST TO LAST.
                                4521X *
                                4522X *       IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM
                                4523X *       LAST TO FIRST.
                                4524X *
                                4525X *       THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.
                                4526X *
                                4527X *       CALL    $MOVEL
                                4528X *       DW      COUNT
                                4529X *       DW      FROM
                                4530X *       DW      TO
                                4531X *
                                4532X *       ENTRY   ((SP)) = RET
                                4533X *               (RET+0) = COUNT (WORD VALUE)
                                4534X *               (RET+2) = FROM
                                4535X *               (RET+4) = TO
```

```
                        4536X *         EXIT    TO (RET+6)
                        4537X *                 (DE) = ADDRESS OF NEXT FROM BYTE
                        4538X *                 (HL) = ADDRESS OF NEXT *TO* BYTE
                        4539X *                 'C' CLEAR
                        4540X *         USES    ALL
                        4541X
                        4542X
064.164  341           4543X $MOVEL  POP     H               (HL) = RET
064.165  116           4544X         MOV     C,M
064.166  043           4545X         INX     H
064.167  106           4546X         MOV     B,M             (BC) = COUNT
064.170  043           4547X         INX     H
064.171  136           4548X         MOV     E,M
064.172  043           4549X         INX     H
064.173  126           4550X         MOV     D,M             (DE) = FROM
064.174  043           4551X         INX     H
064.175  325           4552X         PUSH    D               ((SP)) = FROM
064.176  136           4553X         MOV     E,M
064.177  043           4554X         INX     H
064.200  126           4555X         MOV     D,M             (DE) = TO
064.201  043           4556X         INX     H
064.202  343           4557X         XTHL                    ((SP)) = RET, (HL) = FROM
064.203  353           4558X         XCHG                    (DE) = FROM , (HL) = TO
064.204  303 252 030   4559X         JMP     $MOVE           MOVE IT
064.207               4560          XTEXT   CPF


                        4562X **        $CPF - COPY FILE NAME
                        4563X *
                        4564X *         $CPF COPIES A FILE NAME FROM ONE LOCATION TO ANOTHER.
                        4565X *
                        4566X *         THE CHARACTERS ARE COPIED UNTIL A DELIMITER (',', ' ', '=', OR 00)
                        4567X *         IS FOUND.
                        4568X *
                        4569X *         THE FILENAME IS THEN TERMINATED WITH A 00 BYTE.
                        4570X *
                        4571X *         ENTRY   (DE) = FROM ADDRESS
                        4572X *                 (HL) = TO ADDRESS
                        4573X *         EXIT    'C' CLEAR IF OK
                        4574X *                  (DE) = ADVANCED PAST NAME AND DELIMITER
                        4575X *                  (HL) POINTS TO 00 BYTE OF DESTINATION
                        4576X *                  (A) = DELIMITER
                        4577X *                 'C' SET IF ERROR
                        4578X *         USES    ALL
                        4579X
                        4580X
064.207  006 022       4581X $CPF    MVI     B,FB.NAML+1             SET MAX LENGTH
064.211  032           4582X $CPF1   LDAX    D
064.212  247           4583X         ANA     A
064.213  312 246 064   4584X         JZ      $CPF2           END
064.216  023           4585X         INX     D
064.217  376 054       4586X         CPI     ','
064.221  312 246 064   4587X         JE      $CPF2
064.224  376 075       4588X         CPI     '='
```

2-83 Change -to limit length to correct size

MVI B,FB.NAML

→ next source

```
064.226  312 246 064  4589X          JE      $CPF2
064.231  376 040      4590X          CPI     ' '
064.233  312 246 064  4591X          JE      $CPF2           IS BLANK
064.236  167          4592X          MOV     M,A             COPY
064.237  043          4593X          INX     H
064.240  005          4594X          DCR     B
064.241  302 211 064  4595X          JNZ     $CPF1           IF MORE GO TO
064.244  067          4596X          STC                     OVERFLOW OF AREA
064.245  311          4597X          RET                                    DCX  H  sb terminator
                      4598X
                      4599X *        DONE.
                      4600X
064.246  066 000      4601X $CPF2    MVI     M,0             TERMINATE
064.250  311          4602X          RET
064.251               4603           XTEXT   INDL


                      4605X **       $INDL - INDEXED LOAD.
                      4606X *
                      4607X *        $INDL LOADS DE WITH THE TWO BYTES AT (HL)+DISPLACMENT
                      4608X *
                      4609X *        THIS ACTS AS AN INDEXED FULL WORD LOAD.
                      4610X *
                      4611X *        (DE) = ( (HL) + DSPLACEMENT )
                      4612X *
                      4613X *        ENTRY   ((RET)) = DISPLACMENT (FULL WORD)
                      4614X *                (HL) = TABLE ADDRESS
                      4615X *        EXIT    TO (RET+2)
                      4616X *        USES    A,F,D,E
                      4617X
                      4618X
030.234               4619X $INDL    EQU     30234A          IN H17 ROM
064.251               4620           XTEXT   MOVE


                      4622X **       $MOVE - MOVE DATA
                      4623X *
                      4624X *        $MOVE MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
                      4625X *        IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
                      4626X *        FIRST TO LAST.
                      4627X *
                      4628X *        IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM
                      4629X *        LAST TO FIRST.
                      4630X *
                      4631X *        THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.
                      4632X *
                      4633X *        ENTRY   (BC) = COUNT
                      4634X *                (DE) = FROM
                      4635X *                (HL) = TO
                      4636X *        EXIT    MOVED
                      4637X *                (DE) = ADDRESS OF NEXT FROM BYTE
                      4638X *                (HL) = ADDRESS OF NEXT *TO* BYTE
```

```
                        4639X *                 'C' CLEAR
                        4640X *         USES    ALL
                        4641X
                        4642X
030.252                 4643X $MOVE     EQU     30252A          IN H17 ROM
064.251                 4644            XTEXT   CRLF


                        4646X **        $CRLF - TYPE CARRIAGE RETURN/ LINE FEED
                        4647X *
                        4648X *         $CRLF IS USED TO GENERATE PADDED CRLF'S.
                        4649X *
                        4650X *         ENTRY   NONE
                        4651X *         EXIT    (A) = 0
                        4652X *         USES    A,F
                        4653X
                        4654X
064.251 076 012         4655X $CRLF     MVI     A,NL
064.253 377 002         4656X           DB      SYSCALL,.SCOUT
064.255 257             4657X           XRA     A
064.256 311             4658X           RET
064.257                 4659            XTEXT   DADA


                        4661X **        $DADA - PERFORM (H,L) = (H,L) + (0,A)
                        4662X *
                        4663X *         ENTRY   (H,L) = BEFORE VALUE
                        4664X *                 (A) = BEFORE VALUE
                        4665X *         EXIT    (H,L) = (H,L) + (0,A)
                        4666X *                 'C' SET IF OVERFLOW
                        4667X *         USES    F,H,L
                        4668X
                        4669X
030,072                 4670X $DADA     EQU     30072A          IN H17 ROM
064.257                 4671            XTEXT   UOD


                        4673X **        $UOD - UNPACK OCTAL DIGITS.
                        4674X *
                        4675X *         UOD CONVERTS A SINGLE BYTE INTO 3 OCTAL DIGITS., ZERO FILL
                        4676X *
                        4677X *         ENTRY   (A) = BYTE VALUE
                        4678X *                 (H,L) = ADDRESS OF 3 BYTE AREA FOR DIGITS
                        4679X *         EXIT    (H,L) = (H,L)+3
                        4680X *         USES    A,H,L
                        4681X
                        4682X
064.257 305             4683X $UOD      PUSH    B
064.260 006 003         4684X           MVI     B,3             (B) = LOOP COUNT
064.262 247             4685X           ANA     A               CLEAR CARRY
```

```
                        4686X
064.263  027            4687X UOD1   RAL
064.264  027            4688X        RAL
064.265  027            4689X        RAL
064.266  365            4690X        PUSH    PSW             SAVE VALUE
064.267  346 007        4691X        ANI     7
064.271  306 060        4692X        ADI     '0'
064.273  167            4693X        MOV     M,A             STORE DIGIT
064.274  043            4694X        INX     H
064.275  361            4695X        POP     PSW             RESTORE VALUE
064.276  005            4696X        DCR     B
064.277  302 263 064     4697X        JNZ     UOD1            IF MORE TO GO
064.302  301            4698X        POP     B               RESTORE (B,C)
064.303  311            4699X        RET                       EXIT
064.304                 4700         XTEXT   DU66


                        4702X **     $DU66 - UNSIGNED 16 / 16 DIVIDE.
                        4703X *
                        4704X *      (HL) = (BC)/(DE)
                        4705X *
                        4706X *      ENTRY   (BC), (DE) PRESET
                        4707X *      EXIT    (HL) = RESULT
                        4708X *                (DE) = REMAINDER
                        4709X *      USES    ALL
                        4710X
                        4711X
030.106                 4712X $DU66  EQU     30106A          IN H17 ROM
064.304                 4713         XTEXT   MU66


                        4715X **     $MU66 - UNSIGNED 16X16 MULTIPLY.
                        4716X *
                        4717X *      ENTRY (BC) = MULTIPLICAND
                        4718X *             (DE) = MULTIPLIER
                        4719X *      EXIT    (HL) = RESULT
                        4720X *               'Z' SET IF NOT OVERFLOW
                        4721X *      USES    ALL
                        4722X
                        4723X
030.337                 4724X $MU66  EQU     30337A          IN H17 ROM
064.304                 4725         XTEXT   TBLS
```

```
                                4727X **      $TBLS - TABLE SEARCH
                                4728X *
                                4729X *       TABLE FORMAT
                                4730X *
                                4731X *       DB      KEY1,VAL1,
                                4732X *       .       .
                                4733X *       .       .
                                4734X *       DB      KEYN,VALN
                                4735X *       DB      0
                                4736X *
                                4737X *       ENTRY   (A) = PATTERN
                                4738X *               (H,L) = TABLE FWA
                                4739X *       EXIT    (A) = PATTERN IF FOUND
                                4740X *               'Z' SET IF FOUND
                                4741X *               'Z' CLEAR IF NOT FOUND OR PATTERN=0     /78.10.GC/
                                4742X *       USES    A,F,H,L
                                4743X
                                4744X
064,304  305                   4745X $TBLS   PUSH    B
064,305  376 000               4746X         CPI     0                                         /78.10.GC/
064,307  312 331 064           4747X         JZ      TBL2                                      /78.10.GC/
064,312  107                   4748X         MOV     B,A
064,313  176                   4749X TBL1    MOV     A,M             (A) = CHARACTER
064,314  043                   4750X         INX     H
064,315  270                   4751X         CMP     B
064,316  312 333 064           4752X         JZ      TBL3            IF MATCH
064,321  247                   4753X         ANA     A
064,322  043                   4754X         INX     H               SKIP PAST
064,323  302 313 064           4755X         JNZ     TBL1            IF NOT END OF TABLE
064,326  053                   4756X         DCX     H
064,327  053                   4757X         DCX     H
064,330  257                   4758X         XRA     A               SET TO ZERO FOR OLD USERS  /78.10.GC/
064,331  376 001               4759X TBL2    CPI     1               CLEAR ZERO                 /78.10.GC/
                                4760X
                                4761X *       DONE
                                4762X
064,333  301                   4763X TBL3    POP     B
064,334  311                   4764X         RET
064,335                        4765          XTEXT   TBRA



                                4767X **      $TBRA - BRANCH RELATIVE THOUGH TABLE.
                                4768X *
                                4769X *       $TBRA USES THE SUPPLIED INDEX TO SELECT A BYTE FROM THE
                                4770X *       JUMP TABLE. THE CONTENTS OF THIS BYTE ARE ADDED TO THE
                                4771X *       ADDRESS OF THE BYTE, YEILDING THE PROCESSOR ADDRESS.
                                4772X *
                                4773X *       CALL    $TBRA
                                4774X *       DB      LAB1-*          INDEX = 0 FOR LAB1
                                4775X *       DB      LAB2-*          INDEX = 1 FOR LAB2
                                4776X *       DB      LABN-*          INDEX = N-1 FOR LABN
                                4777X *
                                4778X *       ENTRY   (A) = INDEX
                                4779X *               (RET) = TABLE FWA
```

```
                        4780X *       EXIT    TO COMPUTED ADDRESS
                        4781X *       USES    F,H,L
                        4782X
                        4783X
     031.076            4784X $TBRA   EQU     31076A          IN H17 ROM
     064.335            4785          XTEXT   TYPT2


                        4787X **      $TYPTX - TYPE TEXT.
                        4788X *
                        4789X *       $TYPTX IS CALLED TO TYPE A BLOCK OF TEXT ON THE SYSTEM CONSOLE.
                        4790X *
                        4791X *       IMBEDDED ZERO BYTES INDICATE A CARRIAGE RETURN LINE FEED,
                        4792X *       A BYTE WITH THE 200Q BIT SET IS THE LAST BYTE IN THE MESSAGE.
                        4793X *
                        4794X *       ENTRY   (RET) = TEXT
                        4795X *       EXIT    TO (RET+LENGTH)
                        4796X *       USES    A,F
                        4797X
                        4798X
     031.136            4799X $TYPTX  EQU     31136A          IN H17 ROM
                        4800X
     031.144            4801X $TYPTX. EQU     31144A          IN H17 ROM
     064.335            4802          XTEXT   ZEROS


                        4804X **      8 CONSTANT ZERO BYTES.
                        4805X
     031.320            4806X $ZEROS  EQU     31320A          IN H17 ROM
     064.335            4807          XTEXT   FOPE


                        4809X **      $FOPEx - OPEN FILE BLOCK FOR I/O
                        4810X *
                        4811X *       $FOPEx IS CALLED BEFORE ANY I/O IS DONE VIA A
                        4812X *       FILE BLOCK. $FOPEx SETS UP THE FILE BLOCK, AND OPENS
                        4813X *       THE FILE VIA *HDOS*.
                        4814X *
                        4815X *       ENTRY   (DE) = ADDRESS OF DEFAULT BLOCK
                        4816X *               (HL) = ADDRESS OF FILE BLOCK
                        4817X *       EXIT    TO $FERROR IF ERROR
                        4818X *               TO CALLER IF OK
                        4819X *       USES    A;F;B;C;D;E
                        4820X
                        4821X
     064.335  315 362 064  4822X $FOPER  CALL    $FOPER.
     064.340  320         4823X          RNC
     064.341  303 135 067  4824X          JMP     $FERROR         IN ERROR
                        4825X
     064.344  315 365 064  4826X $FOPEW  CALL    $FOPEW.
```

```
064.347  320            4827X          RNC
064.350  303 135 067    4828X          JMP     $FERROR         IN ERROR
                        4829X
064.353  315 370 064    4830X $FOPEU   CALL    $FOPEU.
064.356  320            4831X          RNC
064.357  303 135 067    4832X          JMP     $FERROR         IN ERROR
                        4833X
                        4834X
064.362  076 002        4835X $FOPER.  MVI     A,FT.OR         FILE TYPE OF OPEN FOR READ
064.364  001            4836X          DB      001Q            LXI,B TO SKIP NEXT MVI
064.365  076 004        4837X $FOPEW.  MVI     A,FT.OW         OPEN FOR WRITE
064.367  001            4838X          DB      001Q            LXI,B TO SKIP NEXT MIV
064.370  076 006        4839X $FOPEU.  MVI     A,FT.OR+FT.OW
                        4840X
                        4841X *        (A) = FILE FLAGS
                        4842X
064.372  345            4843X          PUSH    H               SAVE FILE BLOCK ADDRESS
064.373  365            4844X          PUSH    PSW             SAVE NEW FLAGS
000.000                 4845X          ERRNZ   FB.CHA
064.374  106            4846X          MOV     B,M             (B) = CHANNEL NUMBER
064.375  305            4847X          PUSH    B               SAVE HANNEL NUMBER
000.000                 4848X          ERRNZ   FB.FLG-FB.CHA-1
064.376  043            4849X          INX     H
064.377  117            4850X          MOV     C,A             (C) = NEW FILE FLAGS
065.000  176            4851X          MOV     A,M             (A) = CURRENT TYPE
065.001  247            4852X          ANA     A
065.002  171            4853X          MOV     A,C             (A) = NEW FLAGS TO BE SET
065.003  312 015 065    4854X          JZ      $FOPE1          NOT ALREADY OPEN
                        4855X
                        4856X *        ALREADY OPEN, SQUACK
                        4857X
065.006  301            4858X          POP     B               RESTORE (BC)
065.007  361            4859X          POP     PSW             DISCARD NEW FLAGS
065.010  341            4860X          POP     H               (HL) = FB ADDRESS
065.011  076 031        4861X          MVI     A,EC.FAO        FILE ALREADY OPEN
065.013  067            4862X          STC
065.014  311            4863X          RET
                        4864X
000.000                 4865X          ERRNZ   FB.FWA-FB.FLG-1
065.015  043            4866X $FOPE1   INX     H               (HL) = *FB,FWA
065.016  116            4867X          MOV     C,M
065.017  043            4868X          INX     H
065.020  106            4869X          MOV     B,M             (BC) = FB.FWA
065.021  043            4870X          INX     H
000.000                 4871X          ERRNZ   FB.PTR-FB.FWA-2
065.022  161            4872X          MOV     M,C             SET FB.PTR = FB.FWA
065.023  043            4873X          INX     H
065.024  160            4874X          MOV     M,B
065.025  043            4875X          INX     H
000.000                 4876X          ERRNZ   FB.LIM-FB.PTR-2
065.026  161            4877X          MOV     M,C             SET FB.LIM = FB.FWA
065.027  043            4878X          INX     H
065.030  160            4879X          MOV     M,B
065.031  043            4880X          INX     H
000.000                 4881X          ERRNZ   FB.NAM-FB.LIM-4
065.032  043            4882X          INX     H
```

```
065.033  043            4883X          INX     H               (HL) = $FB.NAM
                        4884X
                        4885X *        FILE BLOCK POINTERS SETUP. OPEN FILE
                        4886X
065.034  345            4887X          PUSH    H               SAVE NEW ADDRESS FOR NAME
065.035  041 066 065    4888X          LXI     H,$FOPEB
065.040  247            4889X          ANA     A                                       /78.10.BC/
065.041  312 050 065    4890X          JZ      $FOPE2
000.000                 4891X          ERRNZ   .EXIT
065.044  315 304 064    4892X          CALL    $TBLS           FIND CODE
065.047  176            4893X          MOV     A,M
065.050  062 056 065    4894X $FOPE2   STA     $FOPEA          SET SYSCALL CODE
065.053  341            4895X          POP     H               (HL) = $FB.NAM
065.054  361            4896X          POP     PSW             (A) = CHANNEL NUMBER
065.055  377 000         4897X          DB      SYSCALL,.EXIT
065.056                 4898X $FOPEA   EQU     *-1             SYSCALL CODE
065.057  321            4899X          POP     D               (D) = NEW FLAG
065.060  341            4900X          POP     H               (HL) = FILE BLOCK ADDRESS
065.061  330            4901X          RC                      EXIT IF ERROR
065.062  043            4902X          INX     H
000.000                 4903X          ERRNZ   FB.FLG-1
065.063  162            4904X          MOV     M,D             SET NEW FLAGS
065.064  053            4905X          DCX     H               RESTORE (HL)
065.065  311            4906X          RET
                        4907X
065.066  002 042        4908X $FOPEB   DB      FT.OR,.OPENR    TABLE OF SYSCALL CODES
065.070  004 043        4909X          DB      FT.OW,.OPENW
065.072  006 044        4910X          DB      FT.OR+FT.OW,.OPENU
065.074  000            4911X          DB      0               SHOULD NOT OCCUR
065.075                 4912           XTEXT   FCLO



                        4914X **       $FCLO - CLOSE FILE BLOCK.
                        4915X *
                        4916X *        $FCLO IS CALLED TO TERMINATE PROCESSING THROUGH A FILE
                        4917X *        BLOCK.
                        4918X *
                        4919X *        ENTRY   (HL) = FILE BLOCK ADDRESS
                        4920X *        EXIT    TO $FERROR IF ERROR
                        4921X *                 TO CALLER IF OK
                        4922X *        USES    A,F,B,C,D,E
                        4923X
                        4924X
065.075  315 104 065    4925X $FCLO    CALL    $FCLO.
065.100  320            4926X          RNC                     NO ERROR
065.101  303 135 067    4927X          JMP     $FERROR
                        4928X
065.104  345            4929X $FCLO.   PUSH    H               SAVE FILE BLOCK ADDRESS
000.000                 4930X          ERRNZ   FB.FLG-1
065.105  043            4931X          INX     H               (HL) = $FB.FLG
065.106  176            4932X          MOV     A,M
065.107  066 000        4933X          MVI     M,0             CLEAR FLAG
065.111  247            4934X          ANA     A
065.112  312 200 065    4935X          JZ      $FCL04          FILE NOT OPEN
```

```
065.115  346 004       4936X          ANI     FT.OW
065.117  312 172 065   4937X          JZ      $FCLO3          NO WRITING, NO FLUSHING NEEDED
                       4938X
                       4939X *        WAS OPEN FOR WRITE, SEE IF NEED FLUSH THE LAST SECTOR
                       4940X
065.122  315 234 030   4941X          CALL    $INDL
065.125  003 000       4942X          DW      FB.PTR-FB.FLG
065.127  325           4943X          PUSH    D               SAVE (FB.PTR)
065.130  315 234 030   4944X          CALL    $INDL           (DE) = (FB.FWA)
065.133  001 000       4945X          DW      FB.FWA-FB.FLG
065.135  341           4946X          POP     H               (HL) = (FB.PTR)
065.136  175           4947X          MOV     A,L
065.137  223           4948X          SUB     E
065.140  117           4949X          MOV     C,A
065.141  174           4950X          MOV     A,H
065.142  232           4951X          SBB     D
065.143  107           4952X          MOV     B,A             (BC) = AMOUNT IN BLOCK
065.144  261           4953X          ORA     C
065.145  312 172 065   4954X          JZ      $FCLO3          NONE TO FLUSH
                       4955X
                       4956X *        NEED TO FLUSH BUFFER
                       4957X *
                       4958X *        (BC) = DATA AMOUNT
                       4959X *        (DE) = FWA
                       4960X *        (HL) = LWA+1
                       4961X
065.150  171           4962X          MOV     A,C
065.151  247           4963X          ANA     A
065.152  312 165 065   4964X          JZ      $FCLO2          DONT HAVE PARTIAL SECTOR
                       4965X
                       4966X *        ZERO FILL PARTIAL SECTOR
                       4967X
065.155  066 000       4968X $FCLO1   MVI     M,0
065.157  043           4969X          INX     H
065.160  014           4970X          INR     C
065.161  302 155 065   4971X          JNZ     $FCLO1
065.164  004           4972X          INR     B               COUNT ANOTHER FULL SECTOR
065.165  341           4973X $FCLO2   POP     H               (HL) = FB.FWA
065.166  176           4974X          MOV     A,M             (A) = CHANNEL NUMBER
000.000                4975X          ERRNZ   FB.CHA
065.167  345           4976X          PUSH    H
065.170  377 005       4977X          DB      SYSCALL+.WRITE+FLUSH
                       4978X
                       4979X *        READY TO CLOSE FILE.
                       4980X *
                       4981X *        'C' SET IF ERROR
                       4982X *          (A) = ERROR CODE
                       4983X
065.172  341           4984X $FCLO3   POP     H               (HL) = FILE BLOCK ADDRESS
065.173  330           4985X          RC                      ERROR
000.000                4986X          ERRNZ   FB.CHA
065.174  176           4987X          MOV     A,M             (A) = CHANNEL NUMBER
065.175  345           4988X          PUSH    H
065.176  377 046       4989X          DB      SYSCALL+.CLOSE  CLOSE CHANNEL
065.200  341           4990X $FCLO4   POP     H               (HL) = FILE BLOCK ADDRESS
065.201  311           4991X          RET
```

```
065.202              XTEXT   FCLEAR


                 **       $FCLEAR - CLEAR FILE BLOCK.
                 *
                 *        $FCLEAR CLEARS OUT A FILE BLOCK BY SETTING THE POINTERS TO
                 *        EMPTY, AND CLEARING ANY ERROR OR EOF FLAGS.
                 *
                 *        THE DISK (OR WHATEVER) FILE IS NOT POSITIONED, READ, WRITEN
                 *        OPENED OR CLOSED.
                 *
                 *        ENTRY   (HL) = FB ADDRESS
                 *        EXIT    NONE
                 *        USES    A,F,B,C


065.202          $FCLEAR EQU     *
065.202  345             PUSH    H                       SAVE FILE BLOCK ADDRESS
000.000                  ERRNZ   FB.FLG-FB.CHA-1
065.203  043             INX     H
000.000                  ERRNZ   FB.FWA-FB.FLG-1
065.204  043             INX     H                       (HL) = #FB.FWA
065.205  116             MOV     C,M
065.206  043             INX     H
065.207  106             MOV     B,M                     (BC) = FB.FWA
065.210  043             INX     H
000.000                  ERRNZ   FB.PTR-FB.FWA-2
065.211  161             MOV     M,C                     SET FB.PTR = FB.FWA
065.212  043             INX     H
065.213  160             MOV     M,B
065.214  043             INX     H
000.000                  ERRNZ   FB.LIM-FB.PTR-2
065.215  161             MOV     M,C                     SET FB.LIM = FB.FWA
065.216  043             INX     H
065.217  160             MOV     M,B
065.220  341             POP     H                       (HL) = FB FWA
065.221  311             RET
065.222                  XTEXT   FREAL


                 **       $FREAL - READ BYTES FROM FILE BUFFER.
                 *
                 *        $FREAL IS CALLED TO READ A NUMBER OF BYTES FROM A FILE BUFFER.
                 *
                 *        ENTRY   (BC) = BYTE COUNT
                 *                (DE) = FWA FOR BYTES
                 *                (HL) = ADDRESS OF FILE BUFFER
                 *        EXIT    TO *FERROR* IF ERROR
                 *                TO CALLER IF OK
                 *                 (BC) = UNREAD BYTE COUNT (ONLY IF EOF)
                 *                 (DE) = ADDRESS OF FIRST UNUSED BYTE
                 *                 'C' SET IF EOF DURING READ
```

```
                          5042X *       USES    A,F,B,C,D,E
                          5043X
                          5044X
065.222 315 235 065       5045X $FREAL  CALL    $FREAL.
065.225 320               5046X         RNC                     RETURN IF OK
065.226 376 001           5047X         CPI     EC.EOF
065.230 302 135 067       5048X         JNE     $FERROR         ERROR IS NOT EOF
065.233 067               5049X         STC
065.234 311               5050X         RET                     ERROR IS SIMPLY EOF
                          5051X
                          5052X
065.235                   5053X $FREAL. EQU     *
065.235 013               5054X         DCX     B               (BC) = COUNT NOT ENCLUDING 00 BYTE
065.236 257               5055X         XRA     A
065.237 062 134 067       5056X         STA     EOFFLG          CLEAR EOF FLAG
065.242 345               5057X         PUSH    H
065.243 315 360 066       5058X         CALL    CBT             COPY BUFFER POINTERS TO TEMP CELLS
                          5059X
                          5060X *       COPY DATA FROM BUFFER TO TARGET
                          5061X
065.246 325               5062X $REAL2  PUSH    D               SAVE TARGET ADDRESS
065.247 072 123 067       5063X         LDA     T.FLG
065.252 346 002           5064X         ANI     FT.OR
065.254 076 011           5065X         MVI     A,EC.FNO
065.256 067               5066X         STC                     ASSUME FILE NOT OPEN
065.257 312 013 066       5067X         JZ      $REAL8          ERROR
065.262 170               5068X         MOV     A,B
065.263 261               5069X         ORA     C
065.264 312 013 066       5070X         JZ      $REAL8          ALL DONE
                          5071X
                          5072X *       COMPUTE MIN( DATA IN BUFFER, DATA REQUESTED)
                          5073X
065.267 052 126 067       5074X $REAL3  LHLD    T.PTR
065.272 353               5075X         XCHG                    (DE) = (FB.PTR) = ADDRESS OF DATA
065.273 052 130 067       5076X         LHLD    T.LIM           (HL) = LIMIT ADDRESS
065.276 175               5077X         MOV     A,L
065.277 223               5078X         SUB     E
065.300 157               5079X         MOV     L,A
065.301 174               5080X         MOV     A,H
065.302 232               5081X         SBB     D
065.303 147               5082X         MOV     H,A             (HL) = NUMBER OF BYTES IN BUFFER
065.304 171               5083X         MOV     A,C
065.305 225               5084X         SUB     L               COMPARE TO REQUESTED COUNT
065.306 170               5085X         MOV     A,B
065.307 234               5086X         SBB     H
065.310 322 315 065       5087X         JNC     $REAL4          LESS THAN REQUESTED COUNT
065.313 140               5088X         MOV     H,B
065.314 151               5089X         MOV     L,C             DONT TRANSFER MORE THAN LIMIT
065.315 174               5090X $REAL4  MOV     A,H
065.316 265               5091X         ORA     L
065.317 302 333 065       5092X         JNZ     $REAL6          SOME IN BUFFER
                          5093X
                          5094X *       BUFFER IS EMPTY, RE-FILL IT
                          5095X
065.322 315 040 067       5096X         CALL    $FFB            FILL FILE BUFFER
065.325 332 013 066       5097X         JC      $REAL8          ERROR CONDITION
```

```
065.330  303 267 065  5098X          JMP     $REAL3          COUNT THE DATA
                      5099X
                      5100X *        GOT THE DATA. MOVE IT FROM BUFFER TO TARGET
                      5101X *
                      5102X *        (BC) = LIMIT COUNT
                      5103X *        (DE) = FROM
                      5104X *        (HL) = COUNT
                      5105X *        ((SP)) = TO
                      5106X
065.333  171          5107X $REAL6   MOV     A,C
065.334  225          5108X          SUB     L
065.335  117          5109X          MOV     C,A
065.336  170          5110X          MOV     A,B
065.337  234          5111X          SBB     H
065.340  107          5112X          MOV     B,A             REMOVE BYTES ABOUT TO BE MOVED FROM REQUEST COUNT
065.341  305          5113X          PUSH    B
065.342  343          5114X          XTHL                    (HL) = REMAINING REQUEST COUNT
065.343  301          5115X          POP     B               (BC) = COUNT FOR THIS COPY
065.344  343          5116X          XTHL                    (HL) = TARGET ADDR, ((SP)) = REMAINING REQ. COUNT
065.345  032          5117X $REAL7   LDAX    D
065.346  023          5118X          INX     D
065.347  167          5119X          MOV     M,A
065.350  043          5120X          INX     H
065.351  247          5121X          ANA     A               SEE IF 00 BYTE
065.352  302 361 065  5122X          JNZ     $REL7.3         NOT 00
                      5123X
                      5124X *        IS 00 BYTE. IGNORE IT
                      5125X
065.355  343          5126X          XTHL
065.356  043          5127X          INX     H               ADD ONE TO UNREQUITED COUNT
065.357  343          5128X          XTHL
065.360  053          5129X          DCX     H               BACKSPACE OVER CHARACTER
065.361  013          5130X $REL7.3  DCX     B
065.362  376 012      5131X          CPI     NL
065.364  312 004 066  5132X          JE      $REL7.5         IS END OF LINE
065.367  170          5133X          MOV     A,B
065.370  261          5134X          ORA     C
065.371  302 345 065  5135X          JNZ     $REAL7          MORE TO GO
065.374  353          5136X          XCHG
065.375  042 126 067  5137X          SHLD    T.PTR           UPDATE POINTER
066.000  301          5138X          POP     B               (BC) = REMAINING COUNT
066.001  303 246 065  5139X          JMP     $REAL2          SEE IF MORE IN BUFFER
                      5140X
                      5141X *        END OF CODED LINE
                      5142X
066.004  353          5143X $REL7.5  XCHG
066.005  033          5144X          DCX     D               BACK OVER NL CHARACTER
066.006  042 126 067  5145X          SHLD    T.PTR           UPDATE POINTER
066.011  301          5146X          POP     B               (BC) = REMAINING COUNT
066.012  325          5147X          PUSH    D               SAVE TARGET LWA
                      5148X
                      5149X *        READ COMPLETE.
                      5150X *
                      5151X *        (PSW) = COMPLETION FLAGS
                      5152X
066.013  321          5153X $REAL8   POP     D               RESTORE TARGET ADDRESS
```

```
066.014  365            5154X          PUSH    PSW             SAVE RETURN CODE
066.015  257            5155X          XRA     A
066.016  022            5156X          STAX    D               FLAG END OF LINE
066.017  361            5157X          POP     PSW             RESTORE RESULT FLAGS
066.020  023            5158X          INX     D               POINT TO NEXT FREE
066.021  341            5159X $REAL9   POP     H
066.022  303 006 067     5160X          JMP     CTB             COPY TEMP POINTERS BACK TO BLOCK, EXIT
066.025                 5161           XTEXT   FWRIL


                        5163X **       $FWRIL - WRITE LINE FROM FILE BUFFER.
                        5164X *
                        5165X *        $FWRIL IS CALLED TO WRITE A LINE TO A FILE BUFFER.
                        5166X *
                        5167X *        ENTRY   (DE) = FWA FOR BYTES
                        5168X *                (HL) = ADDRESS OF FILE BUFFER
                        5169X *        EXIT    TO *FERROR* IF ERROR
                        5170X *                TO CALLER IF OK
                        5171X *                 (DE) = ADDRESS OF FIRST UNWRITTEN BYTE
                        5172X *        USES    A,F,B,C,D,E
                        5173X
                        5174X
066.025  315 034 066     5175X $FWRIL   CALL    $FWRIL.
066.030  320             5176X          RNC                     RETURN IF OK
066.031  303 135 067     5177X          JMP     $FERROR         ERROR
                        5178X
                        5179X *        SCAN FOR END OF LINE
                        5180X
066.034  325             5181X $FWRIL.  PUSH    D               SAVE LINE POINTER
066.035  001 377 377     5182X          LXI     B,-1            (BC) = COUNT
066.040  032             5183X $FWRIL1  LDAX    D
066.041  023             5184X          INX     D
066.042  003             5185X          INX     B
066.043  247             5186X          ANA     A
066.044  302 040 066     5187X          JNZ     $FWRIL1         MORE TO GO
066.047  321             5188X          POP     D
066.050  315 072 066     5189X          CALL    $FWRIB          WRITE BYTES
066.053  330             5190X          RC                      ERROR
                        5191X
                        5192X *        WRITE 'NL' CHARACTER
                        5193X
066.054  023             5194X          INX     D
066.055  325             5195X          PUSH    D
066.056  001 001 000     5196X          LXI     B,1
066.061  021 071 066     5197X          LXI     D,$FWRILA
066.064  315 072 066     5198X          CALL    $FWRIB
066.067  321             5199X          POP     D
066.070  311             5200X          RET
                        5201X
066.071  012             5202X $FWRILA  DB      NL
066.072                 5203           XTEXT   FWRIB
```

```
                                5205X **      $FWRIB - WRITE BYTES FROM FILE BUFFER.
                                5206X *
                                5207X *       $FWRIB IS CALLED TO WRITE A NUMBER OF BYTES FROM A FILE BUFFER.
                                5208X *
                                5209X *       ENTRY   (BC) = BYTE COUNT
                                5210X *               (DE) = FWA FOR BYTES
                                5211X *               (HL) = ADDRESS OF FILE BUFFER
                                5212X *       EXIT    TO *FERROR* IF ERROR
                                5213X *               TO CALLER IF OK
                                5214X *                (DE) = ADDRESS OF FIRST UNWRITTEN BYTE
                                5215X *       USES    A;F;B;C;D;E
                                5216X
                                5217X
066.072  315 101 066            5218X $FWRIB  CALL    $FWRIB.
066.075  320                    5219X         RNC                     RETURN IF OK
066.076  303 135 067            5220X         JMP     $FERROR         ERROR
                                5221X
                                5222X
066.101                         5223X $FWRIB. EQU     *
066.101  345                    5224X         PUSH    H
066.102  315 360 066            5225X         CALL    CBT             COPY BUFFER POINTERS TO TEMP CELLS
                                5226X
                                5227X *       COPY DATA FROM USER AREA TO BUFFER
                                5228X
066.105  325                    5229X $WRIB2  PUSH    D               SAVE AREA ADDRESS
066.106  072 123 067            5230X         LDA     T.FLG
066.111  346 004                5231X         ANI     FT.OW           SEE IF OPEN FOR WRITE
066.113  312 247 066            5232X         JZ      $WRIB8          FILE NOT OPEN FOR WRITE
066.116  170                    5233X         MOV     A,B
066.117  261                    5234X         ORA     C
066.120  312 247 066            5235X         JZ      $WRIB8          ALL DONE
                                5236X
                                5237X *       COMPUTE MIN(ROOM IN BUFFER, WRITE COUNT REQUESTED)
                                5238X
066.123  052 126 067            5239X $WRIB3  LHLD    T.PTR
066.126  353                    5240X         XCHG                    (DE) = (FB.PTR) = ADDRESS OF ROOM
066.127  052 132 067            5241X         LHLD    T.LWA           (HL) = LIMIT ADDRESS
066.132  175                    5242X         MOV     A,L
066.133  223                    5243X         SUB     E
066.134  157                    5244X         MOV     L,A
066.135  174                    5245X         MOV     A,H
066.136  232                    5246X         SBB     D
066.137  147                    5247X         MOV     H,A             (HL) = BYTES OF ROOM IN BUFFER
066.140  171                    5248X         MOV     A,C             COMPARE REQUESTED COUNT TO BUFFER ROOM
066.141  225                    5249X         SUB     L
066.142  170                    5250X         MOV     A,B
066.143  234                    5251X         SBB     H
066.144  322 151 066            5252X         JNC     $WRIB4          MORE REQUESTED THEN ROOM
066.147  140                    5253X         MOV     H,B
066.150  151                    5254X         MOV     L,C             USE REQUESTED COUNT
066.151  174                    5255X $WRIB4  MOV     A,H
066.152  265                    5256X         ORA     L
066.153  302 213 066            5257X         JNZ     $WRIB6          SOME ROOM IN BUFFER
                                5258X
                                5259X *       BUFFER IS FULL, EMPTY IT
                                5260X
```

```
066.156  305            5261X          PUSH   B                SAVE COUNT
066.157  052 124 067     5262X          LHLD   T.FWA
066.162  042 126 067     5263X          SHLD   T.PTR            CLEAR REMOVAL POINTER
066.165  353            5264X          XCHG
066.166  052 132 067     5265X          LHLD   T.LWA
066.171  175            5266X          MOV    A,L
066.172  223            5267X          SUB    E
066.173  117            5268X          MOV    C,A
066.174  174            5269X          MOV    A,H
066.175  232            5270X          SBB    D
066.176  107            5271X          MOV    B,A              (BC) = DATA IN BUFFER
066.177  072 122 067     5272X          LDA    T.CHA
066.202  377 005        5273X          DB     SYSCALL,.WRITE   WRITE BUFFER
066.204  301            5274X          POP    B                (BC) = DESIRED COUNT
066.205  322 123 066     5275X          JNC    $WRIB3           GOT THE DATA
                        5276X
                        5277X *        ERROR ON WRITE.
                        5278X
066.210  303 247 066     5279X          JMP    $WRIB8           HAVE ERROR
                        5280X
                        5281X *        GOT THE DATA. MOVE IT FROM BUFFER TO TARGET
                        5282X *
                        5283X *        (BC) = REQUEST COUNT
                        5284X *        (DE) = TO
                        5285X *        (HL) = COUNT
                        5286X *        ((SP)) = FROM
                        5287X
066.213  171            5288X $WRIB6   MOV    A,C
066.214  225            5289X          SUB    L
066.215  117            5290X          MOV    C,A
066.216  170            5291X          MOV    A,B
066.217  234            5292X          SBB    H
066.220  107            5293X          MOV    B,A              REMOVE BYTES ABOUT TO BE MOVED FROM REQUEST COUNT
066.221  305            5294X          PUSH   B
066.222  343            5295X          XTHL                    (HL) = REMAINING REQUEST COUNT
066.223  301            5296X          POP    B                (BC) = COUNT FOR THIS COPY
066.224  343            5297X          XTHL                    (HL) = TARGET ADDR, ((SP)) = REMAINING REQ. COUNT
066.225  176            5298X $WRIB7   MOV    A,M
066.226  022            5299X          STAX   D
066.227  023            5300X          INX    D
066.230  043            5301X          INX    H
066.231  013            5302X          DCX    B
066.232  170            5303X          MOV    A,B
066.233  261            5304X          ORA    C
066.234  302 225 066     5305X          JNZ    $WRIB7           MORE TO GO
066.237  353            5306X          XCHG
066.240  042 126 067     5307X          SHLD   T.PTR            UPDATE POINTER
066.243  301            5308X          POP    B                (BC) = REMAINING COUNT
066.244  303 105 066     5309X          JMP    $WRIB2           SEE IF MORE IN BUFFER
                        5310X
                        5311X *        WRITE COMPLETE.
                        5312X *
                        5313X *        (PSW) = COMPLETION FLAGS
                        5314X
066.247  321            5315X $WRIB8   POP    D                RESTORE TARGET ADDRESS
066.250  341            5316X          POP    H
```

```
066.251  303 006 067  5317X          JMP     CTB             COPY TEMP POINTERS BACK TO BLOCK, EXIT


                      5319X **       $FWBRK  -  BREAKOUTPUT                                  /80.02.GC/
                      5320X *
                      5321X *        $FWBRK empties the specified buffer by filling it with NULLs
                      5322X *        and then writing it.  Note this is used to insure that block
                      5323X *        mode I/O is output if it is not really a serial device (eg.
                      5324X *        writing to AT: from  *EDIT*.
                      5325X *
                      5326X *
                      5327X *        ENTRY:  HL      =  FILE BLOCK POINTER
                      5328X *
                      5329X *        EXIT:   HL      =  FILE BLOCK POINTER
                      5330X *                TO  $FERROR IF ERROR
                      5331X *
                      5332X *        USES:   PSW,BC,DE
                      5333X *
                      5334X
066.254  315 263 066  5335X $FWBRK   CALL    $FWBRK.
066.257  320          5336X           RNC                     NO ERROR
                      5337X
066.260  303 135 067  5338X           JMP     $FERROR
                      5339X
066.263  345          5340X $FWBRK.   PUSH    H
066.264  315 360 066  5341X           CALL    CBT             COPY BUFFER TO TEMPORARY
066.267  315 277 066  5342X           CALL    $FWBRK1
066.272  341          5343X           POP     H
066.273  315 006 067  5344X           CALL    CTB             COPY TEMPORARY TO BUFFER
066.276  311          5345X           RET
                      5346X
066.277  052 132 067  5347X $FWBRK1  LHLD    T.LWA
066.302  353          5348X           XCHG                    DE = BUFFER LWA
066.303  052 126 067  5349X           LHLD    T.PTR           HL = BUFFER PTR
066.306  173          5350X           MOV     A,E
066.307  225          5351X           SUB     L
066.310  117          5352X           MOV     C,A
066.311  172          5353X           MOV     A,D
066.312  234          5354X           SBB     H
066.313  107          5355X           MOV     B,A             BC = DE - HL
066.314  261          5356X           ORA     C
066.315  310          5357X           RZ                      THE BUFFER IS ALREADY FLUSHED
                      5358X
                      5359X *        FILL THE BUFFER WITH NULLS
                      5360X
066.316  170          5361X FWBRK2   MOV     A,B
066.317  261          5362X           ORA     C
066.320  312 332 066  5363X           JZ      FWBRK3          NO MORE LEFT TO FILL
                      5364X
066.323  066 000      5365X           MVI     M,0
066.325  043          5366X           INX     H
066.326  013          5367X           DCX     B
066.327  303 316 066  5368X           JMP     FWBRK2
                      5369X
```

```
066.332  052 124 067  5370X FWBRK3  LHLD   T.FWA
066.335  042 126 067  5371X         SHLD   T.PTR
066.340  353          5372X         XCHG                DE = BUFFER FWA
066.341  052 132 067  5373X         LHLD   T.LWA        HL = BUFFER LWA
066.344  175          5374X         MOV    A,L
066.345  223          5375X         SUB    E
066.346  117          5376X         MOV    C,A
066.347  174          5377X         MOV    A,H
066.350  232          5378X         SBB    D
066.351  107          5379X         MOV    B,A          BC = HL - DE  ( BC = COUNT )
066.352  072 122 067  5380X         LDA    T.CHA
066.355  377 005      5381X         DB     SYSCALL,.WRITE
066.357  311          5382X         RET
066.360               5383          XTEXT  FUTIL



                      5385X **      $FUTIL - UTILITY ROUTINES FOR FILE BLOCK ROUTINES.
                      5386X
                      5387X **      CBT - COPY BLOCK POINTERS TO TEMP CELLS.
                      5388X *
                      5389X *       ENTRY  (HL) = FILE BLOK FWA
                      5390X *       EXIT   NONE
                      5391X *       USES   A,F,H,L
                      5392X
066.360  325          5393X CBT     PUSH   D
066.361  305          5394X         PUSH   B            SAVE REGISTERS
000.000               5395X         ERRNZ  TLEN-10      ASSUME 10 BYTES TO MOVE
066.362  021 122 067  5396X         LXI    D,T.CHA      (DE) = TARGET FOR MOVE
066.365  006 005      5397X         MVI    B,10/2
066.367  176          5398X CBT1    MOV    A,M          COPY FILE BUFFER INTO WORK AREA
066.370  022          5399X         STAX   D
066.371  043          5400X         INX    H
066.372  023          5401X         INX    D
066.373  176          5402X         MOV    A,M
066.374  022          5403X         STAX   D
066.375  043          5404X         INX    H
066.376  023          5405X         INX    D
066.377  005          5406X         DCR    B
067.000  302 367 066  5407X         JNZ    CBT1         MORE TO GO
067.003  301          5408X         POP    B
067.004  321          5409X         POP    D            (DE) = DATA TARGET ADDRESS
067.005  311          5410X         RET
                      5411X
                      5412X
                      5413X **      CTB - COPY TEMP CELLS BACK TO FILE BLOCK.
                      5414X *
                      5415X *       ENTRY  (HL) = FILE BLOCK ADDRESS
                      5416X *       EXIT   NONE
                      5417X *       USES   NONE
                      5418X
067.006  365          5419X CTB     PUSH   PSW
067.007  325          5420X         PUSH   D
067.010  305          5421X         PUSH   B
067.011  345          5422X         PUSH   H            SAVE REGISTERS
```

```
067.012  006 004      5423X          MVI     B,8/2
067.014  021 122 067  5424X          LXI     D,T.CHA
067.017  032          5425X CTB1     LDAX    D
067.020  167          5426X          MOV     M,A
067.021  023          5427X          INX     D
067.022  043          5428X          INX     H
067.023  032          5429X          LDAX    D
067.024  167          5430X          MOV     M,A
067.025  023          5431X          INX     D
067.026  043          5432X          INX     H
067.027  005          5433X          DCR     B
067.030  302 017 067  5434X          JNZ     CTB1            RESTORE FILE BUFFER VALUES
067.033  341          5435X          POP     H
067.034  301          5436X          POP     B
067.035  321          5437X          POP     D
067.036  361          5438X          POP     PSW
067.037  311          5439X          RET


                      5441X **       $FFB - FILE FILE BUFFER.
                      5442X *
                      5443X *        $FFB FILLS THE FILE BUFFER BY READING FROM THE FILE.
                      5444X *
                      5445X *        ENTRY   NONE
                      5446X *        EXIT    'C' SET IF READ INCOMPLETE
                      5447X *                 (A) = ERROR CODE
                      5448X *                 'C' CLEAR IF READ COMPLETEE
                      5449X *                 DATA IN BUFFER
                      5450X *        USES    A,F,D,E,H,L
                      5451X
                      5452X
067.040  072 134 067  5453X $FFB     LDA     EOFFLG
067.043  037          5454X          RAR
067.044  330          5455X          RC                      EOF
                      5456X
                      5457X *        CAN READ MORE. DO SO
                      5458X
067.045  305          5459X          PUSH    B               SAVE COUNT
067.046  052 124 067  5460X          LHLD    T.FWA
067.051  042 126 067  5461X          SHLD    T.PTR           CLEAR REMOVAL POINTER
067.054  353          5462X          XCHG
067.055  052 132 067  5463X          LHLD    T.LWA
067.060  042 130 067  5464X          SHLD    T.LIM           SET DATA LIMIT
067.063  175          5465X          MOV     A,L
067.064  223          5466X          SUB     E
067.065  117          5467X          MOV     C,A
067.066  174          5468X          MOV     A,H
067.067  232          5469X          SBB     D
067.070  107          5470X          MOV     B,A             (BC) = ROOM IN BUFFER
067.071  072 122 067  5471X          LDA     T.CHA
067.074  377 004      5472X          DB      SYSCALL+.READ   READ BUFFER
067.076  120          5473X          MOV     D,B             (D) = SECTORS UNREAD
067.077  301          5474X          POP     B               (BC) = DESIRED COUNT
067.100  320          5475X          RNC                     GOT THE DATA
```

```
                        5476X
                        5477X *       ERROR ON READ, SEE IF EOF
                        5478X
067.101  027           5479X         RAL
067.102  062 134 067   5480X         STA     EOFFLG          SET EOF, WE HOPE
067.105  376 003       5481X         CPI     EC.EOF*2+1
067.107  037           5482X         RAR
067.110  300           5483X         RNE                     IS NOT EOF, RETURN NOW!
067.111  072 131 067   5484X         LDA     T.LIM+1
067.114  222           5485X         SUB     D
067.115  062 131 067   5486X         STA     T.LIM+1         SET AMOUNT OF DATA WE DID GET
067.120  247           5487X         ANA     A
067.121  311           5488X         RET                     EXIT WITH DATA
                        5489X
                        5490X
                        5491X **      TEMP CELLS TO HOLD FILE BLOCK POINTERS DURING I/O
                        5492X
000.000                 5493X         ERRNZ   FB.CHA
067.122  000           5494X T.CHA    DB      0               CHANNEL NUMBER
000.000                 5495X         ERRNZ   *-T.CHA-FB.FLG
067.123  000           5496X T.FLG    DB      0               FLAG BYTE
000.000                 5497X         ERRNZ   *-T.CHA-FB.FWA
067.124  000 000       5498X T.FWA    DW      0
000.000                 5499X         ERRNZ   *-T.CHA-FB.PTR
067.126  000 000       5500X T.PTR    DW      0
000.000                 5501X         ERRNZ   *-T.CHA-FB.LIM
067.130  000 000       5502X T.LIM    DW      0
000.000                 5503X         ERRNZ   *-T.CHA-FB.LWA
067.132  000 000       5504X T.LWA    DW      0
000.012                 5505X TLEN     EQU     *-T.CHA         LENGTH OF TEMP CELLS
                        5506X
067.134  000           5507X EOFFLG   DB      0
067.135                 5508          XTEXT   FERROR


                        5510X **      $FERROR - PROCESS FILE ERRORS.
                        5511X *
                        5512X *       $FERROR IS CALLED TO COMPLAIN ABOUT AN ERROR ENCOUNTERED
                        5513X *       WHEN PROCESSING FILES.
                        5514X *
                        5515X *       ENTRY   (A) = ERROR CODE
                        5516X *               (HL) = ADDRESS OF FILE NAME - FB.NAM
                        5517X *       EXIT    TO RESTART
                        5518X *       USES    ALL
                        5519X
                        5520X
067.135  365           5521X $FERROR PUSH    PSW             SAVE CODE
067.136  315 136 031   5522X         CALL    $TYPTX
067.141  012 007 105   5523X         DB      NL,BELL,'ERROR ON FILE',' '+200Q
067.161  021 012 000   5524X         LXI     D,FB.NAM
067.164  031           5525X         DAD     D
                        5526X
                        5527X *       PRINT FILE NAME
                        5528X
```

```
067.165  176          5529X $FERR1  MOV     A,M
067.166  043          5530X         INX     H               ADVANCE MESSAGE
067.167  247          5531X         ANA     A
067.170  312 201 067  5532X         JZ      $FERR2
067.173  315 161 064  5533X         CALL    $WCHAR
067.176  303 165 067  5534X         JMP     $FERR1
                      5535X
                      5536X *       TYPE ERROR MESSAGE
                      5537X
067.201  315 136 031  5538X $FERR2  CALL    $TYPTX
067.204  040 055 240  5539X         DB      ' -',' '+200Q
067.207  046 012       5540X         MVI     H,NL
067.211  361          5541X         POP     PSW             (A) = CODE
067.212  377 057       5542X         DB      SYSCALL,.ERROR
067.214  303 244 043  5543X         JMP     RESTART         EXIT
```

```
                                5546  **      OPCTAB - OPCODE TABLE.
                                5547  *
                                5548  *       OPCTAB CONTAINS AN ENTRY FOR EACH OPCODE.
                                5549  *       THE TABLE IS SEARCHED SERIALLY, SO THE MOST HEAVILY
                                5550  *       USED OPCODES SHOULD BE PLACED TOWARDS THE FRONT.
                                5551  *
                                5552  *       THE TABLE FORMAT IS:
                                5553  *
                                5554  *       DB      'OPCOD'         CHARACTERS 1 - N-1
                                5555  *       DB      'E'+80H         LAST CHARACTER HAS HIGH BIT SET
                                5556  *       DB      F               =1 IF TO ASSEMBLE REGARDLESS OF *IF*S
                                5557  *       DB       F              =1 IF NOT TO AUTOMATICALLY DEFINE LABEL
                                5558  *       DB        IIIIII        = OPCODE TYPE INDEX
                                5559  *       DB      CODE            OPCODE, IF MACHINE OP
                                5560  *                                IF PSEUDO OP, =0 IF IN GROUP 1
                                5561
000.200                         5562  OF.CE   EQU     200Q            CONDITIONAL ASSEMBLY EXCEPTION
000.100                         5563  OF.LD   EQU     100Q            DEFER LABEL DEFINITION
                                5564
                                5565
067.217                         5566  OPCTAB  EQU     *
067.217  101 103 311            5567          DB      'AC','I'+80H,1,316Q
067.224  101 104 303            5568          DB      'AD','C'+80H,3,210Q
067.231  101 104 304            5569          DB      'AD','D'+80H,3,200Q
067.236  101 104 311            5570          DB      'AD','I'+80H,1,306Q
067.243  101 116 301            5571          DB      'AN','A'+80H,3,240Q
067.250  101 116 311            5572          DB      'AN','I'+80H,1,346Q
067.255  103 101 114            5573          DB      'CAL','L'+80H,2,315Q
067.263  103 303 002            5574          DB      'C','C'+80H,2,334Q
067.267  103 305 002            5575          DB      'C','E'+80H,2,314Q                              /WCZ062680/
067.273  103 315 002            5576          DB      'C','M'+80H,2,374Q
067.277  103 115 301            5577          DB      'CM','A'+80H,0,057Q
067.304  103 115 303            5578          DB      'CM','C'+80H,0,077Q
067.311  103 115 320            5579          DB      'CM','P'+80H,3,270Q
067.316  103 116 303            5580          DB      'CN','C'+80H,2,324Q
067.323  103 116 305            5581          DB      'CN','E'+80H,2,304Q                             /WCZ062680/
067.330  103 116 332            5582          DB      'CN','Z'+80H,2,304Q
067.335  103 320 002            5583          DB      'C','P'+80H,2,364Q
067.341  103 120 305            5584          DB      'CP','E'+80H,2,354Q
067.346  103 120 311            5585          DB      'CP','I'+80H,1,376Q
067.353  103 120 317            5586          DB      'CP','O'+80H,2,344Q
067.360  103 332 002            5587          DB      'C','Z'+80H,2,314Q
067.364  104 101 301            5588          DB      'DA','A'+80H,0,047Q
067.371  104 101 304            5589          DB      'DA','D'+80H,7,011Q
067.376  104 302 015            5590          DB      'D','B'+80H,13,1
070.002  104 103 322            5591          DB      'DC','R'+80H,4,005Q
070.007  104 103 330            5592          DB      'DC','X'+80H,7,013Q
070.014  104 311 000            5593          DB      'D','I'+80H,0,363Q
070.020  104 323 016            5594          DB      'D','S'+80H,14,0
070.024  104 327 017            5595          DB      'D','W'+80H,15,1
070.030  105 311 000            5596          DB      'E','I'+80H,0,373Q
070.034  105 112 105            5597          DB      'EJEC','T'+80H,OF.LD+16,0
070.043  105 114 123            5598          DB      'ELS','E'+80H,OF.LD+17+OF.CE,0
070.051  105 116 304            5599          DB      'EN','D'+80H,OF.LD+18+OF.CE,1
070.056  105 116 104            5600          DB      'ENDI','F'+80H,OF.LD+19+OF.CE,0
070.065  105 121 325            5601          DB      'EQ','U'+80H,OF.LD+20,0
```

```
070.072  105 122 122  5602          DB      'ERRM','I'+80H,OF,LD+36,0                /80.09.BB/
070.101  105 122 122  5603          DB      'ERRN','Z'+80H,OF,LD+34,0                /80.09.BB/
070.110  105 122 122  5604          DB      'ERRP','L'+80H,OF,LD+35,0                /80.09.BB/
070.117  105 122 122  5605          DB      'ERRZ','R'+80H,OF,LD+33,0                /80.09.BB/
070.126  110 114 324  5606          DB      'HL','T'+80H,0,166Q
070.133  111 306 126  5607          DB      'I','F'+80H,OF,LD+22,0
070.137  111 316 001  5608          DB      'I','N'+80H,1,333Q
070.143  111 116 322  5609          DB      'IN','R'+80H,4,004Q
070.150  111 116 330  5610          DB      'IN','X'+80H,7,003Q
070.155  112 303 002  5611          DB      'J','C'+80H,2,332Q
070.161  112 305 002  5612          DB      'J','E'+80H,2,312Q
070.165  112 315 002  5613          DB      'J','M'+80H,2,372Q
070.171  112 115 320  5614          DB      'JM','P'+80H,2,303Q
070.176  112 116 303  5615          DB      'JN','C'+80H,2,322Q
070.203  112 116 305  5616          DB      'JN','E'+80H,2,302Q
070.210  112 116 332  5617          DB      'JN','Z'+80H,2,302Q
070.215  112 320 002  5618          DB      'J','P'+80H,2,362Q
070.221  112 120 305  5619          DB      'JP','E'+80H,2,352Q
070.226  112 120 317  5620          DB      'JP','O'+80H,2,342Q
070.233  112 332 002  5621          DB      'J','Z'+80H,2,312Q
070.237  114 104 301  5622          DB      'LD','A'+80H,2,072Q
070.244  114 104 101  5623          DB      'LDA','X'+80H,9,012Q
070.252  114 110 114  5624          DB      'LHL','D'+80H,2,052Q
070.260  114 117 306  5625          DB      'LO','F'+80H,OF,LD+23,0
070.265  114 117 316  5626          DB      'LO','N'+80H,OF,LD+24,0
070.272  114 130 311  5627          DB      'LX','I'+80H,11,001Q
070.277  115 117 326  5628          DB      'MO','V'+80H,12,100Q
070.304  115 126 311  5629          DB      'MV','I'+80H,8,006Q
070.311  116 117 320  5630          DB      'NO','P'+80H,0,000Q
070.316  117 122 301  5631          DB      'OR','A'+80H,3,260Q
070.323  117 122 307  5632          DB      'OR','G'+80H,OF,LD+25,0
070.330  117 122 311  5633          DB      'OR','I'+80H,1,366Q
070.335  117 125 324  5634          DB      'OU','T'+80H,1,323Q
070.342  120 103 110  5635          DB      'PCH','L'+80H,0,351Q
070.350  120 117 320  5636          DB      'PO','P'+80H,6,301Q
070.355  120 125 123  5637          DB      'PUS','H'+80H,6,305Q
070.363  122 101 314  5638          DB      'RA','L'+80H,0,027Q
070.370  122 101 322  5639          DB      'RA','R'+80H,0,037Q
070.375  122 303 000  5640          DB      'R','C'+80H,0,330Q
071.001  122 305 000  5641          DB      'R','E'+80H,0,310Q
071.005  122 105 324  5642          DB      'RE','T'+80H,0,311Q
071.012  122 114 303  5643          DB      'RL','C'+80H,0,007Q
071.017  122 315 000  5644          DB      'R','M'+80H,0,370Q
071.023  122 116 303  5645          DB      'RN','C'+80H,0,320Q
071.030  122 116 305  5646          DB      'RN','E'+80H,0,300Q
071.035  122 116 332  5647          DB      'RN','Z'+80H,0,300Q
071.042  122 320 000  5648          DB      'R','P'+80H,0,360Q
071.046  122 120 305  5649          DB      'RP','E'+80H,0,350Q
071.053  122 120 317  5650          DB      'RP','O'+80H,0,340Q
071.060  122 122 303  5651          DB      'RR','C'+80H,0,017Q
071.065  122 123 324  5652          DB      'RS','T'+80H,10,307Q
071.072  122 332 000  5653          DB      'R','Z'+80H,0,310Q
071.076  123 102 302  5654          DB      'SB','B'+80H,3,230Q
071.103  123 102 311  5655          DB      'SB','I'+80H,1,336Q
071.110  123 103 101  5656          DB      'SCAL','L'+80H,1,377Q
071.117  123 105 324  5657          DB      'SE','T'+80H,OF,LD+26,0
```

```
071.124  123 110 114  5658          DB      'SHL','D'+80H,2,042Q
071.132  123 120 101  5659          DB      'SPAC','E'+80H,0F.LD+27,0
071.141  123 120 110  5660          DB      'SPH','L'+80H,0,371Q
071.147  123 124 301  5661          DB      'ST','A'+80H,2,062Q
071.154  123 124 101  5662          DB      'STA','X'+80H,9,002Q
071.162  123 124 303  5663          DB      'ST','C'+80H,0,067Q
071.167  123 124 314  5664          DB      'ST','L'+80H,0F.LD+28,0
071.174  123 125 302  5665          DB      'SU','B'+80H,3,220Q
071.201  123 125 311  5666          DB      'SU','I'+80H,1,326Q
071.206  124 111 124  5667          DB      'TITL','E'+80H,0F.LD+29,0
071.215  130 103 110  5668          DB      'XCH','G'+80H,0,353Q
071.223  130 122 301  5669          DB      'XR','A'+80H,3,250Q
071.230  130 122 311  5670          DB      'XR','I'+80H,1,356Q
071.235  130 124 110  5671          DB      'XTH','L'+80H,0,343Q
071.243  103 117 104  5672          DB      'COD','E'+80H,0F.LD+30,0
071.251  111 116 103  5673          DB      'INCL','U'+80H,0F.LD+31,0
071.260  130 124 105  5674          DB      'XTEX','T'+80H,0F.LD+31,0
071.267  116 117 122  5675          DB      'NORE','F'+80H,0F.LD+32,0                    /WCZ062680/
                      5676
071.276  000         5677          DB      0       END OF TABLE
```

```
                                5680  **      THE FOLLOWING AREAS ARE ASSEMBLY AREAS FOR LISTING LINES.


                                5682  **      HEADING LINE.
                                5683  *
                                5684
                                5685
071.277                         5686  HEADING EQU     *               START OF PAGE HEADING
071.277  040 040 040           5687  TTLTXT  DB      '                                                       '
000.062                         5688  TTXTL   EQU     *-TTLTXT        LENGTH
071.361  011                   5689          DB      TAB                                             /80.09.BB/
071.362  110 105 101            5690          DB      'HEATH ASM #104.06.00'                          /80.09.BB/
072.006  012                   5691          DB      NL              NEW LINE
                                5692
072.007  040 040 040           5693  STLTXT  DB      '                                                       '
000.062                         5694  STXTL   EQU     *-STLTXT        LENGTH
072.071  011                   5695          DB      TAB                                             /80.09.BB/
072.072  060 066 055            5696  HEADC   DB      '06-DEC-79'
000.011                         5697  HEADCL  EQU     *-HEADC
072.103  040 040 120           5698          DB      '  Page '
072.113                         5699  HEADA   DS      3                                               /WCZ062680/
000.003                         5700  HEADAL  EQU     *-HEADA                                         /WCZ062680/
072.116  012 012                5701          DB      NL,NL           2 BLANK LINES
000.221                         5702  HEADLEN EQU     *-HEADING       HEADER LENGTH


                                5704  **      LISTING LINE WORK AREA
                                5705
                                5706
072.120                         5707  DSPLIN  DS      0
072.120  040 040 040           5708          DB      '   '           ERROR FLAGS
072.123  040 040 040            5709  DSPLNA  DB      '   '           BANK NUMBER
072.126  056                    5710          DB      '.'
072.127  040 040 040            5711          DB      '   '           BANK ADDRESS
072.132  040 040                5712          DB      '  '
072.134  040 040 040            5713  DSPLNB  DB      '   '           BYTE 1
072.137  040                    5714          DB      ' '
072.140  040 040 040            5715          DB      '   '           BYTE 2
072.143  040                    5716          DB      ' '
072.144  040 040 040            5717  DSPLNC  DB      '   '           BYTE 3
072.147  040                    5718          DB      ' '
072.150                         5719  DSPLIM  EQU     *               LIMIT FOR OCTAL BYTES
072.150  060 060 060            5720  DSPLND  DB      '00000'         LINE NUMBER                     /80.02.GC/
072.155  130                    5721  DSPLNE  DB      'X'             XTEXT FLAG                      /80.02.GC/
072.156  040 040                5722          DB      '  '                                            /80.02.GC/
000.040                         5723  DSPLEN  EQU     *-DSPLIN        LENGTH OF HEADER
072.160  000                    5724          DB      0               TERMINATES DSPLIN FOR $DTB
072.161  200                    5725          DB      200Q            TERMINATES DSPLIN FOR .PRINT
```

```
072.162  000 000      5728  OBBPTR  DW      0               BYTE DECODE POINTER
                       5729
072.164  000          5730  PASS    DB      0               PASS NUMBER
072.165  000 000      5731  ERRCNT  DW      0               NUMBER OF ERRORS IN PASS
072.167  000          5732  ERRSHO  DB      0               <>0 IF TO TYPE ERRORS ON CONSOLE
072.170  000 000      5733  STATNO  DW      0               STATEMENT NUMBER
072.172  000          5734  LSTCTL  DB      0               LISTING CONTROL OPTIONS
072.173  000          5735  LSTCTLS DB      0               FORCED SET LISTING CONTROL
072.174  000          5736  LSTCTLC DB      0               FORCED CLEAR LISTING CONTROL (INVERTED MASK)
072.175  000          5737  ENDFLG  DB      0               NON-ZERO IF END STATEMENT READ
072.176  000 000      5738  ORG     DW      0               ORIGIN POINTER
072.200  000 000      5739  SORG    DW      0               VALUE OF *ORG* AT BEGINNING OF STATEMENT
072.202  000          5740  ERRFLG  DB      0               ERROR FLAGS FOR THIS STATEMENT
072.203  000          5741  GRPFLG  DB      0               <>0 IF HAVE ASSEMBLED 2ND GROUP INSTRUCTIONS
072.204  000 000      5742  XREFCNT DW      0               Cross Reference Count          /80.03.sc/
072.206  000          5743  XTXFLG  DB      0               <>0 IF READING FROM XTEXT
072.207  000          5744  XTXLINE DB      0               <>0 IF CURRENT LINE FROM XTEXT
                       5745
                       5746  *       CODE GENERATION FLAG
                       5747
072.210  000          5748  FTFLAG  DB      0               FILE TYPE (FT.PIC, FT.ABS)
072.211  000          5749  RELFLG  DB      0               ST.REL IF RELOCATABLE ASSEMBLY
072.212  000          5750  CODEFLG DB      0               <>0 IF 'CODE' PSEUDO ENCOUNTERED THIS PASS
                       5751
072.213  000          5752  LARGE   DB      0               <>0 IF TO SWAP OVERLAY
                       5753
                       5754
                       5755  *       BINARY OUTPUT MANAGEMENT
                       5756
072.214                5757  BINFNAM DS      FB.NAML         BINARY FILE NAME (=0 IF NONE)
072.235  000           5758  BINCSN  DB      0               CURRENT SECTOR NUMBER IN BINBUF
072.236  000           5759  BINSKW  DB      0               BYTES OF HEADER ON FRONT OF BINARY FILE
                       5760
072.237                5761  ABSHDR  DS      0               HEADER FOR ABS BINARY FILE
072.237  377 000       5762          DB      377Q,FT.ABS
072.241  377 377       5763  ABSFWA  DW      -1              LOWEST ADDRESS GENERATED (=0 IF PIC)
072.243  000 000       5764  ABSLEN  DW      0               LENGTH
072.245  200 042       5765  ABSENT  DW      USERFWA         ENTRY POINT
                       5766
072.247  000 000       5767  ABSLWA  DW      0               MAX ADDRESS GENERATED
                       5768
072.251                5769  PICHDR  DS      0               HEADER FOR PIC BINARY FILE
072.251  377 001       5770          DB      377Q,FT.PIC
072.253  000 000       5771  PICLEN  DW      0               LENGTH OF ENTIRE THING
072.255  000 000       5772  PICPTR  DW      0               POINTER TO REL TABLE


                       5774  **      LISTING FORMAT AND CONTROL FLAGS
                       5775
072.257  000           5776  WIDE    DB      0               <>0 IF WIDE SWITCH
072.260  074           5777  PAGEDP  DB      60              DEPTH OF PAGE
072.261  000           5778  FORMDP  DB      0               FORM DEPTH (ONLY IF PRINTER WONT TAKE FORMFEED)
                       5779
                       5780
                       5781  **      DYNAMIC TABLE ALLOCATION
```

```
                                5782
072.262  154 102                5783  SYMFWA  DW      SYMTAB          FWA
072.264  154 102                5784  SYMPTR  DW      SYMTAB          LWA+ (SMALL SLOP FACTOR)
072.266  000 000                5785  RELLWA  DW      0               REL TABLE END
072.270  000 000                5786  RELPTR  DW      0               REL TAB ACTIVE POINTER (IT GROWS DOWN)


                                5788  **      FILE BUFFERS
                                5789
072.272                         5790  LISTFB  DS      0               LISTING FILE BLOCK
072.272  001                    5791          DB      CN.LST          LISTING CHANNEL
072.273  000                    5792          DB      0               FLAG
072.274  336 074                5793          DW      LISTBUF
072.276  336 074                5794          DW      LISTBUF
072.300  336 074                5795          DW      LISTBUF
072.302  336 076                5796          DW      LISTBUF+LISTBFL
072.304                         5797          DS      FB.NAML         LISTING FILE NAME
                                5798
072.325                         5799  SORCFB  DS      0               SOURCE FILE BLOCK
072.325  002                    5800          DB      CN.SOU          SOURCE CHANNEL
072.326  000                    5801          DB      0               FLAG
072.327  336 076                5802          DW      SORCBUF
072.331  336 076                5803          DW      SORCBUF
072.333  336 076                5804          DW      SORCBUF
072.335  336 077                5805          DW      SORCBUF+SORCBFL
072.337                         5806          DS      FB.NAML         LISTING FILE NAME
                                5807
072.360                         5808  XTXFB   DS      0               XTEXT FILE BLOCK
072.360  003                    5809          DB      CN.XTX          XTEXT CHANNEL
072.361  000                    5810          DB      0               FLAG
072.362  336 077                5811          DW      XTXBUF
072.364  336 077                5812          DW      XTXBUF
072.366  336 077                5813          DW      XTXBUF
072.370  336 100                5814          DW      XTXBUF+XTXBFL
072.372                         5815          DS      FB.NAML         XTEXT FILE NAME
                                5816
073.013                         5817  TEMPFB  DS      0               TEMP FILE BLOCK         /80.03.GC/
073.013  004                    5818          DB      CN.TMP                                  /80.03.GC/
073.014  000                    5819          DB      0                                       /80.03.GC/
073.015  336 100                5820          DW      TMPBUF                                  /80.03.GC/
073.017  336 100                5821          DW      TMPBUF                                  /80.03.GC/
073.021  336 100                5822          DW      TMPBUF                                  /80.03.GC/
073.023  336 101                5823          DW      TMPBUF+TMPBFL                           /80.03.GC/
073.025                         5824          DS      FB.NAML         TEMP FILE NAME          /80.03.GC/
                                5825
```

start of buffer (next pointer)

```
                                5827  **      CNDFLG - CONDITIONAL ASSEMBLY FLAG.
                                5828  *
                                5829  *       =000 - NO CONDITIONS
                                5830  *       =200 - AM ASSEMBLING
                                5831  *       =201 - AM SKIPPING
                                5832
073.046  000                   5833  CNDFLG  DB      0               CONDITIONAL ASSEMBLY FLAG
                                5834
073.047  001                   5835  EJEFLG  DB      1               NON-ZERO IF TO EJECT
073.050  000                   5836  LINCNT  DB      0               LINES PER LAGE
073.051  000                   5837  PAGNUM  DB      0               PAGE NUMBER
                                5838
                                5839  *       RELOCATION FLAGS
                                5840
073.052  000                   5841  EXPREL  DB      0               =ST.REL IF EXPRESION IS RELOCATABLE
073.053  000                   5842  TOKREL  DB      0               =ST.REL IF TOKEN IS RELOCATABLE
                                5843
                                5844
073.054                         5845  PATCH   DS      64              PATCH AREA
```

```
                                5848  **      PRS - PRESET ASSEMBLER.
                                5849  *
                                5850  *       PRS IS THE INITIAL ENTRY POINT FOR THIS PROGRAM.
                                5851  *
                                5852  *       IT GETS THE COMMAND LINE (IF IT WASENT PASSED ON THE STACK)
                                5853  *       CRACKS THE FILE NAMES AND SWITCHES, AND SETS UP THE ASSEMBLER.
                                5854  *
                                5855  *       ***********
                                5856  *       * N O T E *     THIS CODE IS OVERLAID DURING ASSEMBLY BY BUFFERS AND
                                5857  *       ***********     WORKAREAS. IT MAY NOT BE RE-ENTERED AFTER THE INITIAL TIME.
                                5858  *
                                5859  *
                                5860  *       PRS PERFORMS 2 TASKS:
                                5861  *
                                5862  *       1) GET COMMAND LINE, CRACK SWITCHES, AND OPEN FILES:
                                5863  *               BINARY FILE
                                5864  *               LISTING FILE
                                5865  *               SOURCE FILE
                                5866  *       2) SETUP DYNAMIC TABLES
                                5867  *               SYMBOL TABLE
                                5868  *               RELOCATION TABLE
                                5869  *
                                5870  *       PRS IS THE ENTRY POINT FOR THIS ASSEMBLER. IF THE STACK IS NON-EMPTY
                                5871  *       IT IS ASSUMED TO CONTAIN THE COMMAND LINE. (1ST CHARACTER PUSHED
                                5872  *       ON LAST)
                                5873  *
                                5874  *       FROM THEN ON, STACK DISCIPLINE IS * * NOT MAINTAINED * *
                                5875  *       FOR THIS ROUTINE. ITS SUBROUTINES MAY VECTOR BACK TO IT FOR EXCEPTIONAL
                                5876  *       CASES, WITH THE STACK UNCLEAR. THE STACK IS KEPT 'EMPTY' ((SP) = STACK)
                                5877  *       WHILE IN PRS, PRS EXIT TO 'ASM' VIA A JUMP.
                                5878  *       ENTRY   FROM SYSTEM
                                5879  *       EXIT    TO H8ASM
                                5880  *       USES    ALL
                                5881
                                5882
073.154                         5883  PRS     EQU     *
                                5884
                                5885  *       CHECK THE HDOS VERSION
                                5886
073.154  377 011               5887          DB      SYSCALL,.VERS                                   /79.12.GC/
073.156  332 166 075           5888          JC      PRSERR1         PROBABLY NO VERSION SYSTEM CALL /79.12.GC/
073.161  376 040               5889          CPI     VERS                                            /79.12.GC/
073.163  302 166 075           5890          JNZ     PRSERR1         NOT THE CORRECT VERSION OF HDOS /79.12.GC/
                                5891
                                5892  *       Initialize XREF link and temp default device and type   /80.06.sc/
                                5893
073.166  021 010 102           5894          LXI     D,LINE          Address for device              /80.06.sc/
073.171  041 016 102           5895          LXI     H,LINE+6        anywhere for name               /80.06.sc/
073.174  076 377               5896          MVI     A,-1            link channel                    /80.06.sc/
073.176  377 054               5897          SCALL   .NAME           decode entry name               /80.06.sc/
                                5898
073.200  052 010 102           5899          LHLD    LINE                                            /80.06.sc/
073.203  042 214 043           5900          SHLD    XREF            stuff device                    /80.06.sc/
073.206  042 033 076           5901          SHLD    SDVA3                                           /WCZ062580/
073.211  072 012 102           5902          LDA     LINE+2                                          /80.06.sc/
073.214  062 216 043           5903          STA     XREF+2          stuff unit                      /80.06.sc/
```

```
073.217  062 035 076  5904          STA     SDVA3+2                                 /WCZ062580/
                      5905
                      5906  *       SEE IF A COMMAND IS ON THE STACK
                      5907
073.222  041 154 102  5908          LXI     H,RMEML
073.225  377 052      5909          DB      SYSCALL,.SETTP  SET LIMIT (TEMPORARILY, UNTIL *BOOT*)
073.227  332 170 075  5910          JC      PRSERR          NOT ENOUGH MEMORY
073.232  041 232 043  5911          LXI     H,CCHIT
073.235  076 003      5912          MVI     A,CTLC
073.237  377 041      5913          DB      SYSCALL,.CTLC   SETUP CTL-C PROCESSING
073.241  041 000 000  5914          LXI     H,0
073.244  071          5915          DAD     SP              (HL) = STACK
073.245  353          5916          XCHG                    (DE) = STACK VALUE
073.246  076 200      5917          MVI     A,#STACK
073.250  223          5918          SUB     E
073.251  117          5919          MOV     C,A
073.252  076 042      5920          MVI     A,STACK/256
073.254  232          5921          SBB     D
073.255  107          5922          MOV     B,A             (BC) = BYTES ON STACK
073.256  261          5923          ORA     C
073.257  312 275 073  5924          JZ      PRS1            READ COMMAND LINE
073.262  041 010 102  5925          LXI     H,LINE
073.265  315 252 030  5926          CALL    $MOVE           MOVE IN LINE
073.270  046 000      5927          MVI     H,0             GUARANTEE TERMINATOR
073.272  303 002 074  5928          JMP     PRS3            CRACK LINE
                      5929
                      5930  *       ANNOUNCE PRODUCT
                      5931
073.275  315 136 031  5932  PRS1    CALL    $TYPTX
073.300  012 110 104  5933          DB      NL,'HDOS ASsembler Issue #104.06.00 /',ENL      /78.12.GC/
                      5934
                      5935  *       READ COMMAND LINE
                      5936
073.342  377 056      5937  PRS2    DB      SYSCALL,.CLEARA CLEAR ALL CHANNELS
073.344  257          5938          XRA     A
073.345  062 023 051  5939          STA     XIEXIE          SAY NO XIEXT DEFAULT DEVICES    /WCZ062780/
073.350  062 273 072  5940          STA     LISTFB+FB.FLG
073.353  062 326 072  5941          STA     SORCFB+FB.FLG   CLEAR FILE BUFFERS
073.356  062 014 073  5942          STA     TEMPFB+FB.FLG                           /80.06.GC/
073.361  061 200 042  5943          LXI     SP,STACK        CLEAN STACK
073.364  315 136 031  5944          CALL    $TYPTX
073.367  012 252      5945          DB      NL,'*'+200Q
073.371  041 010 102  5946          LXI     H,LINE
073.374  315 106 064  5947          CALL    $RTL            READ LINE               /78.10.GC/
073.377  332 211 043  5948          JC      EXIT            CTL-D STRUCK
                      5949
                      5950  *       HAVE COMMAND LINE, DECODE SWITCHES
                      5951
074.002  315 354 075  5952  PRS3    CALL    SDV             SET DEFAULT VALUES
074.005  021 057 077  5953          LXI     D,SWITAB
074.010  041 010 102  5954          LXI     H,LINE
074.013  315 160 101  5955          CALL    $DRS            DECODE AND REMOVE SWITCHES
074.016  332 140 075  5956          JC      SW.ERR          SWITCH ERROR
                      5957
                      5958  *       HAVE CRACKED SWITCHES FROM COMMAND LINE, NOW DECODE FILE NAMES
                      5959
```

```
074.021  257            5960          XRA     A
074.022  062 214 072     5961          STA     BINFNAM         CLEAR BINARY FILE NAME
074.025  062 304 072     5962          STA     LISTFB+FB.NAM   CLEAR LISTING FILE NAME
074.030  062 025 073     5963          STA     TEMPFB+FB.NAM   CLEAR TEMP FILE NAME          /80.03.GC/
074.033  041 010 102     5964          LXI     H,LINE
                         5965
074.036  176             5966  PRS4    MOV     A,M             CHECK FOR '='
074.037  376 075         5967          CPI     '='
074.041  043             5968          INX     H
074.042  312 063 074     5969          JE      PRS5            GOT '='
074.045  247             5970          ANA     A
074.046  302 036 074     5971          JNZ     PRS4            MORE TO CHECK
074.051  041 010 102     5972          LXI     H,LINE          NO LISTING OR BINARY         /WCZ062780/
074.054  315 142 101     5973          CALL    $SOB            SKIP OVER BLANKS AT BEGINNING /WCZ062780/
074.057  353             5974          XCHG                    MOVE POINTER TO REG DE       /WCZ062780/
074.060  303 204 074     5975          JMP     PRS7
                         5976
                         5977  *       HAVE '='. HAS SPECIFIED LISTING AND/OR BINARY
                         5978
074.063  041 010 102     5979  PRS5    LXI     H,LINE          POINTER TO COMMAND LINE      /WCZ062780/
074.066  315 142 101     5980          CALL    $SOB            SKIP OVER BLANKS AT BEGINNING /WCZ062780/
074.071  353             5981          XCHG                    MOVE POINTER TO REG DE       /WCZ062780/
074.072  041 214 072     5982          LXI     H,BINFNAM
074.075  315 207 064     5983          CALL    $CPF            COPY FILE NAME
074.100  332 367 074     5984          JC      PRS10           FORMAT ERROR
074.103  376 054         5985          CPI     ','
074.105  302 171 074     5986          JNE     PRS6            NOT LISTING FILE
                         5987
074.110  041 304 072     5988          LXI     H,LISTFB+FB.NAM
074.113  315 207 064     5989          CALL    $CPF            COPY FILE NAME
074.116  332 367 074     5990          JC      PRS10           FNAME ERROR
074.121  376 054         5991          CPI     ','                                          /80.03.GC/
074.123  302 171 074     5992          JNE     PRS6                                         /80.03.GC/
                         5993                                                               /80.03.GC/
074.126  041 025 073     5994          LXI     H,TEMPFB+FB.NAM                              /80.03.GC/
074.131  315 207 064     5995          CALL    $CPF            GET TEMP FILE NAME           /80.03.GC/
074.134  332 367 074     5996          JC      PRS10                                        /80.03.GC/
                         5997
                         5998  *  DETERMINE IF TEMP FILE IS ON A DIRECTORY TYPE DEVICE.     /WCZ062480/
                         5999
074.137  345             6000          PUSH    H
074.140  325             6001          PUSH    D
074.141  365             6002          PUSH    PSW
074.142  041 025 073     6003          LXI     H,TEMPFB+FB.NAM
074.145  021 001 102     6004          LXI     D,DEFALT
074.150  001 173 073     6005          LXI     B,EXPWRK
074.153  377 053         6006          DB      SYSCALL+.DECODE GET DEVICE INFO ABOUT TEMP FILE
074.155  072 173 073     6007          LDA     EXPWRK          (A) = DEVICE CODE
074.160  346 001         6008          ANI     DT.DD
074.162  301             6009          POP     B
074.163  170             6010          MOV     A,B
074.164  321             6011          POP     D
074.165  341             6012          POP     H
074.166  312 046 075     6013          JZ      PRS12           NOT DIRECTORY DEVICE         /WCZ062480/
                         6014
074.171  376 075         6015  PRS6    CPI     '='
```

```
074.173  302 020 075  6016          JNE     PRS11           FORMAT ERROR
074.176  315 177 075  6017          CALL    OOF             OPEN OUTPUT FILES
074.201  332 342 073  6018          JC      PRS2            ERROR
                      6019
                      6020  *       CRACK SOURCE FILE LIST.
                      6021
074.204  041 337 072  6022  PRS7    LXI     H,SORCFB+FB.NAM
074.207  315 207 064  6023          CALL    $CPF            COPY FILE NAME
                      6024
074.212  376 054      6025          CPI     ','             Q. XTEXT DEVICES POSSIBLY       /WCZ062780/
074.214  314 041 076  6026          CZ      XTI                CALL IF YES                  /WCZ062780/
                      6027
074.217  041 337 072  6028          LXI     H,SORCFB+FB.NAM
074.222  021 373 101  6029          LXI     D,DEFALTI
074.225  001 173 073  6030          LXI     B,EXPWRK
074.230  377 053      6031          DB      SYSCALL,.DECODE         GET DEVICE INFO ABOUT INPUT FILE
074.232  072 173 073  6032          LDA     EXPWRK+0        (A) = DEVICE CODE
074.235  346 001      6033          ANI     DT.DD
074.237  312 300 074  6034          JZ      PRS9            NOT DIRECTORY DEVICE
074.242  041 325 072  6035          LXI     H,SORCFB
074.245  021 373 101  6036          LXI     D,DEFALTI       (DE) = INPUT DEFAULT POINTER
074.250  315 362 064  6037          CALL    $FOPER
074.253  322 264 074  6038          JNC     PRS8            ALL OK
074.256  315 314 075  6039          CALL    PFE             PRESET FILE ERROR
074.261  303 342 073  6040          JMP     PRS2            RE-TRY
                      6041
                      6042  *       GET THE CURRENT DATE
                      6043
074.264              6044  PRS8    EQU     *
074.264  315 164 064  6045          CALL    $MOVEL                                          /79.12.GC/
074.267  011 000      6046          DW      HEADCL                                          /79.12.GC/
074.271  277 040      6047          DW      S.DATE                                          /79.12.GC/
074.273  072 072      6048          DW      HEADC                                           /79.12.GC/
                      6049
074.275  303 200 042  6050          JMP     START           START ASSEMBLY
                      6051
                      6052  *       INPUT FILE NOT ON DIRECTORY DEVICE
                      6053
074.300  315 136 031  6054  PRS9    CALL    $TYPTX
074.303  007 123 157  6055          DB      BELL,'Source File Must be on Mounted Directory Device',200Q
                      6056  *                                                               /80.03.GC/
074.364  303 342 073  6057          JMP     PRS2            TRY AGAIN
                      6058
                      6059  *       ERROR IN FILE NAME
                      6060
074.367  315 136 031  6061  PRS10   CALL    $TYPTX
074.372  007 111 154  6062          DB      BELL,'Illegal File Name',200Q
075.015  303 342 073  6063          JMP     PRS2            TRY AGAIN
                      6064
                      6065  *       ILLEGAL SYNTAX
                      6066
075.020  315 136 031  6067  PRS11   CALL    $TYPTX
075.023  007 111 154  6068          DB      BELL,'Illegal Syntax',200Q
075.043  303 342 073  6069          JMP     PRS2
                      6070
                      6071  *       TEMP FILE NOT ON DIRECTORY DEVICE.                      /WCZ062480/
```

```
                                6072
075.046  315 136 031   6073  PRS12   CALL    $TYPTX
075.051  007 130 122   6074          DB      BELL,'XREF Temp File Must be on Mounted '
075.114  104 151 162   6075          DB      'Directory Device',200Q
075.135  303 342 073   6076          JMP     PRS2            TRY AGAIN                       /WCZ062480/
                                6077
                                6078  *       SWITCH ERROR
                                6079
075.140  315 136 031   6080  SW.ERR  CALL    $TYPTX
075.143  007 111 154   6081          DB      BELL,'Illegal Switch',ENL
075.163  303 342 073   6082          JMP     PRS2
                                6083
075.166  076 050        6084  PRSERR1 MVI     A,EC.NCV                NOT CORRECT VERSION OF HDOS
                                6085
075.170  046 012        6086  PRSERR  MVI     H,NL
075.172  377 057        6087          DB      SYSCALL,.ERROR
075.174  257            6088          XRA     A
075.175  377 000        6089          DB      SYSCALL,.EXIT
                                6090
```

```
                                6094  **      OOF - OPEN OUTPUT FILES.
                                6095  *
                                6096  *       OOF IS CALLED TO OPEN THE BINARY AND LISTINF FILES,
                                6097  *       TO THEIR RESPECTIVE CHANNELS.
                                6098  *
                                6099  *       ENTRY   BINFNAM = BINARY FILE NAME (=0 IF NONE)
                                6100  *               LISTFB+FB.NAM = LISTING FILE NAME (=0 IF NONE)
                                6101  *               TEMPFB+FB.NAM = TEMP FILE NAME (=0 IF DEFAULT)  /80.03.GC/
                                6102  *       EXIT    'C' CLEAR IF OK
                                6103  *               'C' SET IF ERROR
                                6104  *                ERROR IS MESSAGED BY OOF
                                6105  *       USES    A,F
                                6106
                                6107
075.177  305                   6108  OOF     PUSH    B               SAVE REGISTERS
075.200  325                   6109          PUSH    D
075.201  345                   6110          PUSH    H
075.202  041 214 072           6111          LXI     H,BINFNAM
075.205  176                   6112          MOV     A,M
075.206  247                   6113          ANA     A
075.207  312 227 075           6114          JZ      OOF1            NO BINARY
                                6115
                                6116  *       OPEN BINARY FILE
                                6117
075.212  021 357 101           6118          LXI     D,DEFALTB
075.215  076 000               6119          MVI     A,CN.BIN
075.217  377 043               6120          DB      SYSCALL,.OPENW
075.221  041 202 072           6121          LXI     H,BINFNAM-FB.NAM
075.224  332 304 075           6122          JC      OOF3            ERROR
                                6123
                                6124  *       OPEN LISTING FILE
                                6125
075.227  041 272 072           6126  OOF1    LXI     H,LISTFB
075.232  072 304 072           6127          LDA     LISTFB+FB.NAM
075.235  247                   6128          ANA     A
075.236  312 255 075           6129          JZ      OOF1.5          NO LIST FILE (FORCE NO XREF)    /WCZ062580/
075.241  021 365 101           6130          LXI     D,DEFALTL
075.244  315 365 064           6131          CALL    $FOPEW,         OPEN FOR WRITE
075.247  332 304 075           6132          JC      OOF3            ERROR                           /80.03.GC/
075.252  303 261 075           6133          JMP     OOF2                                            /WCZ062580/
                                6134
075.255  257                   6135  OOF1.5  XRA     A               CAN'T HAVE TEMP FILE WITHOUT    /WCZ062580/
075.256  062 025 073           6136          STA     TEMPFB+FB.NAM     LIST FILE                     /WCZ062580/
                                6137
                                6138  *       OPEN TEMP FILE                                        /80.03.GC/
                                6139
075.261  041 013 073           6140  OOF2    LXI     H,TEMPFB                                        /80.03.GC/
075.264  072 025 073           6141          LDA     TEMPFB+FB.NAM                                   /80.03.GC/
075.267  247                   6142          ANA     A                                               /80.03.GC/
075.270  312 310 075           6143          JZ      OOF4            BR IF NO FILE SPECIFIED         /WCZ063080/
075.273  021 001 102           6144          LXI     D,DEFALTT                                       /80.03.GC/
075.276  315 344 064           6145          CALL    $FOPEW          OPEN FOR UPDATE                 /80.03.GC/
075.301  322 310 075           6146          JNC     OOF4            NO ERROR                        /WCZ063080/
                                6147
                                6148  *       ERROR IN FILE
                                6149
```

```
075.304  315 314 075  6150  00F3  CALL  PFE          PRESET FILE ERROR
075.307  067          6151        STC
075.310  341          6152  00F4  POP   H
075.311  321          6153        POP   D
075.312  301          6154        POP   B
075.313  311          6155        RET


                      6157  **    PFE - PRESET FILE ERROR.
                      6158  *
                      6159  *     PFE IS CALLED TO PRINT AN ERROR MESSAGE.
                      6160  *
                      6161  *     ENTRY  (A) = CODE
                      6162  *            (HL) = FILE BLOCK ADDRESS (ONLY USED TO GET FB.NAM)
                      6163  *     EXIT   NONE
                      6164  *     USES   ALL
                      6165
                      6166
075.314  365          6167  PFE   PUSH  PSW          SAVE CODE
075.315  315 136 031  6168        CALL  $TYPTX
075.320  007 105 162  6169        DB    BELL,'Error On File',' '+200Q
075.337  001 012 000  6170        LXI   B,FB.NAM
075.342  011          6171        DAD   B
075.343  315 074 064  6172        CALL  $TYPLZ        TYPE FNAME
075.346  361          6173        POP   PSW
075.347  046 012      6174        MVI   H,NL
075.351  377 057      6175        DB    SYSCALL,.ERROR  PRINT ERROR MEANING
075.353  311          6176        RET


                      6178  **    SDV - SET DEFAULT SWITCH VALUES.
                      6179  *
                      6180  *     SDV IS CALLED BY PRS TO SET ALL SWITCH FLAGS TO THEIR DEFAULT
                      6181  *     VALUES. THEIR VALUES CANNOT BE SIMPLY ASSEMBLED IN, BECUASE
                      6182  *     AN INCORRECT COMMAND LINE SWITCH MAY CHANGE THEM, BEFORE
                      6183  *     THE ERROR IS DETECTED. SDV RESETS THEM FOR THE
                      6184  *     NEXT TRY.
                      6185  *
                      6186  *     ENTRY  NONE
                      6187  *     EXIT   NONE
                      6188  *     USES   A,F,H,L
                      6189
                      6190
075.354  076 074      6191  SDV   MVI   A,60
075.356  062 260 072  6192        STA   PAGEDP        SET PAGE DEPTH
075.361  257          6193        XRA   A
075.362  062 261 072  6194        STA   FORMDP        USE PAGE FORM CONTROL
075.365  062 257 072  6195        STA   WIDE          CLEAR /WIDE SWITCH
075.370  062 173 072  6196        STA   LSTCTLS
075.373  057          6197        CMA
075.374  062 174 072  6198        STA   LSTCTLC
075.377  315 164 064  6199        CALL  $MOVEL
```

```
076.002  030 000 011  6200          DW      SDVB-SDVA,SDVA,DEFALTB                  /80.03.GC/
000.000               6201          ERRNZ   DEFALTI-DEFALTB-6
000.000               6202          ERRNZ   DEFALTI-DEFALTL-6
000.000               6203          ERRNZ   DEFALTI-DEFALTI-6                       /80.03.GC/
076.010  311          6204          RET
                      6205
076.011  123 131 060  6206  SDVA    DB      'SYOABS'        DEFAULT FOR BINARY
076.017  123 131 060  6207          DB      'SYOLST'        DEFAULT FOR LISTING
076.025  123 131 060  6208          DB      'SYOASM'        DEFAULT FOR INPUT
076.033  123 131 060  6209  SDVA3   DB      'SYOTMP'        DEFAULT FOR TEMP        /80.03.GC/
076.041               6210  SDVB    EQU     *               DEFAULT FOR BINARY      /80.03.GC/


                      6212  **      XTI - PICKUP XTEXT DEFAULT DRIVES SPECIFY ORDER  /WCZ062780/
                      6213  *               OF SEARCH FOR XTEXT FILES.
                      6214  *
                      6215  *       ENTRY   (DE) = POINTER TO NEXT CHARACTER IN LINE
                      6216  *       EXIT    (DE) UPDATED
                      6217  *               (A) = DELIMITER
                      6218  *               TABLE AT XTEXTH IS FILLED IN AND XTEXTE
                      6219  *                 IS FILLED IN WITH THE NUMBER OF DEFAULTS.
                      6220  *       USES    F,B,C,H,L
                      6221  *
                      6222
076.041               6223  XTI     EQU     *
                      6224
076.041  062 046 077  6225          STA     XTIA            SAVE DELIMITER
                      6226
076.044               6227  XTI0    EQU     *
076.044  072 046 077  6228          LDA     XTIA
076.047  376 054      6229          CPI     ','             CHECK DELIMITER
076.051  300          6230          RNZ                     NO MORE TO GO
                      6231
076.052  001 005 000  6232          LXI     B,5             ONLY ALLOW DEVICE NAME 'DDX:' OR 'DD:'
000.014               6233          ERRMI   FB.NAML-5
076.055  041 372 072  6234          LXI     H,XTXFB+FB.NAM
076.060  315 211 064  6235          CALL    $CPF1           COPY FILENAME (DEVICE NAME ONLY)
076.063  332 226 076  6236          JC      XTI6            TOO LONG, THEREFORE ERROR
076.066  062 046 077  6237          STA     XTIA            SAVE DELIMITER
076.071  353          6238          XCHG
076.072  042 047 077  6239          SHLD    XTIC            SAVE COMMAND LINE POINTER
076.075  353          6240          XCHG
076.076  072 375 072  6241          LDA     XTXFB+FB.NAM+3  Q. FORM FOR DEVICE NAME MUST BE
076.101  247          6242          ANA     A                 DEVICE NAME
076.102  312 112 076  6243          JZ      XTI0.5              MUST BE
076.105  376 072      6244          CPI     ':'                   DDX: OR DD:
076.107  302 226 076  6245          JNZ     XTI6                    BR IF DEFINITELY NOT
076.112               6246  XTI0.5  EQU     *
076.112  066 130      6247          MVI     M,'X'           PLACE DUMMY NAME AFTER DEVICE NAME
076.114  043          6248          INX     H               SO DECODE WILL BE HAPPY
076.115  066 000      6249          MVI     M,0             PLACE DELIMITER AFTER FILENAME
000.013               6250          ERRMI   FB.NAML-6
                      6251
076.117  041 372 072  6252          LXI     H,XTXFB+FB.NAM
```

```
076.122  021 051 077  6253          LXI     D,XTID
076.125  001 173 073  6254          LXI     B,EXPWRK
076.130  377 053      6255          DB      SYSCALL+.DECODE CHECK IF VALID FILENAME BY DECODING
076.132  332 226 076  6256          JC      XTI6                BR IF NOT
                      6257
076.135  072 173 073  6258          LDA     EXPWRK
076.140  346 001      6259          ANI     DT.DD
076.142  312 262 076  6260          JZ      XTI7            BR IF NOT DIRECTORY TYPE DEVICE
                      6261
076.145  072 176 073  6262          LDA     EXPWRK+3        CONVERT UNIT NUMBER
076.150  306 060      6263          ADI     '0'               FROM BINARY FORM
076.152  062 176 073  6264          STA     EXPWRK+3            TO CHARACTER FORM
                      6265
                      6266  *       ADD DEVICE TO TABLE.
                      6267
076.155  072 023 051  6268          LDA     XTEXTE
076.160  376 005      6269          CPI     XTEXTF
076.162  322 350 076  6270          JNC     XTI8            TABLE IS FULL
                      6271
076.165  107          6272          MOV     B,A             CALCULATE
076.166  207          6273          ADD     A
076.167  200          6274          ADD     B                 ADDRESS
000.000               6275          ERRNZ   XTEXTG-3
076.170  117          6276          MOV     C,A
076.171  006 000      6277          MVI     B,0                 OF WHERE
076.173  041 024 051  6278          LXI     H,XTEXTH              TO PUT
076.176  011          6279          DAD     B                       DEVICE NAME
076.177  021 174 073  6280          LXI     D,EXPWRK+1
076.202  001 003 000  6281          LXI     B,XTEXTG
076.205  315 252 030  6282          CALL    $MOVE           MOVE DEVICE NAME TO TABLE
                      6283
076.210  072 023 051  6284          LDA     XTEXTE
076.213  074          6285          INR     A
076.214  062 023 051  6286          STA     XTEXTE          ADD 1 TO NUMBER OF DEVICES
                      6287
                      6288  *       GET READY TO PARSE FOR ANOTHER POSSIBLE DEVICE.
                      6289
076.217  052 047 077  6290          LHLD    XTIC
076.222  353          6291          XCHG                    RESTORE LINE POINTER
076.223  303 044 076  6292          JMP     XTI0
                      6293
                      6294  *       ERROR BAD DEVICE NAME.
                      6295
076.226               6296  XTI6    EQU     *
076.226  315 136 031  6297          CALL    $TYPTX
076.231  102 141 144  6298          DB      'Bad XTEXT Device Name',BELL+200Q
076.257  303 041 077  6299          JMP     XTI9
                      6300
                      6301  *       DEVICE SPECIFIED IS NOT A DIRECTORY TYPE DEVICE.
                      6302
076.262               6303  XTI7    EQU     *
076.262  315 136 031  6304          CALL    $TYPTX
076.265  130 124 105  6305          DB      'XTEXT Device must be a Mounted '
076.324  104 151 162  6306          DB      'Directory Device',BELL+200Q
076.345  303 041 077  6307          JMP     XTI9
                      6308
```

```
                          6309  *       XTEXT DEFAULT DEVICE TABLE IS FULL.
                          6310
076.350                   6311  XTI8    EQU     *
076.350  315 136 031      6312          CALL    $TYPTX
076.353  124 157 157      6313          DB      'Too Many XTEXT Devices Specified -- '
077.017  115 141 170      6314          DB      'Maximum is ',BELL+200Q
077.033  076 005          6315          MVI     A,XTEXTF
077.035  306 060          6316          ADI     '0'
000.004                   6317          ERRMI   9-XTEXTF
077.037  377 002          6318          DB      SYSCALL,.SCOUT
000.000                   6319          ERRNZ   XTI9-*
                          6320
                          6321  *       ERROR ENCOUNTERED.
                          6322
077.041                   6323  XTI9    EQU     *
077.041  343              6324          XTHL                    POP RETURN ADDRESS
077.042  341              6325          POP     H                 AND DISCARD
077.043  303 342 073      6326          JMP     PRS2            TRY AGAIN
                          6327
077.046                   6328  XTIA    DS      1
077.047                   6329  XTIC    DS      2
077.051  000 000 000      6330  XTID    DB      0,0,0,0,0,0
```

```
                        6333  **      SWITCH TABLE.
                        6334  *
                        6335  *       THIS TABLE CONTAINS DESCRIPTIONS FOR COMMAND SWITCHES.
                        6336  *       SEE '$DRS' FOR A DESCRIPTION OF IT'S FORMAT.
                        6337
                        6338
077.057                 6339  SWITAB  DS      0
                        6340
                        6341  *       /PAGE:NN
                        6342
077.057 120 301 307     6343          DB      'P','A'+200Q,'G'+200Q,'E'+200Q,200Q
077.064 137 077         6344          DW      SW.PAG
                        6345
                        6346  *       /FORM:NN
                        6347
077.066 106 317 322     6348          DB      'F','O'+200Q,'R'+200Q,'M'+200Q,200Q
077.073 213 077         6349          DW      SW.FOR
                        6350
                        6351  *       /WIDE
                        6352
077.075 127 311 304     6353          DB      'W','I'+200Q,'D'+200Q,'E'+200Q,200Q
077.102 271 077         6354          DW      SW.WID
                        6355
                        6356  *       /LON:CCC
                        6357
077.104 114 117 116     6358          DB      'LON',200Q
077.110 125 100         6359          DW      SW.LON
                        6360
                        6361  *       /LOF:CCC
                        6362
077.112 114 117 106     6363          DB      'LOF',200Q
077.116 202 100         6364          DW      SW.LOF
                        6365
                        6366  *       /LARGE
                        6367
077.120 114 101 322     6368          DB      'LA','R'+200Q,'G'+200Q,'E'+200Q,200Q
077.126 355 077         6369          DW      SW.LAR
                        6370
                        6371  *       /ERR
                        6372
077.130 105 122 122     6373          DB      'ERR',200Q
077.134 042 100         6374          DW      SW.ERS
                        6375
077.136 000             6376          DB      0               END OF TABLE


                        6378  **      SW.PAG - /PAGE:NN
                        6379
077.137 315 103 101     6380  SW.PAG  CALL    $DNS.           DECODE NUMERIC SWITCHES
077.142 332 153 077     6381          JC      SW.PAG1         ERROR
077.145 173             6382          MOV     A,E             'A' = VALUE
077.146 247             6383          ANA     A
077.147 062 260 072     6384          STA     PAGEDP
077.152 300             6385          RNZ                     ;0 IS ILLEGAL
```

```
077.153  315 136 031  6386  SW.PAG1 CALL    $TYPTX
077.156  042 057 120  6387          DB      '"/PAGE:" Value is No Good',' '+200Q
077.210  303 140 075  6388          JMP     SW.ERR


                      6390  **      SW.FOR - /FORM:NN
                      6391
077.213  315 103 101  6392  SW.FOR  CALL    $DNS.           DECODE DECIMAL SWITCH
077.216  332 227 077  6393          JC      SW.FOR1
077.221  173          6394          MOV     A,E             (A) = VALUE
077.222  062 261 072  6395          STA     FORMDP
077.225  247          6396          ANA     A
077.226  300          6397          RNZ                     VALUE OF 0 ILLEGAL
077.227  315 136 031  6398  SW.FOR1 CALL    $TYPTX
077.232  042 057 106  6399          DB      '"/FORM:" Value is No Good -',' '+200Q
077.266  303 140 075  6400          JMP     SW.ERR


                      6402  **      SW.WID - /WIDE
                      6403
077.271  312 302 077  6404  SW.WID  JE      SW.WID1         NO VALUE ALLOWED
077.274  076 001      6405          MVI     A,1
077.276  062 257 072  6406          STA     WIDE
077.301  311          6407          RET
                      6408
077.302  315 136 031  6409  SW.WID1 CALL    $TYPTX
077.305  111 155 160  6410          DB      'Improper Format For "/WIDE" Switch -',' '+200Q
077.352  303 140 075  6411          JMP     SW.ERR


                      6413  **      SW.LAR - /LARGE
                      6414
077.355  312 366 077  6415  SW.LAR  JE      SW.LAR1         NO VALUE ALLOWED
077.360  076 001      6416          MVI     A,1
077.362  062 213 072  6417          STA     LARGE
077.365  311          6418          RET
                      6419
077.366  315 136 031  6420  SW.LAR1 CALL    $TYPTX
077.371  111 155 160  6421          DB      'Improper Format for "/LARGE" Switch -',' '+200Q
100.037  303 140 075  6422          JMP     SW.ERR


                      6424  **      SW.ERS - /ERR
                      6425
100.042  312 053 100  6426  SW.ERS  JE      SW.ERS1         NO VALUE ALLOWED
100.045  076 001      6427          MVI     A,1
100.047  062 167 072  6428          STA     ERRSHO
100.052  311          6429          RET
                      6430
100.053  315 136 031  6431  SW.ERS1 CALL    $TYPTX
100.056  111 155 160  6432          DB      'Improper Format for "/ERR" Switch -',' '+200Q
100.122  303 140 075  6433          JMP     SW.ERR
```

```
                                6435  **      SW.LON - /LON:CCC
                                6436
100.125  315 260 100            6437  SW.LON  CALL    DLS             DECODE LISTING SWITCHES
100.130  332 141 100            6438          JC      SW.LON1
100.133  041 173 072            6439          LXI     H,LSTCTLS
100.136  266                    6440          ORA     M
100.137  167                    6441          MOV     M,A             SET SWITCHES
100.140  311                    6442          RET
                                6443
100.141  315 136 031            6444  SW.LON1 CALL    $TYPTX
100.144  042 057 114            6445          DB      '*/LON:* Value is No Good -',' '+200Q
100.177  303 140 075            6446          JMP     SW.ERR


                                6448  **      SW.LOF - /LOF:CCC
                                6449
100.202  315 260 100            6450  SW.LOF  CALL    DLS             DECODE LISTING SWITCHES
100.205  332 217 100            6451          JC      SW.LOF1
100.210  041 174 072            6452          LXI     H,LSTCTLC
100.213  057                    6453          CMA
100.214  246                    6454          ANA     M
100.215  167                    6455          MOV     M,A             SET 0 BITS FOR SPECIFED OPTIONS
100.216  311                    6456          RET
                                6457
100.217  315 136 031            6458  SW.LOF1 CALL    $TYPTX
100.222  042 057 114            6459          DB      '*/LOF:* Value is No Good -',' '+200Q
100.255  303 140 075            6460          JMP     SW.ERR


                                6462  **      DLS - DECODE LISTING SWITCHES.
                                6463  *
                                6464  *       DLS IS CALLED TO DECODE THE SPECIFIED LIST SUBOPTIONS FOR THE /LON
                                6465  *       AND THE /LOF SWITCHES.
                                6466  *
                                6467  *       THE OPTIONS ARE ANALYZED, AND REPLACED WITH BLANKS
                                6468  *
                                6469  *       ENTRY   (HL) = ADDRESS OF ':CCC'
                                6470  *       EXIT    'C' CLEAR IF OK
                                6471  *                (A) = BITS SET FOR EACH SPECIFED OPTION
                                6472  *               'C' SET IF ERROR
                                6473  *       USES    ALL
                                6474
                                6475
100.260  176                    6476  DLS     MOV     A,M             (A) = SUPPOSED ':'
100.261  376 072                6477          CPI     ':'             CHECK UP ON HIM
100.263  067                    6478          STC
100.264  300                    6479          RNE                     NOT ':'
100.265  353                    6480          XCHG                    (DE) = ADDRESS
100.266  006 000                6481          MVI     B,0
                                6482
                                6483  *       DECODE NEXT SWITCH
                                6484
100.270  076 040                6485  DLS1    MVI     A,' '
```

```
100.272  022           6486           STAX    D               CLEAR THAT ONE
100.273  023           6487           INX     D
100.274  032           6488           LDAX    D
100.275  376 040       6489           CPI     ' '
100.277  312 270 100   6490           JE      DLS1            SKIP
100.302  376 057       6491           CPI     '/'
100.304  312 336 100   6492           JE      DLS2            DONE
100.307  376 054       6493           CPI     ','
100.311  312 336 100   6494           JE      DLS2            DONE
100.314  247           6495           ANA     A
100.315  312 336 100   6496           JZ      DLS2            DONE
                       6497
                       6498  *        MUST BE A SUBOPTION
                       6499
100.320  041 332 046   6500           LXI     H,LSTA
100.323  315 304 064   6501           CALL    $TBLS
100.326  067           6502           STC
100.327  300           6503           RNZ                     NOT A GOOD OPTION
100.330  176           6504           MOV     A,M
100.331  260           6505           ORA     B               SET FLAGS
100.332  107           6506           MOV     B,A
100.333  303 270 100   6507           JMP     DLS1            GET ANOTHER
                       6508
                       6509  *        ALL DONE
                       6510
100.336  170           6511  DLS2     MOV     A,B
100.337  247           6512           ANA     A
100.340  067           6513           STC
100.341  310           6514           RZ                      NONE FOUND: ERROR
100.342  247           6515           ANA     A               CLEAR CARRY
100.343  311           6516           RET                     RETURN WITH OK
```

```
100.344                 6519          XTEXT   CVD


                        6521X **      $CVD - CHECK FOR VALID DIGIT.
                        6522X *
                        6523X *       CVD EXAMINES A DIGIT TO SEE IF IT IS A VALID DECIMAL DIGIT.
                        6524X *
                        6525X *       ENTRY   (HL) = ADDRESS OF CHARACTER
                        6526X *       EXIT    'C' SET IF ILLEGAL
                        6527X *                (A) = VALUE
                        6528X *       USES    A,F
                        6529X
                        6530X
100.344  176           6531X $CVD     MOV     A,M             (A) = CHARACTER
100.345  326 060       6532X $CVD.    SUI     '0'
100.347  330           6533X          RC                      ILLEGAL
100.350  376 012       6534X          CPI     9+1
100.352  077           6535X          CMC
100.353  311           6536X          RET
100.354                 6537          XTEXT   MU86


                        6539X **      $MU86 - MULTIPLY 8X16 UNSIGNED.
                        6540X *
                        6541X *       $MU86 MULTIPLIES A 16 BIT VALUE BY A 8
                        6542X *       BIT VALUE.
                        6543X *
                        6544X *       ENTRY   (A) = MULTIPLIER
                        6545X *               (DE) = MULTIPLICAND
                        6546X *       EXIT    (HL) = RESULT
                        6547X *               'Z' SET IF NOT OVERFLOW
                        6548X *       USES    A,F,H,L
                        6549X
                        6550X
031.007                 6551X $MU86   EQU     31007A          IN H17 ROM
100.354                 6552          XTEXT   DNV


                        6554X **      $DNV - DECODE NUMERIC VALUE.
                        6555X *
                        6556X *       $DNV DECODES A NUMERIC VALUE (IN THE FORM OF AN ASCII STRING)
                        6557X *       INTO A BINARY NUMBER. THE MAXIMUM MAGNITUDE IS
                        6558X *       65535D.
                        6559X *
                        6560X *       THE NUMBER MAY CONTAIN A POSTRADIX OF 'B' (BINARY)
                        6561X *       'O' OR 'Q' (OCTAL) OR 'D' (DECIMAL)
                        6562X *
                        6563X *       ENTRY   (HL) = ADDRESS OF FIRST BYTE OF NUMBER
                        6564X *               (A) = DEFAULT BASE (2 FOR BINARY, 10 FOR DECIMAL, ETC.)
                        6565X *       EXIT    'C' CLEAR IF OK
```

```
                                6566X *              (HL) ADVANCED PAST NUMBER (AND POSTRADIX)
                                6567X *              (DE) = VALUE
                                6568X *              'C' SET IF ERROR
                                6569X *       USES   ALL
                                6570X
                                6571X
100.354  062 071 101            6572X $DNV    STA    $DNVA           SET DEFAULT BASE
100.357  104                    6573X         MOV    B,H
100.360  115                    6574X         MOV    C,L             (BC) = TEXT ADDRESS
                                6575X
                                6576X *       SCAN FOR POSTRADIX
                                6577X
100.361  176                    6578X $DNV1   MOV    A,M
100.362  315 345 100            6579X         CALL   $CVD.           CHECK FOR VALID DECIMAL DIGIT
100.365  043                    6580X         INX    H
100.366  322 361 100            6581X         JNC    $DNV1           MORE TO GO
100.371  053                    6582X         DCX    H               REMOVE EXTRA INCREMENT
100.372  171                    6583X         MOV    A,C
100.373  275                    6584X         CMP    L               SEE IF THERE WERE ANY NUMBERS
100.374  067                    6585X         STC                    ASSUME NOT
100.375  310                    6586X         RE                     ERROR
                                6587X
                                6588X *       OUT OF NUMBERS. SEE IF POSTRADIX FOLLOWS
                                6589X
100.376  176                    6590X         MOV    A,M             (A) = PROPOSED POSTRADIX
100.377  345                    6591X         PUSH   H               SAVE END ADDRESS
101.000  041 072 101            6592X         LXI    H,$DNVB
101.003  247                    6593X         ANA    A
101.004  312 024 101            6594X         JZ     $DNV2           NO POSTRADIX
101.007  315 304 064            6595X         CALL   $TBLS
101.012  176                    6596X         MOV    A,M
101.013  302 024 101            6597X         JNE    $DNV2           NOT POSTRADIX
101.016  341                    6598X         POP    H
101.017  043                    6599X         INX    H               SKIP POSTRADIX
101.020  345                    6600X         PUSH   H
101.021  062 071 101            6601X         STA    $DNVA           SET NEW POSTRADIX
101.024  021 000 000            6602X $DNV2   LXI    D,0             (DE) = ACCUMULATOR
                                6603X
                                6604X *       BUILD NUMBER
                                6605X
101.027  072 071 101            6606X $DNV3   LDA    $DNVA           (A) = BASE
101.032  365                    6607X         PUSH   PSW             SAVE BASE
101.033  315 007 031            6608X         CALL   $MU86           MULTIPLY
101.036  321                    6609X         POP    D               (D) = BASE
101.037  332 067 101            6610X         JC     $DNV4           OVERFLOW
101.042  012                    6611X         LDAX   B               (A) = DIGIT
101.043  326 060                6612X         SUI    '0'
101.045  003                    6613X         INX    B
101.046  272                    6614X         CMP    D               COMPARE TO BASE
101.047  077                    6615X         CMC
101.050  332 067 101            6616X         JC     $DNV4           TOO LARGE A DIGIT
101.053  315 101 030            6617X         CALL   $DADA.          ADD TO VALUE
101.056  353                    6618X         XCHG                   (DE) = VALUE
101.057  012                    6619X         LDAX   B
101.060  315 345 100            6620X         CALL   $CVD.
101.063  322 027 101            6621X         JNC    $DNV3           MORE TO GO
```

```
101.066  247           6622X          ANA     A               CLEAR CARRY
101.067  341           6623X $DNV4    POP     H               RESTORE POINTER
101.070  311           6624X          RET                     EXIT
                       6625X
101.071  000           6626X $DNVA    DB      0               DEFAULT BASE
101.072  102 002       6627X $DNVB    DB      'B',2           POSTRADIX TABLE
101.074  117 010       6628X          DB      'O',8
101.076  121 010       6629X          DB      'Q',8
101.100  104 012       6630X          DB      'D',10
101.102  000           6631X          DB      0
101.103               6632           XTEXT   DNS


                       6634X **       $DNS - DECODE NUMERIC SWITCH.
                       6635X *
                       6636X *        $DNS DECODES A NUMERIC SWITCH OF THE FORM:
                       6637X *
                       6638X *        !NNN
                       6639X *
                       6640X *        A POSTRADIX OF D, Q, O, OR B IS ALLOWED. IF THE VALUE
                       6641X *        IS SYNTATICALLY VALID, IT IS REPLACED WITH BLANKS.
                       6642X *
                       6643X *        ENTRY   (HL) = ADDRESS IF '!'
                       6644X *                (A) = DEFAULT BASE (2, 8 OR 10)
                       6645X *        EXIT    'C' CLEAR IF OK
                       6646X *                 (HL) ADVANCED PAST VALUE
                       6647X *                 VALUE BLANKED
                       6648X *                 (DE) = VALUE
                       6649X *                 'C' SET IF ERROR
                       6650X *        USES    ALL
                       6651X
                       6652X
101.103  076 012       6653X $DNS.    MVI     A,10            BASE 10 DEFAULT
101.105  107           6654X $DNS     MOV     B,A             (B) = DEFAULT BASE
101.106  176           6655X          MOV     A,M
101.107  376 072       6656X          CPI     '!'
101.111  067           6657X          STC
101.112  300           6658X          RNE                     NOT '!'
101.113  345           6659X          PUSH    H               SAVE ADDRESS OF SWITCH START
101.114  043           6660X          INX     H
101.115  170           6661X          MOV     A,B
101.116  315 354 100   6662X          CALL    $DNV            DECODE NUMERIC VALUE
101.121  301           6663X          POP     B               (BC) = ADDRESS OF '!'
101.122  330           6664X          RC                      ERROR
101.123  076 040       6665X $DNS1    MVI     A,' '
101.125  002           6666X          STAX    B               BLANK LINE
101.126  003           6667X          INX     B               INCREMENT ADDRESS
101.127  175           6668X          MOV     A,L
101.130  271           6669X          CMP     C
101.131  302 123 101   6670X          JNE     $DNS1
101.134  170           6671X          MOV     A,B
101.135  274           6672X          CMP     H               SEE IF IN RIGHT BANK
101.136  302 123 101   6673X          JNE     $DNS1
101.141  311           6674X          RET                     RETURN WITH 'C' CLEAR AND VALUE
```

```
101.142                 6675          XTEXT   SOB


                        6677X **      $SOB - SKIP OVER BLANKS.
                        6678X *
                        6679X *       $SOB IS CALLED TO SKIP AN ARBITRARILY LONG STRING OF BLANKS AND TABS.
                        6680X *
                        6681X *       ENTRY   (HL) = FWA OF (POSSIBLE) BLANK STRING
                        6682X *       EXIT    (HL) = LWA+1 OF BLANK STRING (UNCHANGED IF NO BLANKS)
                        6683X *               (A) = FIRST NON-BLANK, NON-TAB CHARACTER EEN
                        6684X *       USES    A,F,H,L
                        6685X
                        6686X
101.142  053            6687X $SOB    DCX     H               PRE-DECREMENT
101.143  043            6688X $SOB1   INX     H
101.144  176            6689X         MOV     A,M
101.145  376 040        6690X         CPI     ' '
101.147  312 143 101    6691X         JE      $SOB1           GOT BLANK
101.152  376 011        6692X         CPI     TAB
101.154  312 143 101    6693X         JE      $SOB1           GOT TAB
101.157  311            6694X         RET
101.160                 6695          XTEXT   DRS


                        6697X **      $DRS - DECODE AND REMOVE SWITCHES.
                        6698X *
                        6699X *       $DRS IS CALLED TO DECODE COMMAND SWITCHES FROM A LINE
                        6700X *       OF TEXT. SWITCHES TAKE THE FORM:
                        6701X *
                        6702X *       /XXXXX
                        6703X *
                        6704X *       AFTER A SWITCH HAS BEEN LOCATED, IT (AND THE PRECEDING '/')
                        6705X *       ARE REPLACED WITH BLANKS.
                        6706X *
                        6707X *       VALID SWITCH DESCRIPTIONS ARE ENCODED INTO A TABLE
                        6708X *       SUPPLIED BY THE CALLER, IN THE FORMAT:
                        6709X *
                        6710X *       DB      'X...X'         REQUIRED SWITCH CHARACTERS
                        6711X *       DB      'C'+200Q,...,'C'+200Q   OPTIONAL CHARACTERS
                        6712X *       DB      200Q            END OF CHARACTERS
                        6713X *       DW      ADDR            PROCESSOR ADDRESS (CALLED WHEN SWITCH DETECTED)
                        6714X *
                        6715X *       DB      'Y...Y'         NEXT SWITCH
                        6716X *       .        .
                        6717X *       .        .
                        6718X *       .        .
                        6719X *
                        6720X *       DB      0               FLAGS END OF TABLE
                        6721X *
                        6722X *       SWITCHES MUST BE FOLLOWED BY A ':', A '/' (ANOTHER SWITCH)
                        6723X *       A ',', OR A 00 BYTE.
                        6724X *
```

```
                                6725X *       UPON DETECTION OF A VALID SWITCH, $DRS CALLS THE USER PROCESS
                                6726X *       ROUTINE, UPON ENTRY,
                                6727X *       (HL) = ADDRESS OF THE FIRST BYTE FOLLOWING THE SWTICH
                                6728X *       'Z' CLEAR IF CHARACTER = '/', ',', OR 00
                                6729X *       'Z' SET IF CHARACTER = ';'
                                6730X *
                                6731X *       THE USER ROUTINE CAN DECODE SWITCH SUB-OPTIONS, IF DESIRED.
                                6732X *       THE USER ROUTINE MAY USE ALL REGISTERS.
                                6733X *
                                6734X *       ENTRY   (DE) = SWITCH TABLE FWA
                                6735X *               (HL) = LINE FWA
                                6736X *       EXIT    'C' CLEAR IF OK
                                6737X *               'C' SET IF ERROR
                                6738X *                (HL) = ADDRESS OF START OF BAD SWITCH
                                6739X *                (A) = ERROR CODE
                                6740X *       USES    ALL
                                6741X
                                6742X
101.160                         6743X $DRS    EQU     *
                                6744X
                                6745X *       LOOK FOR SWITCHES
                                6746X
101.160  176                    6747X $DRS1   MOV     A,M
101.161  247                    6748X         ANA     A
101.162  310                    6749X         RZ                      END OF LINE
101.163  043                    6750X         INX     H
101.164  376 057                6751X         CPI     '/'
101.166  302 160 101            6752X         JNE     $DRS1           NOT A SWITCH
101.171  042 355 101            6753X         SHLD    $DRSB           ($DRSB) = SWITCH FWA (AFTER '/')
                                6754X
                                6755X *       GOT A SWITCH. LOOK FOR A MATCH IN THE CALLER'S TABLE
                                6756X
101.174  325                    6757X         PUSH    D               SAVE TABLE FWA
101.175  052 355 101            6758X $DRS2   LHLD    $DRSB           (HL) = SWITCH FWA
101.200  032                    6759X $DRS3   LDAX    D               (A) = TABLE ENTRY
101.201  346 177                6760X         ANI     177Q
101.203  312 253 101            6761X         JZ      $DRS6           GOT A MATCH
101.206  276                    6762X         CMP     M
101.207  302 217 101            6763X         JNE     $DRS4           NO MATCH
101.212  023                    6764X         INX     D
101.213  043                    6765X         INX     H
101.214  303 200 101            6766X         JMP     $DRS3           SEE IF MORE MATCH
                                6767X
                                6768X *       HAVE MIS-MATCH. SEE IF THE MISSING CHARACTER IS SIGNIFICANT
                                6769X
101.217  176                    6770X $DRS4   MOV     A,M             (A) = LINE CHARACTER WE COULDNT MATCH
101.220  315 324 101            6771X         CALL    $DRS15          SEE IF OK TERMINATOR
101.223  302 233 101            6772X         JNE     $DRS4.5         NO MATCH ON THIS SWITCH
101.226  032                    6773X         LDAX    D               (A) = NEXT CHARACTER IN SWITCH PATTERN
101.227  247                    6774X         ANA     A
101.230  372 253 101            6775X         JM      $DRS6           HAVE SUFFICIENT MATCH
101.233  315 337 101            6776X $DRS4.5 CALL    $DRS20          SKIP TABLE ENTRY
101.236  032                    6777X         LDAX    D
101.237  247                    6778X         ANA     A
101.240  302 175 101            6779X         JNZ     $DRS2           MORE SWITCHES IN TABLE TO CHECK
                                6780X
```

```
                                6781X *       BAD SWITCH
                                6782X
101.243  321                    6783X $DRS5   POP     D               RESTORE STACK
101.244  052 355 101            6784X         LHLD    $DRSB           POINT TO BAD SWITCH
101.247  067                    6785X         STC
101.250  076 032                6786X         MVI     A,EC.IS         ILLEGAL SWITCH
101.252  311                    6787X         RET
                                6788X
                                6789X *       HAVE SWITCH, CHECK IT'S FOLLOWING CHARACTER
                                6790X
101.253  315 142 101            6791X $DRS6   CALL    $SOB            SKIP OVER BLANKS
101.256  176                    6792X         MOV     A,M
101.257  315 324 101            6793X         CALL    $DRS15          CHECK CHARACTER
101.262  302 243 101            6794X         JNE     $DRS5           IN ERROR
101.265  315 337 101            6795X         CALL    $DRS20          GET PROCESSOR ADDRESS
101.270  021 302 101            6796X         LXI     D,$DRS7
101.273  345                    6797X         PUSH    H               SAVE (HL)
101.274  325                    6798X         PUSH    D               SET RETURN ADDRESS FOR TABLE CODE
101.275  305                    6799X         PUSH    B               SAVE PROCESSOR ADDRESS
101.276  176                    6800X         MOV     A,M             (A) = NEXT CHARACTER
101.277  376 072                6801X         CPI     ':'             SET CONDITION CODES
101.301  311                    6802X         RET                     CALL USER PROCESS
                                6803X
                                6804X *       USER PROCESS RETURNS HERE
                                6805X
101.302  321                    6806X $DRS7   POP     D               (DE) = LAST CHARACTER OF SWITCH+1
101.303  052 355 101            6807X         LHLD    $DRSB           (HL) = FIRST CHARACTER OF SWITCH AFTER /
101.306  053                    6808X         DCX     H               (HL) = ADDRESS OF '/'
                                6809X
                                6810X *       REPLACE SWITCH WITH BLANKS
                                6811X
101.307  066 040                6812X $DRS8   MVI     M,' '
101.311  043                    6813X         INX     H
101.312  315 216 030            6814X         CALL    $CDEHL
101.315  302 307 101            6815X         JNE     $DRS8           NOT THERE YET
101.320  321                    6816X         POP     D               (DE) = SWITCH TABLE FWA
101.321  303 160 101            6817X         JMP     $DRS1           LOOK FOR MORE SWITCHES

                                6819X **      $DRS15 - CHECK FOR VALID DELIMITER CHARACTER.
                                6820X *
                                6821X *       $DRS15 CHECKS THE NEXT TEXT CHARACTER TO SEE IF IT IS
                                6822X *
                                6823X *       00, '/', ',', ':'
                                6824X *
                                6825X *       ENTRY   (A) = CHARACTER
                                6826X *       EXIT    'Z' SET IFF CHARACTER IS ONE OF THE ABOVE
                                6827X *       USES    F
                                6828X
101.324  247                    6829X $DRS15  ANA     A
101.325  310                    6830X         RZ                      IS 00
101.326  376 057                6831X         CPI     '/'
101.330  310                    6832X         RE
101.331  376 054                6833X         CPI     ','
101.333  310                    6834X         RE
101.334  376 072                6835X         CPI     ':'
```

```
101.336  311          6836X          RET

                      6838X **       $DRS20 - GET PROCESSOR ADDRESS.
                      6839X *
                      6840X *        $DRS20 IS CALLED TO GET THE PROCESSOR ADDRESS FIELD OUT OF
                      6841X *        AN ENTRY IN THE SWITCH TABLE. THE CALLER SUPPLIES A POINTER
                      6842X *        TO SOMEWHERE IN THE TEXT PART OF THE SWITCH DESCRIPTION;
                      6843X *        $DRS20 ADVANCES THE POINTER TO THE PROCESSOR ADDRESS.
                      6844X *
                      6845X *        ENTRY   (DE) = POINTER TO TEXT PART OF SWITCH ENTRY
                      6846X *        EXIT    (DE) = POINTER TO 1ST BYTE OF NEXT SWITCH TABLE ENTRY
                      6847X *                (BC) = PROCESSOR ADDRESS FROM TABLE
                      6848X *        USES    A,F,B,C,D,E
                      6849X
                      6850X
101.337  032          6851X $DRS20   LDAX    D
101.340  023          6852X          INX     D
101.341  376 200      6853X          CPI     200Q
101.343  302 337 101  6854X          JNE     $DRS20
101.346  032          6855X          LDAX    D               (A) = LOW BYTE OF PROCESSOR ADDRESS
101.347  117          6856X          MOV     C,A
101.350  023          6857X          INX     D
101.351  032          6858X          LDAX    D
101.352  107          6859X          MOV     B,A             (BC) = PROCESSOR ADDRESS
101.353  023          6860X          INX     D
101.354  311          6861X          RET
                      6862X
101.355  000 000      6863X $DRSB    DW      0               POINTER TO SWITCH BEING PROCESSED
                      6864
101.357               6865  DEFALTB  DS      6               DEFAULTS FOR BINARY FILE NAME
101.365               6866  DEFALTL  DS      6               DEFAULTS FOR LISTINF FILE NAME
101.373               6867  DEFALTI  DS      6               DEFAULTS FOR INPUT FILE NAME
102.001               6868  DEFALTT  DS      6               DEFAULTS FOR TEMP FILE NAME     /80.03.GC/
                      6869
102.007               6870  MEML     EQU     *               LWA LOADED MEMORY
                      6871
```

```
                        6874
        073.154         6875            ORG     PRS             THESE BUFFERS OVERLAY PRS
                        6876
                        6877    **      STATEMENT UNPACK FIELDS. SETUP BY *UNL*.
                        6878
        073.154         6879            DS      1               SMASHED IF NO LABEL
        073.155         6880    LABEL   DS      8               LABEL FIELD
        073.165         6881            DS      1               SMASHED IF NO OPCODE
        073.166         6882    OPCODE  DS      5               OPCODE VALUE
        073.173         6883    EXPWRK  DS      99              EXPRESSION WORK-AREA            /80.02.GC/
        000.000         6884            ERRNZ   *-EXPWRK+1-LINEMAX                              /80.02.GC/
        377.374         6885            ERRPL   *-PRS2                                          /WCZ062780/
        073.173         6886    XTEXTB  EQU     EXPWRK          XTEXTB SCRATCH BUFFER
                        6887
        073.336         6888    BINBFR  DS      256             BINARY BUFFER
        002.000         6889    LISTBFL EQU     512             LISTING BUFFER SIZE
        001.000         6890    SORCBFL EQU     256             SOURCE BUFFER SIZE
        001.000         6891    XTXBFL  EQU     256             XTEXT BUFFER SIZE
        001.000         6892    TMPBFL  EQU     256             TEMP FILE BUFFER SIZE           /80.03.GC/
                        6893
        074.336         6894    LISTBUF DS      LISTBFL
        076.336         6895    SORCBUF DS      SORCBFL
        077.336         6896    XTXBUF  DS      XTXBFL
        100.336         6897    TMPBUF  DS      TMPBFL                                          /80.03.GC/
                        6898
                        6899
                        6900    *       BUFFERS USED BY PRESET
                        6901
        000.000         6902            IF      MEML-*&8000H                                    /WCZ062780/
        101.336         6903            DS      MEML-*+1                                        /WCZ062780/
                        6904            ENDIF                                                   /WCZ062780/
        377.377         6905            ERRPL   MEML-*          MUST NOT OVERLAY PRESET CODE
        102.010         6906    LINE    DS      100             LINE BUFFER
        000.144         6907    LINEMAX EQU     *-LINE          MAX LENGTH
                        6908
        102.154         6909    RMEML   EQU     *               MEM LIMIT WHEN RUNNING *PRS*
                        6910
        102.154         6911    SYMTAB  EQU     *               START OF SYMBOL TABLE
                        6912
                        6913
        102.154         6914            END
ASSEMBLY COMPLETE
  6914 STATEMENTS
     0 ERRORS DETECTED
  8160 BYTES FREE
```

```
$CCO      064026    656      3142     4295L
$CDEHL    030216   2859      3635     4268    4380E   6814
$CHL      030224    621       737     1531    1703    2836    4392E
$CPF      064207   1751      4581L    5983    5989    5995    6023
$CPF1     064211   4582L     4595     6235
$CPF2     064246   4584      4587     4589    4591    4601L
$CRLF     064251    756      3179     4655L
$CVD      100344   6531L
$CVD.     100345   6532L     6579     6620
$DADA     030072   4670E
$DADA.    030101    897      2411     2739    4402E   6617
$DNS      101105   6654L
$DNS.     101103   6380      6392     6653L
$DNS1     101123   6685L     6670     6673
$DNV      100354   6572L     6662
$DNV1     100361   6578L     6581
$DNV2     101024   6594      6597     6602L
$DNV3     101027   6606L     6621
$DNV4     101067   6610      6616     6623L
$DNVA     101071   6572      6601     6606    6626L
$DNVB     101072   6592      6627L
$DRS      101160   5955      6743E
$DRS1     101160   6747L     6752     6817
$DRS1S    101324   6771      6793     6829L
$DRS2     101175   6758L     6779
$DRS20    101337   6776      6795     6851L   6854
$DRS3     101200   6759L     6766
$DRS4     101217   6763      6770L
$DRS4.5   101233   6772      6776L
$DRS5     101243   6783L     6794
$DRS6     101253   6761      6775     6791L
$DRS7     101302   6796      6806L
$DRS8     101307   6812L     6815
$DRSB     101355   6753      6758     6784    6807    6863L
$DTB      063365   3186      3202     4255L
$DTB1     063371   4259L     4262
$DTB2     064000   4267L     4272
$DTB3     064015   4269      4276L
$DU66     030106   2677      4712E
$FCLEAR   065202   3935      5007E
$FCLO     065075    758       762     4059    4925L
$FCLO.    065104   4925      4929L
$FCLO1    065155   4968L     4971
$FCLO2    065165   4964      4973L
$FCLO3    065172   4937      4954    4984L
$FCLO4    065200   4935      4990L
$FERR1    067165   5529L     5534
$FERR2    067201   5532      5538L
$FERROR   067135   4241      4824    4828    4832    4927    5048    5177    5220    5338    5521L
$FFB      067040   5096      5453L
$FOPE1    065015   4854      4866L
$FOPE2    065060   4890      4894L
$FOPEA    065056   4894      4898E
$FOPEB    065066   4888      4908L
$FOPER    064335   4822L
$FOPER.   064362   1775      1806    1832    1838    4822    4835L   6037
$FOPEU    064353   4830L
$FOPEU.   064370   4830      4839L
```

```
$FOPEW  064344  4826L  6145
$FOPEW. 064365  4826   4837L  6131
$FREAL  065222  4054   4065   5045L
$FREAL. 065235  5045   5053E
$FWBRK  066254   653    682   5335L
$FWBRK. 066263  5335   5340L
$FWBRK1 066277  5342   5347L
$FWRIB  066072   679   1241   3195   3488   3542   3556   3593   3851   5189   5198   5218L
$FWRIB. 066101  5218   5223E
$FWRIL  066025  3197   3205   5175L
$FWRIL. 066034  5175   5181L
$FWRIL1 066040  5183L  5187
$FWRILA 066071  5197   5202L
$HLIHL  030211   898   4368E
$INDL   030234  4619E  4941   4944
$MCU    064043  4312L  4334
$MLU    064054  4329L  4454
$MLU1   064057  4332L  4337
$MOVE   030252  1803   4559   4643E  5926   6282
$MOVEL  064164  1708   1718   1753   1771   1828   3367   3842   4070   4543L  6045   6199
$MU66   030337  2327   2670   4724E
$MU86   031007  6551E  6608
$RCHAR  064153  4458   4493L  4494
$REAL2  065246  5062L  5139
$REAL3  065267  5074L  5098
$REAL4  065315  5087   5090L
$REAL6  065333  5092   5107L
$REAL7  065345  5117L  5135
$REAL8  066013  5067   5070   5097   5153L
$REAL9  066021  5159L
$REL7.3 065361  5122   5130L
$REL7.5 066004  5132   5143L
$RSTALL 031047  1099   2723   2782   2808   2810   4236   4299   4419E
$RTL    064115  4452   4456E
$RTL.   064106  4452L  5947
$RTL1   064116  4458L  4464
$RTL2   064150  4460   4481L
$SAVALL 031054  1096   2719   4211   4295   4433E
$SOB    101142  5973   5980   6687L  6791
$SOB1   101143  6688L  6691   6693
$TBLS   064304  1464   3259   4745L  4892   6501   6595
$TBRA   031076  1378   2169   2637   4784E
$TYPLZ  064074  3178   4352L  4357   6172
$TYPTX  031136   752    774   3981   4223   4799E  5522   5538   5932   5944   6054   6061   6067
                6073   6080   6168   6297   6304   6312   6386   6398   6409   6420   6431   6444   6458
$TYPTX. 031144   755   4801E
$UDD    031157  3576   3822   3830   3837   4045   4513E
$UOD    064257  3795   4019   4023   4683L
$WCHAR  064161  4356   4497L  5533
$WRIB2  066105  5229L  5309
$WRIB3  066123  5239L  5275
$WRIB4  066151  5252   5255L
$WRIB6  066213  5257   5288L
$WRIB7  066225  5298L  5305
$WRIB8  066247  5232   5235   5279   5315L
$ZEROS  031320   677   4806E
.CHFLG  000060   237L
.CLEAN  000205   252L
```

```
.CLEAR  000055      234L      673
.CLEARA 000056      235L     5937
.CLOSE  000046      227L      644    4989
.CLRCO  000007      211L
.CONSL  000006      210L     4298
.CTLC   000041      222L     5913
.DAD    000206      253L
.DECODE 000053      232L     1762    6006    6031    6255
.DELET  000050      229L
.DISMT  000061      238L
.DMNMS  000203      250L
.DMOUN  000201      248L
.ERROR  000057      236L     2850    5542    6087    6175
.EXIT   000000      204L      766    4891    4897    6089
.LINK   000040      221L      750
.LOADD  000062      239L
.LOADO  000010      212L
.MONMS  000202      249L
.MOUNT  000200      247L
.NAME   000054      233L     1822    5897
.OPEN   000063      240L
.OPENC  000045      226L
.OPENR  000042      223L     4908
.OPENU  000044      225L     4910
.OPENW  000043      224L     4909    6120
.POSIT  000047      228L     2758    3933    4216
.PRINT  000003      207L      659    3176
.READ   000004      208L     2769    5472
.RENAM  000051      230L
.RESET  000204      251L
.SCIN   000001      205L     4493
.SCOUT  000002      206L     4497    4656    6318
.SETTP  000052      231L     2847    5909
.SYSRES 000012      214L
.VERS   000011      213L     5887
.WRITE  000005      209L     2764    4235    4977    5273    5381
ABS.COD 000010      479L      596    1706    1709
ABS.ENT 000006      477L
ABS.ID  000000      473L
ABS.LDA 000002      475L
ABS.LEN 000004      476L
ABSENT  072245     1518     5765L
ABSFWA  072241      620      625    1530    1658    1685    1702    2728    2791    2800    5763L
ABSHDR  072237     1709     5761L
ABSLEN  072243     1705     5764L
ABSLWA  072247     1700     2802    2809    5767L
ABV     054232      632      637    2719L   3799
ABVO    054350     2759     2766L
ABVOO   054372     2770     2773L
ABV1    054373     2742     2777L
ABV2    055006     2726     2787E
ABV3    055030     2799     2801L
AIO.CGN 041047      437L
AIO.CHA 041116      452L
AIO.CNT 041111      448L
AIO.CSI 041050      438L
AIO.DDA 041041      433E
AIO.DES 041055      442L
```

```
AIO.DEV 041057      443L
AIO.DIR 041062      446L
AIO.DTA 041053      441L
AIO.EOF 041113      450L
AIO.EOM 041112      449L
AIO.FLG 041043      434L
AIO.GRT 041044      435L
AIO.LGN 041051      439L
AIO.LSI 041052      440L
AIO.SPG 041046      436L
AIO.TFP 041114      451L
AIO.UNI 041061      444L
AIO.VEC 041040      432L
ASM     043251      607       611     804L
ASM0.7  043350      827       829E
ASM1    043375      846L      1075
ASM10   044344     1016L      1154    1196    1951
ASM11   044367     1008       1038L   1165    1170    1499    1612    1768    1826    1846    1852
ASM11.  044374     1046L      1173    1318    1372    1579    1582    1599
ASM12   044376     1053L
ASM13   045001      910       917     1019    1022    1059L   1069    1331    1343    1419    1434    1545    1562
                   1570      1958
ASM14   045010      916       1067L   1204    1207    1214    1216    1233    1245    1257    1290    1876
ASM2    044053      861       868L
ASM3    044067      871       875L
ASM3.1  044116      888       890L
ASM4    044153      853       908L
ASM5    044162      874       914L
ASM6    044311      978L      1977    2065
ASM7    044321      866       992L    1988    2117
ASM8    044336      981       1007L   1968    2084    2104    2135
ASM8.   044335     1006L      1997    2009    2018    2029    2036    2080
ASMA    044175      896       919E    939
ASMB    044307      813       965L    1070
BDT     055052      604       2830L   2868
BDT1    055074     2833       2838L
BDT2    055132     2848       2854L
BDT3    055141     2857L      2860
BDT4    055154     2843       2866L
BELL    000007      100E      2849    3982    3982    4224    5523    6055    6062    6068    6074    6081    6169
                   6298      6306    6314
BINBFR  073336     1709       1719    2767    2777    4233    6888L
BINCSN  072235     2740       2749    4212    5758L
BINERR  063357     2765       2772    4219    4240L
BINFNAM 072214      639       2721    4240    5757L   5961    5982    6111    6121
BINSKW  072236     1707       1717    2738    5759L
BKSP    000010      102E
BOOT.P  000001      412E
C.STX   000002      104E
C.SYN   000026      103E
CBT     066360     5058       5225    5341    5393L
CBT1    066367     5398L      5407
CCHIT   043232      774L      5911
CDB.H84 000001      355E
CDB.H85 000000      354E
CEF     055164     1018       1126    1460    1879    2195    2882L
CLE     055174     1203       1215    2902L   3136
CLE1    055215     2908       2912L
```

```
CMA       055221      2062       2114       2131       2926L
CN.BIN    000000       77E        643       2757       2763       2768       4215       4234       6119
CN.LST    000001       78E       5791
CN.SOU    000002       79E       1821       3932       5800
CN.TMP    000004       81E        672       5818
CN.XTX    000003       80E       5809
CNDFLG    073046       815        872        908       1309       1325       1338       5833L
CO.FLG    000001      332E       4297
CODE      047200       957       1610E
CODE0     047206      1611       1614E
CODE1     047235      1624       1629L
CODE2     047260      1098       1632       1636       1645L
CODE2.2   047362      1672       1675       1679L
CODE2.5   050001      1656       1665       1678       1686L
CODE3     050052      1694       1716L
CODEFLG   072212       819       1093       1652       5750L
CODER     050070      1617       1619       1722E
CODER1    050105      1727       1731L
COL       055233      1236       2943L      3184
COL1      055256      2954       2960L
COL3      055263      2956       2964L
CR        000015       96E
CS.FLG    000200      333E
CSL.CHR   000001      309E
CSL.ECH   000200      306E
CSL.RAW   000004      307E
CSL.WRP   000002      308E
CT.ALPH   000200       72E       2401       4107
CTB       067006      5160       5317       5344       5419L
CTB1      067017      5425L      5434
CTLA      000001      111E
CTLB      000002      112E
CTLC      000003      113E       5912
CTLD      000004      114E       4459
CTLO      000017      115E
CTLP      000020      116E
CTLQ      000021      117E
CTLS      000023      118E
CTLZ      000032      119E
CTP.2SB   000010      318E
CTP.BKM   000002      319E
CTP.BKS   000200      314E
CTP.FF    000100      315E
CTP.MLI   000040      316E
CTP.MLO   000020      317E
CTP.TAB   000001      320E
CUS       055270      2983L      3823       3905       3978
D.CON     040110       268L
D.RAM     040240       271L
D.VEC     040130       270L
DB        045053       940       1107E
DB1       045056      1109L      1153
DB2       045066      1121L      1122
DB3       045114      1115       1133L
DB4       045127      1125       1127       1129       1141L
DB5       045132      1144L      1146
DB6       045143      1137       1150L
DEF       055333      1569       1590       3034L      3095
```

```
DEF0    055347  3036    3041L
DEF1    056005  3051    3053    3060L
DEFALTB 101357  6118    6200    6201    6865L
DEFALTI 101373  6029    6036    6202    6203    6867L
DEFALTL 101365  6130    6201    6202    6866L
DEFALTT 102001  6004    6144    6203    6868L
DEV.DDA 000004   161L
DEV.DVB 000015   174L
DEV.DVL 000013   173L
DEV.FLG 000006   162L
DEV.JMP 000003   160L
DEV.MNU 000010   170L
DEV.MUM 000007   169L
DEV.NAM 000000   152L
DEV.RES 000002   156L
DEV.UNT 000011   171L
DEVELEN 000016   176E
DF.CLR  000376   128E
DF.EMP  000377   127E
DHD     055310  2321    3006L
DHD1    055331  3009    3016L
DIR.ALD 000025   143L
DIR.CLU 000015   136L
DIR.CRD 000023   142L
DIR.EXT 000010   131L
DIR.FGN 000020   139L
DIR.FLG 000016   137L
DIR.LGN 000021   140L
DIR.LSI 000022   141L
DIR.NAM 000000   130L
DIR.PRO 000013   132L
DIR.VER 000014   133L
DIRELEN 000027   145E     446     586
DIRIDL  000015   134E
DLH     056013   878     1498    3078L
DLH1    056055  3087    3099L
DLL     056074  1059    3119L   3785
DLL0    056115  3127    3132L
DLL1    056134  3135    3142L
DLL2    056156  3150L   3156    3161
DLL2.5  056200  3154    3165L
DLL2.7  056216  3167    3175L
DLL3    056234  3138    3170    3183E
DLL4    056273  3188    3201L
DLLB    056310  3148    3207E
DLS     100260  6437    6450    6476L
DLS1    100270  6485L   6490    6507
DLS2    100336  6492    6494    6496    6511L
DNT     052074  2168E   2566    2582    2593
DNT1    052142  2185    2190L
DNT10   053016  2318L   2344
DNT11   053070  2320    2346L
DNT12   053112  2322    2343    2360L
DNT13   053113  2181    2189    2209    2364L
DNT14   053122  2352    2369L
DNT15   053137  2355    2374    2380L
DNT2    052145  2175    2195L
DNT3    052207  2205    2221L
```

```
DNT4     052216    2225L     2233
DNT5     052236    2227      2237L
DNT5.5   052275    2255      2259L
DNT5.7   052314    2264      2267E
DNT6     052323    2207      2277L
DNT7     052330    2279L     2287
DNT8     052350    2281      2291E
DNT9     052373    2298      2305L
DNTA     053144    2221      2242    2277    2316    2385L
DNTB     052372    2301E
DNTC     053073    2295      2349E
DNTD     053072    2293      2329    2348E
DR.IM    000001     157E
DR.PR    000002     158E
DRS      056331    1995      2004    2016    2027    2041    2055    2124    2132    3242L
DRS.     056351    3251      3257L
DRS1     056365    3268L     3273
DRS2     057001    3271      3274L
DRS3     057003    3262      3279L
DRSA     057011    1996      2005    2056    2125    2133    3286E
DRSB     057032    2017      2042    3297E
DRSC     057043    2028      3304E
DS       045155     941      1163E
DS1      045200    1168      1171L
DSPLEN   000040    3192      3875    5723E
DSPLIM   072150    3773      5719E
DSPLIN   072120    3147      3175    3193    3201    3874    5707L   5723
DSPLNA   072123    4017      5709L
DSPLNB   072134    3881      5713L
DSPLNC   072144    5717L
DSPLND   072150    3368      4043    5720L
DSPLNE   072155    3130      5721L
DT.CH    000020     167E
DT.CR    000002     164E
DT.CW    000004     165E
DT.DD    000001     163E      6008    6033    6259
DT.RN    000010     166E
DV.EL    000000     153E
DV.NU    000001     154E
DW       045207     942      1181E
DW1      045212    1186L     1195
E8B      057054    1134      1217    1223    1975    2063    2093    3324L
E8B1     057100    3327      3339L
EC.CNA   000004     515L
EC.DDA   000027     534L
EC.DIF   000017     526L
EC.DIW   000035     540L
EC.DNI   000045     548L
EC.DNR   000046     549L
EC.DNS   000005     516L
EC.DSC   000047     550L
EC.EOF   000001     512L     2771    4218    5047    5481
EC.EOM   000002     513L
EC.FAO   000031     536L     4861
EC.FAP   000026     533L
EC.FL    000030     535L
EC.FNF   000014     523L
EC.FNO   000011     520L     5065
```

```
EC.FNR  000034   539L
EC.FOD  000043   548L
EC.FUC  000013   522L
EC.ICN  000016   525L
EC.IDN  000006   517L
EC.IFC  000020   527L
EC.IFN  000007   518L
EC.ILC  000003   514L
EC.ILO  000040   543L
EC.ILR  000012   521L
EC.ILV  000037   542L
EC.IOI  000052   553L
EC.IS   000032   537L   6786
EC.NCV  000050   551L   6084
EC.NEM  000021   528L
EC.NOS  000051   552L
EC.NPM  000044   547L
EC.NRD  000010   519L
EC.NVM  000042   545L
EC.OTL  000053   554L
EC.RF   000022   529L
EC.UNA  000036   541L
EC.UND  000015   524L   1764
EC.UUN  000033   538L
EC.VPM  000041   544L
EC.WF   000023   530L
EC.WP   000025   532L
EC.WPV  000024   531L
EJECT   045242   943    1203L  1226   1235
EJEFLG  073047  1206    2947   2951   5835L
ELSE    046120   944    1325L
END     047001   945    1507L
END.    047074  1541E   3983
END1    047041  1514    1525L
END2    047071  1527    1536L
END3    047074  1521    1534   1543L
ENDFLG  072175   816    1072   1544   5737L
ENDIF   046136   946    1338L
ENL     000212   109E   3862   4227   5933   6081
EOFFLG  067134  5056    5453   5480   5507L
EPO     057105  1164    1307   1492   1515   1561   1668   3364L
EPO2    057154  3374    3392L
EPO3    057175  3388    3396   3399L
EPO4    057176  3381    3400L
EPOA    057204  3368    3386   3393   3405L
EQU     047104   947    1557L
EQU1    047135  1564    1574L
ERR.A   000010   31E    1021   1638   1729   1767   1825   1957   2365   2615   2931   3211   3398
                3689
ERR.D   000004   30E    1581   3057   3106   3210
ERR.F   000040   33E    3038   3213   4095   4110   4172
ERR.O   000100   34E     863   1031   3214
ERR.O.  044363  1030L    1491   1510   1625   1748
ERR.P   000200   35E    1375   2266   3215
ERR.R   000002   29E    2082   2688   3209   3280   3340
ERR.U   000001   28E    1845   1933   2257   3208   3379   3398
ERR.V   000020   32E    1172   1677   2098   2381   3212   3334
ERR1    046213  1373    1378L
```

```
ERRCNT  072165   810   3144  3146  3831  5731L
ERRFLG  072202  1061   1067  3132  3378  3401  3952  3934  5740L
ERRMI   046236  1382   1399L
ERRMI.  046171   963   1364L
ERRNZ   046226  1380   1389L
ERRNZ.  046157   961   1358L
ERRPL   046232  1381   1394L
ERRPL.  046164   962   1361L
ERRSHO  072167  3168   5732L 6428
ERRXX   046173   948   1356  1359  1362  1366  1368E
ERRZR   046222  1379   1384L
ERRZR.  046153   960   1355L
ESC     000033   107E
ESR     057213  1559   1594  3083  3427L
ESR.    057221  1924   1937  2244  3428  3431L
ESR0    057275  3453   3474L
ESR1    057300  3475L  3480
ESR4    057341  3460   3469  3496L
ESRA    057345  3474   3486  3501L 3504
ESRAL   000012  3485   3504E
ESRB    057354  3483   3502L
ESRC    057356  3431   3451  3503L
EVL     053355  1186   1370  1596  1986  2115  2551L 3324  3371
EVL1    054000  2559   2562L
EVL10   054105  2606   2636E
EVL11   054114  2638   2643L
EVL12   054133  2639   2654L
EVL13   054161  2640   2669L
EVL14   054170  2641   2673L
EVL15   054177  2671   2678L
EVL16   054211  2651   2666  2687L
EVL2    054014  2569   2574L
EVL20   054216  2574   2601  2699L
EVL3    054017  2571   2575L
EVL4    054027  2582L  2609
EVL5    054040  2567   2568  2583  2584  2585  2586  2592L
EVL6    054052  2596   2601L
EVL7    054055  2598   2602L
EVL7.5  054071  2594   2595  2597  2599  2613L
EVL8    054072  2570   2572  2587  2614L
EVL9    054076  2588   2619L
EVLA    054102  2562   2623E
EXIT    043211   605    763   765L  4228  5948
EXIT.   043212   766L   777   786
EXPREL  073052  2554   2578  2645  2662  2664  2679  3041  3092  3325  3897  5841L
EXPWRK  073173   903   4120  6005  6007  6030  6032  6254  6258  6262  6264  6280  6883L
                6884   6886
FB.CHA  000000   497L   4845  4848  4975  4986  5009  5493
FB.FLG  000001   498L    650   663  3165  4848  4865  4903  4930  4942  4945  5009  5011
                5495   5940  5941  5942
FB.FWA  000002   499L   4865  4871  4945  5011  5017  5497
FB.LIM  000006   501L   4876  4881  5022  5501
FB.LWA  000010   502L   5503
FB.NAM  000012   503L    504   824  1750  1761  3438  4240  4881  5524  5962  5963  5988
                5994   6003  6022  6028  6121  6127  6136  6141  6170  6234  6241  6252
FB.NAML 000021   504E   4581  5757  5797  5806  5815  5824  6233  6250
FB.RTR  000004   500L   4871  4876  4942  5017  5022  5499
FBENL   000033   505E
```

```
FF       000014    110E    3612
FLGERA   044354   1020L    1256   1289   1308   1312   1328   1342   1465   1493   1669
FNP      057357    2961    3520E  3847
FNP.     057364   3524E    3607
FNP0     060002    3527    3532E
FNP1     060030    3535    3547L
FNP2     060043   3552L    3559
FNP3     060065    3530    3543   3551   3563E
FNP4     060121   3580L    3586
FNP5     060136    3582    3588   3589E
FNP6     060163     649    3604E
FNPA     060201    3540    3612L
FNPB     060202    3553    3613L
FNPC     060203    3525    3529   3614L
FNPD     060204    3522    3564   3606   3615L
FORMDP   072261     707    3533   5778L  6194   6395
FT.ABS   000000    463E     597   1097   1490   1512   1631   1635   1654   5762
FT.BAC   000003    466E
FT.DD    000001    565E
FT.OC    000020    569E
FT.OR    000002    566E    4835   4839   4908   4910   5064
FT.OU    000010    568E
FT.OW    000004    567E    4837   4839   4909   4910   4936   5231
FT.PIC   000001    464E     613   1635   1647   1693   3628   5770
FT.REL   000002    465E
FTFLAG   072210     612     820   1488   1511   1646   1691   3627   5748L
FWBRK2   066316   5361L    5368
FWBRK3   066332    5363    5370L
GCP      045033     893    1093L
GRPFLG   072203     818     890   1620   5741L
GRT      060205    1536    3627L
GRT1     060221   3635L    3647
GRT2     060245    3636    3649L
GSC      060255    1121    1145   1268   2180   2184   2188   3667L
GSC3     060277    3678    3683L
GSC4     060301    3670    3688L
H8ASM1   042301     614     639L
H8ASM2   042317     641     648L
H8ASM3   042375     670     676L
H8ASM3M  043100     730     732L
H8ASM3N  043112     734     736L
H8ASM30  043130     739     742L
H8ASM4   043203     665     674    759    761L
HEADA    072113    3574    3578   5699L  5700
HEADAL   000003    3575    3579   5700E
HEADC    072072   5696L    5697   6048
HEADCL   000011   5697E    6046
HEADING  071277    3591    5686E  5702
HEADLEN  000221    3590    5702E
I.CONFL  000004    335E     336   4296
I.CONTY  000001    322E     323
I.CONWI  000003    328E     329
I.CSLMD  000000    311E
I.CUSOR  000002    325E     326
IF       046067     949    1307L
IF1      046115    1316    1318L
IMM      051226     924    1975L
INDX     051355     932    2075L
```

```
INDX1    051376    2077    2083L
INX      051307     930    2035L
INX1     051315    2035    2041L  2111
IOC.CGN  000010     574L
IOC.CSI  000011     575L
IOC.DDA  000002     562L     570   584
IOC.DES  000016     581L
IOC.DEV  000020     582L
IOC.DIL  000021     584E
IOC.DIR  000023     586L
IOC.DRL  000010     578E
IOC.DTA  000014     580L
IOC.FLG  000004     564L     578
IOC.GRT  000005     572L
IOC.LGN  000012     576L
IOC.LNK  000000     561L
IOC.LSI  000013     577L
IOC.SPG  000007     573L
IOC.SQL  000003     570E
IOC.UNI  000022     583L
IOCCTD   000001     590E
IOCELEN  000052     588E
LABEL    073155    1574    3034  3045  3078  3099  3427  4080  4091  4101  4137  6880L
LARGE    072213    2831    2867  5752L 6417
LCT      053167    1881    1892  2202  2225  2279  2403L
LCT.     053174    2296    2307  2406E 4106
LCT1     053213    2405    2408  2416L
LCTA     053215    2410    2419E
LEV      000000      23E
LF       000012      97E
LINCNT   073050    1224    1231  2951  3549  3599  3609  3845  3852  3854  5836L
LINE     102010    3177    3185  3196  3880  4048  4071  4073  4076  5894  5895  5899  5902
                   5925    5946  5954  5964  5972  5979  6906L 6907
LINEMAX  000144    4047    6884  6907E
LISTBFL  002000    5796    6889E 6894
LISTBUF  074336    5793    5794  5795  5796  6894L
LISTFB   072272     650     652   713   728   731   761  1238  3165  3194  3204  3541  3554
                   3592    3850  5790L 5940  5962  5988  6126  6127
LOF      046261     950    1433L 1443
LON      046242     951    1418L 1424
LST      046304    1418    1433  1459L
LST.C    000004      41E    1477  2913
LST.G    000200      43E    1474  3780
LST.I    000002      40E     915  1475
LST.L    000001      39E     822  1473  2904
LST.R    000010      42E     828  1476  3459
LSTA     046332    1463    1472E 6500
LSTCTL   072172     830     837   842   914  1467  2902  3458  3779  5734L
LSTCTLC  072174     840    1421  5736L 6198  6452
LSTCTLS  072173     836     840  1440  5735L 6196  6439
LVT      060313     860    3712E
LVT.     060311    3395    3710L 3971
LVT0     060316    3714L   3742
LVT1     060322    3721L   3727
LVT2     060337    3723    3734L 3737
LVTA     060347    3713    3740L
LXI      052027     934    2111L
MEML     102007     599    6870E 6902  6903  6905
```

```
MEMOVR  062061    3906    3981L
MI.INXH 000043     89E    3710
MI.JMP  000303     86E
MI.LDA  000072     87E
MI.PSHD 000325     90E
MI.RET  000311     88E
MOD     000101     24E
MOV     052046    935    2124L
MVI     051331    931    2055L
NL      000012   108E     109    753    768   1247   3613   3857   3857   3858   3859   3860   3982
                 3982    3982   3982   4224   4225   4463   4655   5131   5202   5523   5540   5691   5701
                 5701    5933   5945   6086   6174
NOREF   051051    959    1873E
NOREF1  051060   1878E   1946
NOREF4  051104   1891E   1900
NOREF5  051124   1894    1905E
NOREF6  051162   1918    1931E
NOREF7  051176   1926    1938   1942E
NOREF8  051205   1880    1950E
NOREF9  051210   1883    1955E
NOREFA  051217   1887    1922   1935   1960L
NUL2    000000     99E
NULL    000200     98E
NULTITL 046064    805     807   1297L
OBB     060354    979     993    996   1007   1136   1144   1189   1191   1520   3644   3646   3650
                 3652    3764L
OBB1    060376   3772L    3789
OBB2    061033   3775    3793L
OBB3    061044   3771    3782   3798L
OBBA    061034   3768    3794E  3798
OBBPTR  072162   3772    3797   3882   5728L
OF.CE   000200    869    5562E  5598   5599   5600
OF.LD   000100    876    5563E  5597   5598   5599   5600   5601   5602   5603   5604   5605   5607
                 5625    5626   5632   5657   5659   5664   5667   5672   5673   5674   5675
OOF     075177   6017    6108L
OOF1    075227   6114    6126L
OOF1.5  075255   6129    6135L
OOF2    075261   6133    6140L
OOF3    075304   6122    6132   6150L
OOF4    075310   6143    6146   6152L
OPCODE  073166    858    4114   4140   6882L
OPCTAB  067217    857    5566E
ORG     072176    618     628    635    832    846   1166   1169   1496   1528   1660   1681   2701
                 2730    2793   3089   3765   3767   3907   5738L
OVL.IN  000001    379E
OVL.NUM 000014    381E
OVL.RES 000002    380E
OVL.UCS 000200    382E
PAGEDP  072260    705    1229   3547   3597   3608   5777L  6192   6384
PAGNUM  073051    709     817   3568   3570   5837L
PAS     061054    648    3817L
PAS0    061104   3826    3828L
PAS2    061151   3841    3844E
PASA    061206    658    3849   3857L  3861
PASAL   000104   3848    3861E
PASB    061210   3820    3858L
PASC    061243   3828    3859L
PASD    061264   3834    3843   3860L
```

```
PASE    061313  3843    3863L
PASS    072164   804     899   1525  1686  1874  2724  3085  3119  3372  3432  3769  3900
                5730L
PATCH   073054  5845L
PDL     061320   848   3786  3874L
PDL1    061325  3876L  3879
PFE     075314  6039   6150  6167L
PIC.COD 000006   491L  1659  1674  1683  1719
PIC.ID  000000   486L
PIC.LEN 000002   488L   626
PIC.PTR 000004   489L
PICHDR  072251  1719  5769L
PICLEN  072253  5771L
PICPTR  072255  1533  5772L
PORG    046346   952  1488L
PRS     073154   600  5883E  6875
PRS1    073275  5924  5932L
PRS10   074367  5984  5990  5996  6061L
PRS11   075020  6016  6067L
PRS12   075046  6013  6073L
PRS2    073342  5937L  6018  6040  6057  6063  6069  6076  6082  6326  6885
PRS3    074002  5928  5952L
PRS4    074036  5966L  5971
PRS5    074063  5969  5979L
PRS6    074171  5986  5992  6015L
PRS7    074204  5975  6022L
PRS8    074264  6038  6044E
PRS9    074300  6034  6054L
PRSERR  075170  5910  6086L
PRSERR1 075166  5888  5890  6084L
PSIND   000015   886   939E
QUOTE   000047   105E  1114  1262  2174  3671  3677
RAO     051244   926  1995L
RAT     051254   927  2004L
RELFLG  072211   821  1650  1732  2699  3091  5749L
RELLWA  072266  2845  2856  3633  5785L
RELPTR  072270  2846  2986  3631  3909  3914  5786L
RESTART 043244   785L  5543
RMEML   102154  5908  6909E
ROMBOOT 030000   263E
RPO     051267   928  2016L
RPT     051277   929  2027L
RRI     061346   994  1187  3897L
RSF     062011   609  3931L
RST     052002   933  2093L
RST1    052017  2096  2099L
RUBOUT  000177   101E
S.BAUD  040344   356L
S.BDA   041120   454L
S.BOOTF 041034   411L
S.CAADR 040333   339L
S.CACC  041006   395L
S.CCTAB 040335   340L
S.CDB   040343   353L
S.CFWA  040352   363L
S.CODE  041007   396L
S.CONFL 040332   337L
S.CONTY 040327   324L
```

```
S.CONWI 040331  330L
S.CSLMD 040326  312L  323   326   329   336
S.CUSOR 040330  327L
S.DATC  040310  293L
S.DATE  040277  292L  6047
S.DCS   041033  409L
S.DDDTA 040366  374L
S.DDGRP 040364  371L
S.DDLDA 040360  369L
S.DDLEN 040362  370L
S.DDOPC 040370  375L
S.DFWA  040354  364L
S.DIREA 041016  403L
S.DLINK 040346  361L
S.FASER 041013  402L
S.FCI   041021  404L
S.GRT0  024000  259E
S.GRT1  025000  260E
S.GRT2  026000  261E
S.GUP   041027  406L
S.HIMEM 040316  295L
S.INT   040343  273L  349
S.JUMPS 041010  400L
S.MOUNT 041032  408L
S.OFWA  040350  362L
S.OMAX  040324  301L  2835
S.OSN   041004  391L
S.OVLE  041000  388L
S.OVLFL 040371  384L
S.OVLS  040376  387L
S.OVSTK 041035  416L
S.RFWA  040356  365L
S.SCI   041024  405L
S.SCR   041121  455L
S.SDD   041010  401L
S.SOVR  041146  275L  277
S.SSN   041002  390L
S.SYSM  040320  297L  2830
S.TIME  040312  294L
S.UCSF  040372  385L
S.UCSL  040374  386L
S.USRM  040322  299L
S.VAL   040277  272L  290
SDV     075354  5952  6191L
SDVA    076011  6200  6200  6206L
SDVA3   076033  5901  5904  6209L
SDVB    076041  6200  6210E
SEF     062026   862  1020  1030  1171  1374  1580  1637  1676  1728  1766  1824  1844
                1932  1956  2081  2097  2256  2265  2364  2380  2614  2687  2930  3037  3056
                3105  3279  3333  3339  3397  3688  3951L 4094  4109  4171
SET     047157   953  1592E
SNG     044336   923  1968E
SORCBFL 001000  5805  6890E 6895
SORCBUF 076336  5802  5803  5804  5805  6895L
SORCFB  072325  3934  4064  5799L 5941  6022  6028  6035
SORG    072200   847  1038  1497  1661  1682  4013  5739L
SPACE   045260   954  1214L
SPC1    045316  1221  1228L
```

```
SPC2    045336  1236L   1244
SPCA    045366  1239    1247L
SST     062041  1575    1913    2250    3046    3100    3387    3466    3970L
SST1    062114  3980    3985L
SST2    062116  3990L   3995
ST.DBL  000200   59E    1577    3054    3102
ST.DNA  000020   56E
ST.EQU  000002   51E    1588
ST.LAB  000001   50E    3090
ST.MSK  000007   53E
ST.NRF  000010   55E    1919    3468
ST.REL  000100   58E    1647    1733    2260    2648
ST.SET  000003   52E    1597    1597
ST.UND  000000   49E    1568    1917    2253    3050
STACK   042200  279E    5917    5920    5943
STACKL  001032  277E
START   042200  602E    6050
STATNO  072170  811     849     851     3482    3817    4040    5733L
STL     046042  955     1288L
STL.    046053  808     1288    1292L
STLTXT  072007  1293    5693L   5694
STXTL   000062  1292    5694E
SW.ERR  075140  5956    6080L   6388    6400    6411    6422    6433    6446    6460
SW.ERS  100042  6374    6426L
SW.ERS1 100053  6426    6431L
SW.FOR  077213  6349    6392L
SW.FOR1 077227  6393    6398L
SW.LAR  077355  6369    6415L
SW.LAR1 077366  6415    6420L
SW.LOF  100202  6364    6450L
SW.LOF1 100217  6451    6458L
SW.LON  100125  6359    6437L
SW.LON1 100141  6438    6444L
SW.PAG  077137  6344    6380L
SW.PAG1 077153  6381    6386L
SW.WID  077271  6354    6404L
SW.WID1 077302  6404    6409L
SWITAB  077057  5953    6339L
SYDD    040130  269E
SYMFWA  072262  717     736     740     2854    3394    3970    5783L
SYMPTR  072264  719     2984    3999    5784L
SYMTAB  102154  2841    5783    5784    6911E
SYSCALL 000377  197E     644     659     766    1762    1822    2758    2764    2769    2847    2850    3176
                3933    4216    4235    4298    4493    4497    4656    4897    4927    4989    5273    5381    5472
                5542    5887    5909    5913    5937    6006    6031    6087    6089    6120    6175    6255    6318
T.CHA   067122  5272    5380    5396    5424    5471    5494L   5495    5497    5499    5501    5503    5505
T.FLG   067123  5063    5230    5496L
T.FWA   067124  5262    5370    5460    5498L
T.LIM   067130  5076    5464    5484    5486    5502L
T.LWA   067132  5241    5265    5347    5373    5463    5504L
T.PTR   067126  5074    5137    5145    5239    5263    5307    5349    5371    5461    5500L
TAB     000011  106E    2884    4163    4190    5689    5695    6692
TBL1    064313  4749L   4755
TBL2    064331  4747    4759L
TBL3    064333  4752    4763L
TEMPFB  073013   663     678     681     715     732     735     757     824    3438    3487    5817L   5942
                5963    5994    6003    6136    6140    6141
THR     051235   925    1986L
```

```
TITLE   045367    956   1255L
TITLE.  046000    806   1255   1259L
TLEN    000012   5395   5505E
TMPBFL  001000   5823   6892E  6897
TMPBUF  100336   5820   5821   5822   5823   6897L
TOKREL  073053   2172   2261   2577   2646   2661   2680   2700   5842L
TTL1    046005   1261L  1294
TTL2    046014   1265L  1270
TTL3    046030   1274L  1277
TTL4    046040   1263   1266   1280L
TTLTXT  071277    711    727   1259   5687L  5688
TTXTL   000062   1260   5688E
UNL     062170    852   4038E  4062
UNL0    062273   4055   4066   4073L
UNL0.3  062322   4086L  4097
UNL0.5  062330   4085   4091L
UNL0.7  062346   4093   4101L
UNL0.9  062374   4104   4108   4114L
UNL00   062250   4052   4064L
UNL1    063007   4121L  4126
UNL10   063103   4187L
UNL2.5  063025   4137L
UNL3    063040   4082   4116   4156L  4167
UNL5    063072   4159   4162   4164   4176L
UNL9    063102   4186L  4189   4191
UNLA    063117   4071   4194L  4195
UNLAL   000027   4071   4195E
UNT.DIS 000006    186L
UNT.FLG 000000    182L
UNT.GRT 000002    184L
UNT.GTS 000004    185L
UNT.SIZ 000010    188E
UNT.SPG 000001    183L
UOD1    064263   4687L  4697
UOL     062141   1108   1182   1507   4013L
UOL.    062144   1053   4014L
USERFWA 042200    280E    596    598    599    831   5765
VER     000002     22E
VERS    000040    195E   5889
WBB     063146    642   2747   4211L
WBB1    063151   4212L
WBB2    063342   4217   4232L
WIDE    072257    703   5776L  6195   6406
XREF    043214    749    754    768L  5900   5903
XREFCNT 072204    667   3490   3492   5742L
XT.EQU  000002     66E   1558
XT.LAB  000001     65E   3082
XT.NRF  000004     68E   1923
XT.REF  000000     64E   1936   2243   3452
XT.SET  000003     67E   1593
XTEXT   050121    958   1746L
XTEXT0D 050201   1763   1770E
XTEXT1  050231   1765   1790E
XTEXT1D 050243   1797E  1814
XTEXT3  050306   1793   1818E
XTEXT3D 050332   1823   1828L
XTEXT7  050370   1777   1843E
XTEXT8  050377   1776   1809   1833   1839   1850L
```

XTEXTA	051007	1754	1772	1773	1800	1804	1819	1829	1830	1854L	
XTEXTB	073173	1759	1820	6886E							
XTEXTC	051015	1837	1855L								
XTEXTD	051020	1754	1829	1856L							
XTEXTE	051023	1791	1857L	5939	6268	6284	6286				
XTEXTF	000005	1858E	1860	6269	6315	6317					
XTEXTG	000003	1754	1772	1802	1810	1829	1859E	1860	6275	6281	
XTEXTH	051024	1795	1860L	6278							
XTEXTI	051043	1760	1772	1861L							
XTI	076041	6026	6223E								
XTI0	076044	6227E	6292								
XTI0.5	076112	6243	6246E								
XTI6	076226	6236	6245	6256	6296E						
XTI7	076262	6260	6303E								
XTI8	076350	6270	6311E								
XTI9	077041	6299	6307	6319	6323E						
XTIA	077046	6225	6228	6237	6328L						
XTIC	077047	6239	6290	6329L							
XTID	077051	6253	6330L								
XTXBFL	001000	5814	6891E	6896							
XTXBUF	077336	5811	5812	5813	5814	6896L					
XTXFB	072360	1750	1761	1774	1805	1831	4053	5808L	6234	6241	6252
XTXFLG	072206	1508	1746	1851	4049	4061	5743L				
XTXLINE	072207	2908	3125	4050	5744L						

5874 BYTES FREE