

!A

LLOAD README1.L,A\$4000

LLOAD README2.L,A\$4000

LLOAD README3.L,A\$4000

LLOAD INCL.L,A\$4000

LLOAD EOS.L,A\$4000

*** End of Pass 1

LLOAD README1.L,A\$4000

LLOAD README2.L,A\$4000

LLOAD README3.L,A\$4000

LLOAD INCL.L,A\$4000

LLOAD CATALOG.L,A\$4000

LLOAD SLOT1.L,A\$4000

LLOAD SLOT2.L,A\$4000

LLOAD SLOT3.L,A\$4000

LLOAD SLOT4.L,A\$4000

LLOAD SLOT5.L,A\$4000

LLOAD SLOT6.L,A\$4000

LLOAD SLOT7.L,A\$4000

LLOAD MENU.L,A\$4000

LLOAD CAT.L,A\$4000

LLOAD ASEOS.L,A\$4000

LLOAD BINEOS.L,A\$4000

LLOAD ZIP.L,A\$4000

LLOAD SUBS.L,A\$4000

LLOAD PAGE1.L,A\$4000

LLOAD TABLES.L,A\$4000

LLOAD BANKS.L,D2,A\$4000

LLOAD EOS.L,A\$4000

*** End of Pass 2

```
0800      1          ttl "EOS+ Source Code, EOS.L"
0800      2          src "EOS.L"
0800      3      ;
0800      4      ;
0800      5      ; EOS.L
0800      6      ;
0800      7      ;
0800      8      ; EOS+ Source Code for a 512 KB EPROM that resides in a
0800      9      ; Primary EPROM Reader Interface Card.
0800     10      ;
0800     11      ; In a power up state, DOS 4.5.06H is loaded into memory.
0800     12      ;
0800     13      ; 2023 September 19
0800     14      ;
0800     15      ;
0800     16      ; DOS 4.5, Build 06
0800     17      ;
0800     18      ; 2024 February 14
0800     19      ;
0800     20      ;
0800     21      ; Start of Source Code: 0x4000
0800     22      ; Start of Symbol List: 0x7800
0800     23      ;
0800     24      ;
0800     25      ; Copyright (c) 2024 February 14 by
0800     26      ; Walland Philip Vrbancic Jr
0800     27      ;
0800     28      ; 6223 East Peabody Street
0800     29      ; Long Beach, California 90808
0800     30      ; Unitied States of America
0800     31      ;
0800     32      ; All Rights Reserved
0800     33      ;
0800     34      ; This software is the confidential and
0800     35      ; proprietary intellectual property of
0800     36      ; Walland Philip Vrbancic Jr
0800     37      ;
0800     38      ;
0800     39          icl "README1.L"
```

LLOAD README1.L,A\$4000

```
0800      1          ttl "EOS+ Source Code, README1.L"
0800      2      ;
0800      3      ;
0800      4      ; README1.L
0800      5      ;
0800      6      ;
0800      7      ; Lisa is configured using the SETUP program in order to
0800      8      ; provide 64 pages for a source code module from 0x4000 to
0800      9      ; to 0x7800. The Symbol Table begins at 0x7800 and it ends
0800     10      ; at 0xB800. This configuration provides enough space from
0800     11      ; 0x0800 to 0x4000 in order to segment object files into
0800     12      ; SEGnn files. SEGnn files usually begin at 01 and there
0800     13      ; can be any number of SEGnn files that are sequentially
0800     14      ; numbered. The control-P command can be used in order to
0800     15      ; sequentially load any number of SEGnn files beginning at
0800     16      ; any SEGnn number.
0800     17      ;
0800     18      ; Whenever a Primary routine is activated such as
0800     19      ; LOADLISA80, EOS+ initializes the Y-register with the LSB
0800     20      ; address and the A-register with the MSB address of
0800     21      ; EPBINEOS and the X-register with the slot number of an
0800     22      ; EPROM card times sixteen in order for the Primary routine
0800     23      ; to utilize the resources of BINEOS. Therefore, any call
0800     24      ; to BINEOS may be initiated at any time by the Primary
0800     25      ; routine as long as the Primary routine configures the
0800     26      ; Y-register with the LSB address and the A-register with
0800     27      ; the MSB address of a valid EOS+ DCB.
0800     28      ;
0800     29      ; I have designed the EPROM Reader interface card to
0800     30      ; respond to its peripheral-card ROM address space for a
0800     31      ; single EPROM card or for either EPROM card when multiple
0800     32      ; EPROM cards are utilized. The peripheral-card ROM space
0800     33      ; allows an Applesoft BASIC program to easily load and to
0800     34      ; run files or catalog files in any EPROM range and on any
0800     35      ; EPROM card of a multiple card system. For example, if
0800     36      ; an EPROM card resides in slot 4, CALL 50276 or 0xC400
0800     37      ; with the required parameters will perform the requested
0800     38      ; action or return the requested information. However, the
0800     39      ; EPROM card must be configured to EPROM 0 and Bank 0 in
0800     40      ; order to utilize the Applesoft BASIC slot interface
0800     41      ; routines. The following shows the commands to issue to
0800     42      ; an EPROM card when it resides in any of the following
0800     43      ; slots:
0800     44      ;
0800     45      ; Slot 1 - POKE 49296,128 (0xC090), CALL 49408 (0xC100)
0800     46      ; Slot 2 - POKE 49312,128 (0xC0A0), CALL 49664 (0xC200)
0800     47      ; Slot 3 - POKE 49328,128 (0xC0B0), CALL 49920 (0xC300)
0800     48      ; Slot 4 - POKE 49344,128 (0xC0C0), CALL 50176 (0xC400)
0800     49      ; Slot 5 - POKE 49360,128 (0xC0D0), CALL 50432 (0xC500)
0800     50      ; Slot 6 - POKE 49376,128 (0xC0E0), CALL 50688 (0xC600)
0800     51      ; Slot 7 - POKE 49392,128 (0xC0F0), CALL 50944 (0xC700)
0800     52      ;
0800     53      ; The hardware design of an EPROM card provides the ability
0800     54      ; to programmatically determine in which slot the EPROM
0800     55      ; card resides. Assembly language programs and Primary
0800     56      ; routines are provided the address of EPBINEOS, thus these
0800     57      ; programs do not require the slot number of the EPROM card
0800     58      ; that provided that address. An Applesoft program does
0800     59      ; require the slot number of the target EPROM card.
0800     60      ;
```

```
0800      61 ; The Applesoft CALL interface is defined as follows. All
0800      62 ; arrays must be dimensioned large enough to hold all of
0800      63 ; the anticipated data. Only Integer variables or arrays
0800      64 ; can be used for EOS+ numerical data. If a required
0800      65 ; parameter is missing, the CALL will return with no data.
0800      66 ; An Applesoft program must be in the RUN state for the
0800      67 ; CALL to complete successfully.
0800      68 ;
0800      69 ; Applesoft treats Integer variables as signed integer
0800      70 ; numbers with the numerical range of -32768:32767, or
0800      71 ; 0x8000:0x7FFF.
0800      72 ;
0800      73 ;
0800      74 ; Load File:
0800      75 ;
0800      76 ;   EP  = Slot number of an EPROM card
0800      77 ;
0800      78 ;   OFF = 128 (EPROM 0, Bank 0, On/Off control set to Off)
0800      79 ;
0800      80 ;   DEV = Device address of an EPROM card
0800      81 ;           EP * 16 + 49280  (EP * 0x10 + 0xC080)
0800      82 ;
0800      83 ;   EOS = Slot address of an EPROM card
0800      84 ;           EP * 256 + 49152  (EP * 0x100 + 0xC000)
0800      85 ;
0800      86 ;   C%  = 1
0800      87 ;
0800      88 ;   S%  = -1
0800      89 ;
0800      90 ;   POKE DEV, OFF
0800      91 ;
0800      92 ;   CALL EOS, C%, S%, E%, F$ [, A%]
0800      93 ;
0800      94 ;       C% = Command
0800      95 ;
0800      96 ;       S% = Status
0800      97 ;
0800      98 ;       E% = EPROM search range
0800      99 ;
0800     100 ;       F$ = File name
0800     101 ;
0800     102 ;       A% = Alternate load address (optional)
0800     103 ;
0800     104 ;
0800     105 ; Run File:
0800     106 ;
0800     107 ;   EP  = Slot number of an EPROM card
0800     108 ;
0800     109 ;   OFF = 128 (EPROM 0, Bank 0, On/Off control set to Off)
0800     110 ;
0800     111 ;   DEV = Device address of an EPROM card
0800     112 ;           EP * 16 + 49280  (EP * 0x10 + 0xC080)
0800     113 ;
0800     114 ;   EOS = Slot address of an EPROM card
0800     115 ;           EP * 256 + 49152  (EP * 0x100 + 0xC000)
0800     116 ;
0800     117 ;   C%  = 2
0800     118 ;
0800     119 ;   S%  = -1
0800     120 ;
0800     121 ;   POKE DEV, OFF
```

```
0800      122 ;
0800      123 ;   CALL EOS, C%, S%, E%, F$ [, A%]
0800      124 ;
0800      125 ;       C% = Command
0800      126 ;
0800      127 ;       S% = Status
0800      128 ;
0800      129 ;       E% = EPROM search range
0800      130 ;
0800      131 ;       F$ = File name
0800      132 ;
0800      133 ;       A% = Alternate run address (optional)
0800      134 ;
0800      135 ;
0800      136 ; Catalog:
0800      137 ;
0800      138 ;   EP  = Slot number of an EPROM card
0800      139 ;
0800      140 ;   OFF = 128 (EPROM 0, Bank 0, On/Off control set to Off)
0800      141 ;
0800      142 ;   M%  = Maximum number of anticipated data entries
0800      143 ;
0800      144 ;   DEV = Device address of an EPROM card
0800      145 ;           EP * 16 + 49280  (EP * 0x10 + 0xC080)
0800      146 ;
0800      147 ;   EOS = Slot address of an EPROM card
0800      148 ;           EP * 256 + 49152 (EP * 0x100 + 0xC000)
0800      149 ;
0800      150 ;   C%  = 3
0800      151 ;
0800      152 ;   S%  = -1
0800      153 ;
0800      154 ;   N%  = 0 (may be any other starting number)
0800      155 ;
0800      156 ;   DIM F$(M%), P%(4,M%)
0800      157 ;
0800      158 ;   POKE DEV, OFF
0800      159 ;
0800      160 ;   CALL EOS, C%, S%, E%, N%, F$(N%) [, P%(0,N%)]
0800      161 ;
0800      162 ;       C% = Command
0800      163 ;
0800      164 ;       S% = Status
0800      165 ;
0800      166 ;       E% = EPROM search range
0800      167 ;
0800      168 ;       N% = Number of entries found (returned)
0800      169 ;
0800      170 ;       F$ = File name array (returned)
0800      171 ;
0800      172 ;       P% = Parameter array (optional, returned)
0800      173 ;
0800      174 ;
0800      175 ; Variable names are at the users discretion, but their
0800      176 ; variable types are not. Only Integer variables can be
0800      177 ; used for all numerical data.
0800      178 ;
0800      179 ;
0800      180 ; Returned status values:
0800      181 ;
0800      182 ;   S% = 0x00 (0)    no error
```

```
0800      183 ; S% = 0xFF (-1) number of parameters exceeded
0800      184 ; S% = 0x01 (1) unknown command
0800      185 ; S% = 0x02 (2) number of parameters invalid
0800      186 ; S% = 0x03 (3) search range invalid
0800      187 ; S% = 0x04 (4) file not found
0800      188 ;
0800      189 ;
0800      190 ; Search range:
0800      191 ;
0800      192 ; E% = (last EPROM number) * 16 + (start EPROM number)
0800      193 ;
0800      194 ; E% = 0-15 for a single EPROM
0800      195 ; E% = 0-15:0-15 for a range of EPROMs
0800      196 ; E% = 0xF0 for all EPROMs
0800      197 ;
0800      198 ;
0800      199 ; File name array returned:
0800      200 ;
0800      201 ; F$(N%) = file names are 1 to 24 ASCII characters
0800      202 ;
0800      203 ;
0800      204 ; Optional Parameter array returned:
0800      205 ;
0800      206 ; P%(0,N%) = Slot/EPROM number
0800      207 ; P%(1,N%) = file type
0800      208 ; P%(2,N%) = EPROM offset
0800      209 ; P%(3,N%) = file size in bytes
0800      210 ; P%(4,N%) = destination memory address
0800      211 ;
0800      212 ;
0800      213 ; File types returned:
0800      214 ;
0800      215 ; 0x00 end of Catalog
0800      216 ; 0x01 for a NULL terminated Text file
0800      217 ; 0x02 for an Applesoft file
0800      218 ; 0x04 for a Binary file, main memory
0800      219 ; 0x08 for a Binary file, Language Card Bank 1
0800      220 ; 0x10 for a Binary file, Language Card Bank 2
0800      221 ; 0x20 for a Reserved file
0800      222 ; 0x40 for a System file
0800      223 ; 0x80 for a Primary file
0800      224 ;
0800      225 ;
0800      226      icl "README2.L"
```

LLOAD README2.L,A\$4000

```
0800      1          ttl "EOS+ Source Code, README2.L"
0800      2      ;
0800      3      ;
0800      4      ; README2.L
0800      5      ;
0800      6      ;
0800      7      ; The assembly language interface to EOS+ is defined below.
0800      8      ; A Data Context Block or DCB is used for the input
0800      9      ; variables and returned status when a call is made to the
0800     10      ; EOS+ entry address EPBINEOS. The structure of the DCB is
0800     11      ; generic and consists of eight bytes.
0800     12      ;
0800     13      ; Primary routines can utilize EPBINEOS in order to load
0800     14      ; and run System programs only after the slot number of an
0800     15      ; EPROM card has been determined. Whenever a Primary
0800     16      ; routine is activated, EOS+ initializes the Y-register
0800     17      ; with the LSB address and the A-register with the MSB
0800     18      ; address of EPBINEOS and the X-register with the slot
0800     19      ; number of an EPROM card times sixteen in order for the
0800     20      ; Primary routine to utilize the resources of BINEOS. The
0800     21      ; address for EPBINEOS can be utilized in order to load or
0800     22      ; to run any number of other routines, programs, and files
0800     23      ; that are found in an EPROM that is located on any EPROM
0800     24      ; card.
0800     25      ;
0800     26      ; The alternate Load/Run address is used only if the MSB
0800     27      ; of the provided alternate address is nonzero.
0800     28      ;
0800     29      ; A Primary routine can use the address that EOS+ provides
0800     30      ; in order to call EPBINEOS at any time. However, other
0800     31      ; routines can also utilize the EPBINEOS resources after
0800     32      ; the routine determines the slot number of an EPROM card.
0800     33      ; Knowing that the entry point of EPBINEOS is located at
0800     34      ; the 0xE0 byte in the peripheral card ROM memory of the
0800     35      ; EPROM card, a call to EPBINEOS can easily be constructed.
0800     36      ; The following list shows the address for EPBINEOS if that
0800     37      ; slot contains an EPROM card:
0800     38      ;
0800     39      ; Slot 1 - 0xC1E0
0800     40      ; Slot 2 - 0xC2E0
0800     41      ; Slot 3 - 0xC3E0
0800     42      ; Slot 4 - 0xC4E0
0800     43      ; Slot 5 - 0xC5E0
0800     44      ; Slot 6 - 0xC6E0
0800     45      ; Slot 7 - 0xC7E0
0800     46      ;
0800     47      ;
0800     48      ; Load File DCB:
0800     49      ;
0800     50      ; EOSDCB equ * ; Load file DCB
0800     51      ; DCBCMD hex 01 ; Load command
0800     52      ; DCBEPN hex F0 ; search all EPROMs
0800     53      ; DCBFALT hex 0000 ; no alternate LOAD address
0800     54      ; DCBSTAT hex FF ; return status
0800     55      ; DCBFLEN byt FILEEND-FILNAM ; filename length
0800     56      ; DCBFADR adr FILNAM ; filename address
0800     57      ;
0800     58      ; FILNAM asc "RamDisk Config"
0800     59      ; FILEEND equ *
0800     60      ;
```

```
0800      61 ;
0800      62 ;          jsr FINDEP
0800      63 ;          bcs FINDERR
0800      64 ;
0800      65 ;          ldy #EOSDCB
0800      66 ;          lda /EOSDCB
0800      67 ;
0800      68 ;          jsr BINJMP
0800      69 ;
0800      70 ;
0800      71 ; Run File DCB:
0800      72 ;
0800      73 ; EOSDCB equ * ; Run file DCB
0800      74 ; DCBCMD hex 02 ; Run command
0800      75 ; DCBEPN hex F0 ; search all EPROMs
0800      76 ; DCBFALT hex 0000 ; no alternate Run address
0800      77 ; DCBSTAT hex FF ; return status
0800      78 ; DCBFLEN byt FILEEND-FILNAM ; filename length
0800      79 ; DCBFADR adr FILNAM ; filename address
0800      80 ;
0800      81 ; FILNAM asc "Volume Copy"
0800      82 ; FILEEND equ *
0800      83 ;
0800      84 ;
0800      85 ;          jsr FINDEP
0800      86 ;          bcs FINDERR
0800      87 ;
0800      88 ;          ldy #EOSDCB
0800      89 ;          lda /EOSDCB
0800      90 ;
0800      91 ;          jsr BINJMP
0800      92 ;
0800      93 ;
0800      94 ; Catalog DCB:
0800      95 ;
0800      96 ; EOSDCB equ * ; Catalog EPROMs DCB
0800      97 ; DCBCMD hex 03 ; Catalog command
0800      98 ; DCBEPN hex F0 ; Catalog all EPROMs
0800      99 ; DCBCALT hex 0000 ; not used
0800     100 ; DCBSTAT hex FF ; return status
0800     101 ; DCBCNUM hex 00 ; number of entries found
0800     102 ; DCBCADR adr CATBUFR ; Catalog buffer address
0800     103 ;
0800     104 ; CATBUFR dfs 32*N,ZERO ; N 32-byte entries
0800     105 ;
0800     106 ;
0800     107 ;          jsr FINDEP
0800     108 ;          bcs FINDERR
0800     109 ;
0800     110 ;          ldy #EOSDCB
0800     111 ;          lda /EOSDCB
0800     112 ;
0800     113 ;          jsr BINJMP
0800     114 ;
0800     115 ;
0800     116 ; Returned status values:
0800     117 ;
0800     118 ; DCBSTAT = 0x00 no error
0800     119 ;          = 0x01 unknown command
0800     120 ;          = 0x02 filename length invalid
0800     121 ;          = 0x03 search range invalid
```



```

0800      122 ;                = 0x04    buffer/filename address invalid
0800      123 ;                = 0x05    file not found
0800      124 ;
0800      125 ;
0800      126 ; Search range:
0800      127 ;
0800      128 ;     DCBEPN  = (last EPROM number)*16 + (start EPROM number)
0800      129 ;
0800      130 ;                = 0-15 for a single EPROM
0800      131 ;                = 0-15:0-15 for a range of EPROMs
0800      132 ;                = 0xF0 for all EPROMs
0800      133 ;
0800      134 ;
0800      135 ; Catalog buffer array returned (each entry is 32 bytes):
0800      136 ;
0800      137 ;   CATBUFR:
0800      138 ;
0800      139 ;   FILEPNUM = Slot/EPROM number
0800      140 ;   FILETYPE = file type
0800      141 ;   SRCVAL   = EPROM offset
0800      142 ;   LENVAL   = file size in bytes
0800      143 ;   DSTVAL   = destination memory address
0800      144 ;   FILENAME = filename, space padded to 24 bytes
0800      145 ;
0800      146 ;
0800      147 ; File types returned:
0800      148 ;
0800      149 ;   FILETYPE = 0x01 for a Text file (NULL terminated file)
0800      150 ;                = 0x02 for an Applesoft file
0800      151 ;                = 0x04 for a Binary file, main memory
0800      152 ;                = 0x08 for a Binary file, Language Card Bank 1
0800      153 ;                = 0x10 for a Binary file, Language Card Bank 2
0800      154 ;                = 0x20 for a Reserved file
0800      155 ;                = 0x40 for a System file
0800      156 ;                = 0x80 for a Primary file
0800      157 ;
0800      158 ;
0800      159 ; Example code to locate an EPROM card:
0800      160 ;
0800 20 00 09 161      jsr FINDEP          ; find EPROM card
0803 B0 07   162      bcs FINDERR
0805         163 ;
0805 A0 5E    164      ldy #EOSLDCB      ; Address of
0807 A9 09    165      lda /EOSLDCB      ; Load DCB
0809         166 ;
0809 20 59 09 167      jsr BINJMP          ; Load the file
080C         168 ;
080C         169 ;      :::
080C         170 ;
080C         171 FINDERR:
080C         172 ;      :::
080C         173 ;
080C         174 ;
080C         175      dfs $100-*&$FF,0
0900         176 ;
0900         177 ;
0900         178 ; Slot cards are tested from Slot 1 to Slot 7 in order to
0900         179 ; discover an EPROM card if there are multiple EPROM cards
0900         180 ; in an Apple computer. This routine finds the first EPROM
0900         181 ; card.
0900         182 ;

```

```

0900 A0 00      183 FINDEP    ldy #ZERO
0902           184 ;
0902 A9 C1      185          lda /PAGEC1          ; page 0xC1
0904           186 ;
0904 84 2A      187          sty SRCPTR          ; store address at
0906 85 2B      188          sta SRCPTR+1        ; source pointer
0908           189 ;
0908 A9 E1      190          lda /PAGEE1          ; page 0xE1
090A           191 ;
090A 84 2E      192          sty DSTPTR          ; store address at
090C 85 2F      193          sta DSTPTR+1        ; destination pointer
090E           194 ;
090E A2 01      195          ldx #1              ; begin with slot 1
0910           196 ;
0910 8E A5 02    197 ^1      stx SLOT
0913 8A         198          txa
0914           199 ;
0914 0A         200          asl                  ; multiply by 16
0915 0A         201          asl
0916 0A         202          asl
0917 0A         203          asl
0918           204 ;
0918 AA         205          tax                  ; use as index
0919           206 ;
0919           207 ;
0919           208 ; Turn the EPROM card ON.
0919           209 ;
0919 A9 00      210          lda #EPONVAL          ; get EPROM card ON value
091B 9D 80 C0    211          sta EPSELC,X
091E           212 ;
091E 2C FF CF    213          bit CLRROM          ; detach shared slot memory
0921           214 ;
0921 A0 F8      215          ldy #EPBINTXT        ; check EPBIN text
0923           216 ;
0923           217 ;
0923           218 ; Begin testing for an EPROM card.
0923           219 ;
0923 B9 44 25    220 ^2      lda EPTEXT-EPBINTXT&NEGONE,Y ; get EPBIN text
0926           221 ;
0926 D1 2A      222          cmp (SRCPTR),Y          ; compare slot page
0928 D0 09      223          bne >3
092A           224 ;
092A D1 2E      225          cmp (DSTPTR),Y        ; compare EPROM page
092C D0 05      226          bne >3
092E           227 ;
092E C8         228          iny
092F D0 F2      229          bne <2
0931           230 ;
0931 F0 11      231          beq >4              ; always taken
0933           232 ;
0933           233 ;
0933           234 ; Go to the next slot to test.
0933           235 ;
0933 2C FF CF    236 ^3      bit CLRROM          ; detach shared slot memory
0936           237 ;
0936 E6 2B      238          inc SRCPTR+1          ; next slot page
0938 E6 2F      239          inc DSTPTR+1        ; next EPROM page
093A           240 ;
093A AE A5 02    241          ldx SLOT              ; get slot number
093D E8         242          inx
093E           243 ;

```

```

093E E0 08      244      cpx #8          ; done testing?
0940 D0 CE      245      bne <1
0942            246      ;
0942 38          247      sec              ; no EPROM card
0943            248      ;
0943 60          249      rts              ; return to caller
0944            250      ;
0944            251      ;
0944            252      ; An EPROM card has been successfully found.
0944            253      ;
0944 A9 80        254      ^4      lda #EPOFFVAL      ; get EPROM card OFF value
0946 9D 80 C0    255      sta EPSELC,X
0949            256      ;
0949 2C FF CF     257      bit CLRROM          ; detach shared slot memory
094C            258      ;
094C AD A5 02    259      lda SLOT            ; get EPROM card slot number
094F 8D 91 02    260      sta EPSLOT          ; save as the card slot number
0952            261      ;
0952 A5 2B        262      lda SRCPTR+1        ; get slot memory address
0954 8D 5D 09    263      sta BINADR+1        ; save to vector
0957            264      ;
0957 18          265      clc              ; EPROM card found
0958            266      ;
0958 60          267      rts              ; return to caller
0959            268      ;
0959            269      ;
0959 6C 5C 09     270      BINJMP      jmp (BINADR)
095C            271      ;
095C E0 0C        272      BINADR      adr EPBINEOS      ; EPBINEOS vector
095E            273      ;
095E            274      ;
095E            275      EOSLDCB      equ *              ; Load File DCB
095E            276      ;
095E 01          277      BINCMD      hex 01            ; Load command
095F F0          278      BINEPN      hex F0            ; search all EPROMs
0960 00 00       279      BINFALT     hex 0000          ; no alternate Load address
0962 FF          280      BINSTAT     hex FF            ; return status
0963 0E          281      BINFLEN     byt FILEEND-FILNAM ; filename length
0964 66 09       282      BINFADR     adr FILNAM         ; filename address
0966            283      ;
0966 D2 E1 ED     284      FILNAM      asc "RamDisk Config"
0969 C4 E9 F3
096C EB A0 C3
096F EF EE E6
0972 E9 E7
0974            285      FILEEND      equ *
0974            286      ;
0974            287      ;
0974            288      icl "README3.L"

```

LLOAD README3.L,A\$4000

```
0974      1          ttl "EOS+ Source Code, README3.L"
0974      2      ;
0974      3      ;
0974      4      ; README3.L
0974      5      ;
0974      6      ;
0974      7      ; Primary files are Binary files that load or run System
0974      8      ; files. The EOS+ EPROM Catalog function cannot directly
0974      9      ; load or run a System file. System files may be Text,
0974     10      ; Applesoft, or other Binary files. System files may be
0974     11      ; attached to a Primary file, or they may be loaded or run
0974     12      ; by activating its associated Primary file either using
0974     13      ; the EOS+ Catalog function or the EOS+ assembly language
0974     14      ; interface.
0974     15      ;
0974     16      ; EOS+ is not designed to handle Integer type files because
0974     17      ; DOS 4.5 does not support Integer type files.
0974     18      ;
0974     19      ;
0974     20      ; An EPROM catalog may contain any number of entries, and
0974     21      ; each entry is variable in size. The catalog is prefaced
0974     22      ; with four sync bytes that are 0xC4, 0xB8, 0x90, and 0xED.
0974     23      ; The catalog is terminated with 0x00. An EPROM catalog
0974     24      ; entry is organized as follows:
0974     25      ;
0974     26      ; byte 0x00 - File Type
0974     27      ;         0x01 - EPROM source address, or offset
0974     28      ;         0x03 - File length or size in bytes
0974     29      ;         0x05 - Destination memory address
0974     30      ;         0x07 - Filename, up to 24 ASCII bytes, 'DCI' format
0974     31      ;
0974     32      ;
0974     33      ; 'DCI' format is a string of ASCII bytes whose MSB is off
0974     34      ; except for the last byte of the string whose MSB is ON.
0974     35      ;
0974     36      ;
0974     37      ; An EPROM card can accommodate EPROMs as large as a 27512
0974     38      ; which can accommodate 64 kilobytes. An EPROM is
0974     39      ; partitioned into banks that are eight kilobytes in size
0974     40      ; and the 27512 holds eight banks. Each bank is accessed
0974     41      ; from memory address 0xE000 to 0xFFFF. When the data
0974     42      ; pointer becomes zero, the next bank is selected in order
0974     43      ; to continue reading data sequentially from the EPROM.
0974     44      ;
0974     45      ; An EPROM card is controlled by writing a value to its
0974     46      ; control register in order to select one of eight EPROMs.
0974     47      ; The quikLoader uses one of its control register data bits
0974     48      ; and two of its sixteen I/O device memory locations in
0974     49      ; order to select one of eight memory banks. The EPROM
0974     50      ; Reader card uses three of its sixteen I/O device memory
0974     51      ; locations in order to select one of eight memory banks.
0974     52      ; The X-register is configured to contain the EPROM card's
0974     53      ; slot number times sixteen plus the desired bank number.
0974     54      ;
0974     55      ; For a quikLoader, the control register is defined as
0974     56      ; follows:
0974     57      ;
0974     58      ;   bit:  7   6   5   4   3   2   1   0
0974     59      ;   use:  x   x   x   0   U   C   B   A
0974     60      ;
```

```

0974      61 ; x = bit not used
0974      62 ; O = On/Off control, 0=ON, 1=OFF
0974      63 ; U = USR bit, 0=even bank, 1=odd bank
0974      64 ; ABC = EPROM number, 0-7
0974      65 ;
0974      66 ; For a quikLoader, bank number is selected as follows:
0974      67 ;
0974      68 ; Bank  A1  A0   U  Memory Address  EPROM Offset
0974      69 ; ----  --  --   --  -----
0974      70 ;      0    0    0    0    0xE000-0xFFFF  0x0000-0x1FFF
0974      71 ;      1    0    0    1    0xE000-0xFFFF  0x2000-0x3FFF
0974      72 ;      2    0    1    0    0xE000-0xFFFF  0x4000-0x5FFF
0974      73 ;      3    0    1    1    0xE000-0xFFFF  0x6000-0x7FFF
0974      74 ;      4    1    0    0    0xE000-0xFFFF  0x8000-0x9FFF
0974      75 ;      5    1    0    1    0xE000-0xFFFF  0xA000-0xBFFF
0974      76 ;      6    1    1    0    0xE000-0xFFFF  0xC000-0xDFFF
0974      77 ;      7    1    1    1    0xE000-0xFFFF  0xE000-0xFFFF
0974      78 ;
0974      79 ;
0974      80 ; For an EPROM Reader card, the control register is
0974      81 ; defined as follows:
0974      82 ;
0974      83 ; bit:  7   6   5   4   3   2   1   0
0974      84 ; use:  O   Z   Y   X   U   C   B   A
0974      85 ;
0974      86 ; O = On/Off control, 0=ON, 1=OFF
0974      87 ; U = USR bit, 0=even bank, 1=odd bank
0974      88 ; ABC = EPROM number, 0-7
0974      89 ; XYZ = Bank number, 0-7, from A0, A1, and A2
0974      90 ;
0974      91 ; For an EPROM Reader card, bank number is selected as
0974      92 ; follows:
0974      93 ;
0974      94 ; Bank  A2  A1  A0  Memory Address  EPROM Offset
0974      95 ; ----  --  --   --  -----
0974      96 ;      0    0    0    0    0xE000-0xFFFF  0x0000-0x1FFF
0974      97 ;      1    0    0    1    0xE000-0xFFFF  0x2000-0x3FFF
0974      98 ;      2    0    1    0    0xE000-0xFFFF  0x4000-0x5FFF
0974      99 ;      3    0    1    1    0xE000-0xFFFF  0x6000-0x7FFF
0974     100 ;      4    1    0    0    0xE000-0xFFFF  0x8000-0x9FFF
0974     101 ;      5    1    0    1    0xE000-0xFFFF  0xA000-0xBFFF
0974     102 ;      6    1    1    0    0xE000-0xFFFF  0xC000-0xDFFF
0974     103 ;      7    1    1    1    0xE000-0xFFFF  0xE000-0xFFFF
0974     104 ;
0974     105 ;
0974     106 ; When EPROM 0 and Bank 0 are selected for the EPROM card,
0974     107 ; the peripheral-card ROM memory at 0xCs00 to 0xCsFF will
0974     108 ; display the same data that is found at memory address
0974     109 ; 0xEs00 to 0xEsFF, where 's' is the slot number in which
0974     110 ; the EPROM card resides. The last eight bytes of the
0974     111 ; EPROM card's peripheral-card ROM memory and the EPROM
0974     112 ; bank memory will both contain the ASCII characters
0974     113 ; "EPBINEOS".
0974     114 ;
0974     115 ;
0974     116 ; The EOS+ EPROM catalog must be contained within one page
0974     117 ; of data so that it does not overflow into the Slot 1
0974     118 ; EOS+ or EPASEOS interface. The layout design of the EOS+
0974     119 ; EPROM is as follows:
0974     120 ;
0974     121 ; Bank  Offset  Memory  Size      Description

```

```

0974      122 ; -----
0974      123 ; 0 0x0000 0xE000 0x0004 Sync bytes
0974      124 ;      0x0004 0xE004 0x00FC Catalog
0974      125 ;      0x0100 0xE100 0x0100 Slot 1 EOS+ interface
0974      126 ;      0x0200 0xE200 0x0100 Slot 2 EOS+ interface
0974      127 ;      0x0300 0xE300 0x0100 Slot 3 EOS+ interface
0974      128 ;      0x0400 0xE400 0x0100 Slot 4 EOS+ interface
0974      129 ;      0x0500 0xE500 0x0100 Slot 5 EOS+ interface
0974      130 ;      0x0600 0xE600 0x0100 Slot 6 EOS+ interface
0974      131 ;      0x0700 0xE700 0x0100 Slot 7 EOS+ interface
0974      132 ;      0x0800 0xE800 0x17FA EOS+ software
0974      133 ;      0x1FFA 0xFFFF 0x0002 NMI vector, EOS+ address
0974      134 ;      0x1FFC 0xFFFF 0x0002 RESET vector, EOS+ address
0974      135 ;      0x1FFE 0xFFFF 0x0002 IRQ/BRK vector, EOS+ addr
0974      136 ; 1 0x2000 0xE000 0x2100 DOS4.5L
0974      137 ; 2 0x4100 0xE100 0x2A00 DOS4.5H
0974      138 ; 3 0x6B00 0xEB00 0x2800 LISA80.1
0974      139 ; 4 0x9300 0xF300 0x1000 LISA80.2
0974      140 ; 5 0xA300 0xE300 0x0640 LISA80.3
0974      141 ; 5 0xA940 0xE940 0x1914 SETUP80
0974      142 ; 6 0xC254 0xE254 0x01C0 LOADLISA80
0974      143 ; 6 0xC414 0xE414 0x1B04 RamDisk Install
0974      144 ; 6 0xDF18 0xFF18 0x1316 FID
0974      145 ; 7 0xF22E 0xF22E 0x064A SetClock
0974      146 ; 7 0xF878 0xF878 0x06D9 ASLIST
0974      147 ; 7 0xFF51 0xFF51 0x00AF unused
0974      148 ;
0974      149 ;
0974      150 ; An EPROM catalog may be any size in order to accommodate
0974      151 ; all of the files that are contained in that EPROM. The
0974      152 ; EPROM data for its files may start at any EPROM offset
0974      153 ; after the catalog terminating NULL byte. The layout
0974      154 ; design of a Data EPROM could be constructed as follows:
0974      155 ;
0974      156 ; Bank Offset Memory Size Description
0974      157 ; ----
0974      158 ; 0 0x0000 0xE000 0x0004 Sync bytes
0974      159 ;      0x0004 0xE004 0x0174 Catalog
0974      160 ;      0x0178 0xE178 0x0345 blank space
0974      161 ;      0x0523 0xE523 0x1234 data for first file
0974      162 ;      0x1757 0xF757 0x4321 data for second file
0974      163 ; 2 0x5A78 0xFA78 0xA588 unused
0974      164 ;
0974      165 ;
0974      166 ; The four sync bytes 0xC4, 0xB8, 0x90, and 0xED must
0974      167 ; preface each and every EPROM on an EPROM card. These
0974      168 ; sync bytes are necessary in order for the EOS+ catalog
0974      169 ; function, the Applesoft interface ASEOS, and the assembly
0974      170 ; language interface BINEOS to "discover" a valid EPROM
0974      171 ; that contains an EPROM catalog. The Ampex Corporation
0974      172 ; determined that these particular byte values are totally
0974      173 ; unique and would never be found naturally in random
0974      174 ; data. Ampex utilized the uniqueness of these four bytes
0974      175 ; in the recording of instrumented data onto their digital
0974      176 ; tape recording hardware. The utilization of these four
0974      177 ; bytes in recording instrumented digital data is
0974      178 ; proprietary to the Ampex Corporation.
0974      179 ;
0974      180 ;
0974      181 ; EOS+ is enabled when the Apple computer is first powered
0974      182 ; ON or when the RESET key is pressed. A menu is displayed

```

```
0974      183 ; showing all of the available options. Options 0-7
0974      184 ; display a catalog of the contents of the respective EPROM
0974      185 ; in order to load or to run any of the programs that are
0974      186 ; contained on that EPROM. CTRL-C enables the SDV editor
0974      187 ; in order to change the Slot, Drive, and Volume values to
0974      188 ; attach or detach that slot from DOS or to Catalog or Run
0974      189 ; the HELLO file on a Disk ][ drive that resides in that
0974      190 ; Slot with that Volume number. The RETURN key toggles
0974      191 ; whether the ZipChip is enabled or not enabled if the
0974      192 ; ZipChip exists in that computer.
0974      193 ;
0974      194 ; To configure the EPROM card to enable EPROM 0 and Bank 0
0974      195 ; while the EPROM card is turned OFF, enter the following
0974      196 ; Monitor command that is based on which slot the EPROM
0974      197 ; card resides:
0974      198 ;
0974      199 ;      Slot      quikLoader      EPROM Reader
0974      200 ;      ----      -
0974      201 ;          2      %C0A0:10      %C0A0:80
0974      202 ;          3      %C0B0:10      %C0B0:80
0974      203 ;          4      %C0C0:10      %C0C0:80
0974      204 ;          5      %C0D0:10      %C0D0:80
0974      205 ;          6      %C0E0:10      %C0E0:80
0974      206 ;          7      %C0F0:10      %C0F0:80
0974      207 ;
0974      208 ;
0974      209 ; EOS+ may also be directly entered from the Monitor. For
0974      210 ; example, if an EPROM card resides in Slot 4, the Monitor
0974      211 ; command C4F0G will enable EOS+ and display the EOS+ Main
0974      212 ; Menu. Enter the following Monitor command that is based
0974      213 ; on the slot in which the EPROM card resides:
0974      214 ;
0974      215 ;      Slot 1 - %C1F0G
0974      216 ;      Slot 2 - %C2F0G
0974      217 ;      Slot 3 - %C3F0G
0974      218 ;      Slot 4 - %C4F0G
0974      219 ;      Slot 5 - %C5F0G
0974      220 ;      Slot 6 - %C6F0G
0974      221 ;      Slot 7 - %C7F0G
0974      222 ;
0974      223 ;
0974      224 ; In view of the fact that it is easy to obtain 512 KB
0974      225 ; EPROMs, there is plenty of room on seven additional
0974      226 ; EPROMs of that size that can contain all of the
0974      227 ; "necessary" programs for the average Apple developer.
0974      228 ; However, the EPROM Reader hardware is designed to allow
0974      229 ; the DMA IN and DMA OUT chaining of multiple EPROM cards.
0974      230 ; EOS+ was redesigned in order to allow the user to access
0974      231 ; any program from any EPROM that is within multiple EPROM
0974      232 ; cards that follow the standard convention in using DMA IN
0974      233 ; and DMA OUT protocol.
0974      234 ;
0974      235 ; When it is desired to chain multiple EPROM cards there
0974      236 ; cannot be any empty slots between these EPROM cards. If
0974      237 ; there are non-EPROM cards between the chained EPROM
0974      238 ; cards, these non-EPROM cards must connect their DMA IN
0974      239 ; port to their DMA OUT port. The highest priority EPROM
0974      240 ; card, or the EPROM card that resides in the lowest card
0974      241 ; slot must not connect or switch in DMA IN; this card
0974      242 ; must connect or switch in DMA OUT. The EPROM card that
0974      243 ; resides in the highest card slot must not connect or
```

```
0974          244 ; switch in DMA OUT; this card must connect or switch in
0974          245 ; DMA IN. All other cards must connect or switch in both
0974          246 ; DMA IN and DMA OUT.
0974          247 ;
0974          248 ;
0974          249      icl "INCL.L"
```

```
LLOAD INCL.L,A$4000
```



```

0974      1          ttl "EOS+ Source Code,INCL.L"
0974      2      ;
0974      3      ;
0974      4      ; INCL.L
0974      5      ;
0974      6      ;
0006      7  XREG      epz $06
0007      8  YREG      epz $07
0008      9  AREG      epz $08
0016     10  XSAV      epz $16
0017     11  YSAV      epz $17
0018     12  ASAV      epz $18
0974     13      ;
0020     14  WNDLFT     epz $20
0021     15  WNDWDTH    epz $21
0022     16  WNDTOP     epz $22
0023     17  WNDBTM     epz $23
0024     18  CH         epz $24
0025     19  CV         epz $25
0974     20      ;
002A     21  SRCPTR     epz $2A
002C     22  LENPTR     epz $2C
002E     23  DSTPTR     epz $2E
0974     24      ;
0033     25  PROMPT     epz $33
0036     26  CSWL       epz $36
0038     27  KSWL       epz $38
0974     28      ;
0067     29  ASPGMST    epz $67
0069     30  ASVARS     epz $69
0974     31      ;
0075     32  CURLIN     epz $75
0974     33      ;
009D     34  DSCTMP     epz $9D
0974     35      ;
00AF     36  ASPEND     epz $AF
0974     37      ;
00B7     38  CHRGOT     epz $B7
00B8     39  CHRADR     epz $B8
0974     40      ;
00CE     41  GENPTR     epz $CE
0974     42      ;
00D8     43  ASONERR    epz $D8
0974     44      ;
00EB     45  MSLOT      epz $EB
00EC     46  DRIVE      epz $EC
00ED     47  VOLUME     epz $ED
00EE     48  CMDPTR     epz $EE
0974     49      ;
00FA     50  EXECPTR    epz $FA
00FC     51  PRNTPTR    epz $FC
0974     52      ;
0974     53          enz
0974     54      ;
0000     55  QLCARD     equ 0
0001     56  EPCARD     equ 1
0974     57      ;
0000     58  ZERO       equ $00
00FF     59  NEGONE     equ $FF
0974     60      ;

```

```
0007      61  QLMASK      equ  $07
000F      62  EPMASK      equ  $0F
000F      63  PRNTMASK    equ  $0F
000F      64  VALUMASK    equ  $0F
001F      65  MENUMASK    equ  $1F
001F      66  CVMASK      equ  $1F
001F      67  BANKMASK    equ  $1F
007F      68  MSBCLR      equ  $7F
0080      69  MSBSET      equ  $80
0974      70  ;
0083      71  CTRLC      equ  $83
0084      72  CTRLD      equ  $84
0088      73  LARROW      equ  $88
008A      74  DARROW      equ  $8A
008B      75  UARROW      equ  $8B
008D      76  RETURN      equ  $8D
0093      77  CTRLS      equ  $93
0095      78  RARROW      equ  $95
009B      79  ESCAPE      equ  $9B
00A0      80  SPACE      equ  $A0
00A4      81  DOLLAR      equ  $A4
00AC      82  COMMA      equ  $AC
00DF      83  LWRMASK     equ  $DF
00E0      84  LWRCASE     equ  $E0
0974      85  ;
0003      86  SLOT3      equ  3
0974      87  ;
0005      88  INDENT      equ  5
0006      89  PARMSIZE    equ  6
0008      90  DCBSIZE     equ  8
0018      91  NAMESIZE    equ  24
0020      92  ENTRYLEN    equ  32
0974      93  ;
0004      94  ASPNUM4     equ  4
0005      95  ASPNUM5     equ  5
0006      96  ASPNUM6     equ  6
0974      97  ;
000C      98  MAXASNUM    equ  2*ASPNUM6
0974      99  ;
0001     100  INTERNAL    equ  $01
0002     101  EXTERNAL    equ  $02
0974     102  ;
0001     103  LOADCMD     equ  $01
0002     104  RUNCMD      equ  $02
0003     105  CATCMD      equ  $03
0070     106  QLSRCH      equ  $70
00F0     107  EPSRCH      equ  $F0
0974     108  ;
0000     109  ERR00       equ  $00
0001     110  ERR01       equ  $01
0002     111  ERR02       equ  $02
0003     112  ERR03       equ  $03
0004     113  ERR04       equ  $04
0005     114  ERR05       equ  $05
0974     115  ;
0020     116  TESTCNT     equ  $20
0974     117  ;
0050     118  MAXCH        equ  $50
0060     119  MINCV        equ  $60
0974     120  ;
0050     121  RTNCMD      equ  $50
```

0051	122	NORMCMD	equ	\$51
0052	123	INITCMD	equ	\$52
0053	124	VIDCMD	equ	\$53
0054	125	KBDCMD	equ	\$54
0055	126	HOMECMD	equ	\$55
0056	127	TABVCMD	equ	\$56
0057	128	EOLCMD	equ	\$57
0058	129	EOPCMD	equ	\$58
0059	130	CNTRCMD	equ	\$59
0974	131	;		
0000	132	EPONVAL	equ	\$00
0008	133	EPUSR	equ	\$08
0010	134	QLOFFVAL	equ	\$10
0080	135	EPOFFVAL	equ	\$80
0974	136	;		
0000	137	ZCONVAL	equ	\$00
000C	138	ZCOPTNS	equ	12
0010	139	ZCOFFVAL	equ	\$10
0010	140	ZCSTAT	equ	\$10
0004	141	ZCNSPEED	equ	4
005A	142	ZCUNLOCK	equ	\$5A
00A5	143	ZCLOCK	equ	\$A5
0974	144	;		
0000	145	ENDCAT	equ	\$00
0001	146	TEXTFILE	equ	\$01
0002	147	APLSOFT	equ	\$02
0004	148	BINARY0	equ	\$04
0008	149	BINARY1	equ	\$08
0010	150	BINARY2	equ	\$10
0020	151	RESERVED	equ	\$20
0040	152	SYSTEM	equ	\$40
0080	153	PRIMARY	equ	\$80
0974	154	;		
00C4	155	SYNCBYT0	equ	\$C4
00B8	156	SYNCBYT1	equ	\$B8
0090	157	SYNCBYT2	equ	\$90
00ED	158	SYNCBYT3	equ	\$ED
0974	159	;		
00FF	160	RUNMODE	equ	\$FF
0974	161	;		
0100	162	STACK	equ	\$100
0100	163	PAGESIZE	equ	\$100
0110	164	STKCODE	equ	\$110
0974	165	;		
0200	166	INPUT	equ	\$200
0974	167	;		
0290	168	PRISLOT	equ	\$290
0291	169	EPSLOT	equ	\$291
0292	170	SLOTMAP	equ	\$292
0293	171	APPLTYPE	equ	\$293
0974	172	;		
0294	173	EPNMBR	equ	\$294
0295	174	EPBANK	equ	\$295
0974	175	;		
0296	176	RTNTYPE	equ	\$296
0297	177	TEMPVAL	equ	\$297
0974	178	;		
0298	179	EPSTRT	equ	\$298
0299	180	EPEND	equ	\$299
0974	181	;		
029A	182	ZSTATUS	equ	\$29A

```

029B      183  ZCACHE      equ  $29B
0974      184  ;
029C      185  NUMIN       equ  $29C
029D      186  NUMSELC     equ  $29D
0974      187  ;
029E      188  FLENGTH     equ  $29E
029F      189  RUNFLAG     equ  $29F
0974      190  ;
02A0      191  ASPRNUM     equ  $2A0
02A1      192  ASSTATUS    equ  $2A1
02A2      193  EPSEARCH    equ  $2A2
02A3      194  FILELEN     equ  $2A3
02A4      195  SLOT16      equ  $2A4
02A5      196  SLOT        equ  $2A5
0974      197  ;
02A6      198  MEMJMP      equ  $2A6          ; 2 bytes
02A8      199  SLOTJMP     equ  $2A8          ; 2 bytes
0974      200  ;
02AA      201  ADDRBUFR    equ  $2AA          ; 2 bytes
02AC      202  SYNCBUFR    equ  $2AC          ; 4 bytes
0974      203  ;
02B0      204  FILEENTRY   equ  $2B0          ; 32 bytes
02B0      205  FILEPNUM     equ  $2B0
02B1      206  FILETYPE     equ  $2B1
02B2      207  SRCVAL       equ  $2B2
02B4      208  LENVAL       equ  $2B4
02B6      209  DSTVAL       equ  $2B6
02B8      210  FILENAME     equ  $2B8          ; 24 bytes maximum
0974      211  ;
0974      212  ;
0974      213  ; Common variable area.
0974      214  ;
02D0      215  ZCSETBL      equ  $2D0          ; ZIP, 12 bytes
0974      216  ;
02D0      217  NUMSCRN      equ  $2D0          ; catalog function
02D1      218  FIRSTIME     equ  $2D1
02D2      219  FILECNT      equ  $2D2
02D3      220  NUMNTRYS     equ  $2D3
02D4      221  LSTOPNTY     equ  $2D4
02D5      222  NTRYSTRT     equ  $2D5
02D6      223  NTRYEND      equ  $2D6
02D7      224  FILTYPE      equ  $2D7
02D8      225  INDEX        equ  $2D8
0974      226  ;
02D0      227  ASPRADRS     equ  $2D0          ; ASEOS
02D0      228  ASPCMD       equ  $2D0
02D2      229  ASPSTAT      equ  $2D2
02D4      230  ASPSRCH      equ  $2D4
02D6      231  ASPFILE      equ  $2D6
02D6      232  ASPNUM       equ  $2D6
02D8      233  ASPADR       equ  $2D8
02D8      234  ASPFILES     equ  $2D8
02DA      235  ASPPARMS     equ  $2DA
0974      236  ;
02D0      237  DCBBUFR      equ  $2D0          ; BINEOS (32 bytes total)
02D0      238  DCBCMD       equ  $2D0          ; command
02D1      239  DCBEPN       equ  $2D1          ; EPROM search
02D2      240  DCBFALT      equ  $2D2          ; alternate load address
02D4      241  DCBSTAT      equ  $2D4          ; return status
02D5      242  DCBFLEN      equ  $2D5          ; cat entries/filename length
02D6      243  DCBFADR      equ  $2D6          ; cat buffer/filename address

```

```
0974      244 ;
0974      245 ;
0974      246 ; Page 3 vectors.
0974      247 ;
03D0      248 DOSWARM equ $3D0
03D3      249 DOSCOLD equ $3D3
03EA      250 HOOKDOS equ $3EA
0974      251 ;
04FB      252 XMODE equ $4FB
0974      253 ;
0974      254 ;
0974      255 ; Private slot variables indexed by PRISLOT.
0974      256 ;
0478      257 PWRUP0 equ $478
0578      258 PWRUP1 equ $578
0678      259 PWRUP2 equ $678
0778      260 PWRUP3 equ $778
0974      261 ;
0974      262 ;
0974      263 ; Screen line addresses.
0974      264 ;
0400      265 LINE00 equ $400
0480      266 LINE01 equ $480
0500      267 LINE02 equ $500
0580      268 LINE03 equ $580
0600      269 LINE04 equ $600
0680      270 LINE05 equ $680
0700      271 LINE06 equ $700
0780      272 LINE07 equ $780
0428      273 LINE08 equ $428
04A8      274 LINE09 equ $4A8
0528      275 LINE10 equ $528
05A8      276 LINE11 equ $5A8
0628      277 LINE12 equ $628
06A8      278 LINE13 equ $6A8
0728      279 LINE14 equ $728
07A8      280 LINE15 equ $7A8
0450      281 LINE16 equ $450
04D0      282 LINE17 equ $4D0
0550      283 LINE18 equ $550
05D0      284 LINE19 equ $5D0
0650      285 LINE20 equ $650
06D0      286 LINE21 equ $6D0
0750      287 LINE22 equ $750
07D0      288 LINE23 equ $7D0
0974      289 ;
0801      290 STARTAS equ $801
0974      291 ;
2000      292 BANKSIZE equ $2000
0974      293 ;
BFF6      294 MNGUSER equ $BFF6
BFF8      295 INITDOS equ $BFF8
0974      296 ;
0800      297 PAGE08 equ $0800
0900      298 PAGE09 equ $0900
2000      299 PAGE20 equ $2000
6000      300 PAGE60 equ $6000
8000      301 PAGE80 equ $8000
9F00      302 PAGE9F equ $9F00
BE00      303 PAGEBE equ $BE00
C000      304 PAGEC0 equ $C000
```

C100	305	PAGEC1	equ	\$C100
C700	306	PAGEC7	equ	\$C700
D000	307	PAGED0	equ	\$D000
DE00	308	PAGEDE	equ	\$DE00
E000	309	PAGEE0	equ	\$E000
E100	310	PAGEE1	equ	\$E100
E700	311	PAGEE7	equ	\$E700
E800	312	PAGEE8	equ	\$E800
EA00	313	PAGEEA	equ	\$EA00
0974	314	;		
C000	315	KEY	equ	\$C000
C000	316	STR80OFF	equ	\$C000
C002	317	RAMRDOFF	equ	\$C002
C004	318	RAMWROFF	equ	\$C004
C006	319	CXROMOFF	equ	\$C006
C007	320	CXROMON	equ	\$C007
C008	321	AUXZPOFF	equ	\$C008
C00A	322	C3ROMOFF	equ	\$C00A
C00B	323	C3ROMON	equ	\$C00B
C00C	324	VID80OFF	equ	\$C00C
C00E	325	ALTCHOFF	equ	\$C00E
0974	326	;		
C015	327	RDCXROM	equ	\$C015
0974	328	;		
C010	329	CLRKEY	equ	\$C010
C030	330	SPKR	equ	\$C030
0974	331	;		
C010	332	HOOKSLT	equ	\$C010
C018	333	UHOOKSLT	equ	\$C018
0974	334	;		
C051	335	TEXTON	equ	\$C051
C054	336	PAGE1ON	equ	\$C054
C056	337	HIRESOFF	equ	\$C056
0974	338	;		
C058	339	ANN1OFF	equ	\$C058
C05A	340	ANN2OFF	equ	\$C05A
C05D	341	ANN3ON	equ	\$C05D
C05F	342	ANN4ON	equ	\$C05F
0974	343	;		
C05A	344	ZCCTRL	equ	\$C05A
C05B	345	ZCSTATS	equ	\$C05B
C05C	346	ZCSLOTS	equ	\$C05C
C05D	347	ZCSPEED	equ	\$C05D
C05E	348	ZCDELAY	equ	\$C05E
C05F	349	ZCCACHE	equ	\$C05F
0974	350	;		
C080	351	LCSELC	equ	\$C080
C080	352	EPSELC	equ	\$C080
0974	353	;		
C080	354	RAM2WP	equ	\$C080
C081	355	ROM2WE	equ	\$C081
C082	356	ROM2WP	equ	\$C082
C083	357	RAM2WE	equ	\$C083
C088	358	RAM1WP	equ	\$C088
C089	359	ROM1WE	equ	\$C089
C08A	360	ROM1WP	equ	\$C08A
C08B	361	RAM1WE	equ	\$C08B
0974	362	;		
CFFF	363	CLRROM	equ	\$CFFF
0974	364	;		
D566	365	RUNAS	equ	\$D566

DEBE	366	CHKCOM	equ	\$DEBE
DFE3	367	PTRGET	equ	\$DFE3
0974	368	;		
E000	369	LISASTRT	equ	\$E000
E3D5	370	STRINI	equ	\$E3D5
0974	371	;		
F941	372	PRNTAX	equ	\$F941
FA62	373	RSETADR1	equ	\$FA62
FB2F	374	INIT	equ	\$FB2F
FC22	375	VTAB	equ	\$FC22
FC42	376	CLREOP	equ	\$FC42
FC58	377	HOME	equ	\$FC58
FC9C	378	CLREOL	equ	\$FC9C
FD8E	379	CROUT	equ	\$FD8E
FDDA	380	PRBYTE	equ	\$FDDA
FDE3	381	PRHEX	equ	\$FDE3
FDED	382	COUT	equ	\$FDED
FE84	383	SETNORM	equ	\$FE84
FE89	384	SETKBD	equ	\$FE89
FE8B	385	INPORT	equ	\$FE8B
FE93	386	SETVID	equ	\$FE93
FE95	387	OUTPORT	equ	\$FE95
FF59	388	RSETADR2	equ	\$FF59
FF65	389	MONITOR	equ	\$FF65
FFFA	390	EOSVCTRS	equ	\$FFFA
0974	391	;		
0974	392	;		
0004	393	SYNC.L	equ	4
0974	394	;		
3000	395	ROM.L	equ	\$3000
D000	396	ROM.D	equ	\$D000
0974	397	;		
2100	398	DOSL.L	equ	\$2100
9F00	399	DOSL.D	equ	\$9F00
0974	400	;		
2A00	401	DOSH.L	equ	\$2A00
BE00	402	DOSH.D	equ	\$BE00
0974	403	;		
01C0	404	LLISA8.L	equ	\$01C0
0900	405	LLISA8.D	equ	\$0900
0974	406	;		
2800	407	LISA81.L	equ	\$2800
D000	408	LISA81.D	equ	\$D000
0974	409	;		
1000	410	LISA82.L	equ	\$1000
D000	411	LISA82.D	equ	\$D000
0974	412	;		
0640	413	LISA83.L	equ	\$0640
B7C0	414	LISA83.D	equ	\$B7C0
0974	415	;		
1914	416	SETUP8.L	equ	\$1914
0900	417	SETUP8.D	equ	\$0900
0974	418	;		
1B04	419	RAMDSK.L	equ	\$1B04
4000	420	RAMDSK.D	equ	\$4000
0974	421	;		
1316	422	FID.L	equ	\$1316
0900	423	FID.D	equ	\$0900
0974	424	;		
064A	425	CLK.L	equ	\$064A
0900	426	CLK.D	equ	\$0900

```
0974          427  ;  
06D9          428  LIST.L    equ  $06D9  
8800          429  LIST.D    equ  $8800  
0974          430  ;  
0974          431  ;  
0974          432          icl  "CATALOG.L"
```

```
LLOAD CATALOG.L,A$4000
```



```

0974          1          ttl "EOS+ Source Code, CATALOG.L"
0974          2          ;
0974          3          ;
0974          4          ; CATALOG.L
0974          5          ;
0974          6          ;
0001          7  DEBUG      equ 1
0001          8  HWCARD     equ EPCARD
0974          9          ;
0974         10          ;
0974         11          .if DEBUG
0800         12          org PAGE08
0800         13          .el
0800         14          org PAGEE0
0800         15          .fi
0800         16          ;
0800         17          obj PAGE08
0800         18          usr
0800         19          ;
0800         20          ;
0800         21          .if HWCARD
00F0         22  SRCHALL    equ EPSRCH
0800         23          .el
00F0         24  SRCHALL    equ QLSRCH
0800         25          .fi
0800         26          ;
0800         27          ;
0800         28  CATALOG:
0800         29          ;
0800 C4         30          byt SYNCBYT0
0801 B8         31          byt SYNCBYT1
0802 90         32          byt SYNCBYT2
0803 ED         33          byt SYNCBYT3
0804         34          ;
0804         35          ;
0804 44         36  DOSLPRMS byt SYSTEM|BINARY0
0805 00 20       37          adr DOSL.O
0807 00 21       38          adr DOSL.L
0809 00 9F       39          adr DOSL.D
080B 44 4F 53    40          dci `DOS4.5.05L`
080E 34 2E 35
0811 2E 30 35
0814 CC
0815          41          ;
0815          42          ;
0815 5C         43  DOSHPRMS byt SYSTEM|BINARY0|BINARY1|BINARY2
0816 00 41       44          adr DOSH.O
0818 00 2A       45          adr DOSH.L
081A 00 BE       46          adr DOSH.D
081C 44 4F 53    47          dci `DOS4.5.06H`
081F 34 2E 35
0822 2E 30 36
0825 C8
0826          48          ;
0826          49          ;
0826 84         50  LISAPRMS byt PRIMARY|BINARY0
0827 54 C2       51          adr LLISA8.O
0829 C0 01       52          adr LLISA8.L
082B 00 09       53          adr LLISA8.D
082D 4C 4F 41    54          dci `LOADLISA80`

```

```

0830 44 4C 49
0833 53 41 38
0836 B0
0837          55 ;
0837          56 ;
0837 48          57 LISA1PRM byt SYSTEM|BINARY1
0838 00 6B          58          adr LISA81.O
083A 00 28          59          adr LISA81.L
083C 00 D0          60          adr LISA81.D
083E 4C 49 53      61          dci 'LISA80.1'
0841 41 38 30
0844 2E B1
0846          62 ;
0846          63 ;
0846 50          64 LISA2PRM byt SYSTEM|BINARY2
0847 00 93          65          adr LISA82.O
0849 00 10          66          adr LISA82.L
084B 00 D0          67          adr LISA82.D
084D 4C 49 53      68          dci 'LISA80.2'
0850 41 38 30
0853 2E B2
0855          69 ;
0855          70 ;
0855 44          71 LISA3PRM byt SYSTEM|BINARY0
0856 00 A3          72          adr LISA83.O
0858 40 06          73          adr LISA83.L
085A C0 B7          74          adr LISA83.D
085C 4C 49 53      75          dci 'LISA80.3'
085F 41 38 30
0862 2E B3
0864          76 ;
0864          77 ;
0864 04          78 SETUPRMS byt BINARY0
0865 40 A9          79          adr SETUP8.O
0867 14 19          80          adr SETUP8.L
0869 00 09          81          adr SETUP8.D
086B 53 45 54      82          dci 'SETUP80'
086E 55 50 38
0871 B0
0872          83 ;
0872          84 ;
0872 04          85 RMDSKPRM byt BINARY0
0873 14 C4          86          adr RAMDSK.O
0875 04 1B          87          adr RAMDSK.L
0877 00 40          88          adr RAMDSK.D
0879 52 61 6D      89          dci 'RamDisk Config'
087C 44 69 73
087F 6B 20 43
0882 6F 6E 66
0885 69 E7
0887          90 ;
0887          91 ;
0887 04          92 FIDPARMS byt BINARY0
0888 18 DF          93          adr FID.O
088A 16 13          94          adr FID.L
088C 00 09          95          adr FID.D
088E 46 49 C4      96          dci 'FID'
0891          97 ;
0891          98 ;
0891 04          99 CLKPARMS byt BINARY0
0892 2E F2        100          adr CLK.O

```

```

0894 4A 06      101      adr CLK.L
0896 00 09      102      adr CLK.D
0898 53 65 74   103      dci 'Set Clock'
089B 20 43 6C
089E 6F 63 EB
08A1           104      ;
08A1           105      ;
08A1 04         106      LISTPRMS byt BINARY0
08A2 78 F8      107      adr LIST.O
08A4 D9 06      108      adr LIST.L
08A6 00 88      109      adr LIST.D
08A8 41 70 70   110      dci 'Applesoft Formatter'
08AB 6C 65 73
08AE 6F 66 74
08B1 20 46 6F
08B4 72 6D 61
08B7 74 74 65
08BA F2
08BB           111      ;
08BB           112      ;
08BB 20         113      ROMPARMS byt RESERVED
08BC 00 D0      114      adr ROM.D
08BE 00 30      115      adr ROM.L
08C0 00 D0      116      adr ROM.D
08C2 52 4F 4D   117      dci 'ROM Copy'
08C5 20 43 6F
08C8 70 F9
08CA           118      ;
08CA           119      ;
08CA 20         120      CATPARMS byt RESERVED
08CB 00 00      121      adr CATALOG&BANKMASK
08CD 04 00      122      adr SYNC.L
08CF AC 02      123      adr SYNCBUFR
08D1 43 61 74   124      dci 'Catalog Sync'
08D4 61 6C 6F
08D7 67 20 53
08DA 79 6E E3
08DD           125      ;
08DD           126      ;
08DD 00         127      byt ENDCAT
08DE           128      ;
08DE           129      ;
08DE           130      dfs PAGESIZE-*)&NEGONE,NEGONE
0900           131      ;
0900           132      ;
0900           133      .if DEBUG
0900           134      .el
0900           135      phs PAGEC1
0900           136      .fi
0900           137      ;
0900           138      ;
0900           139      icl "SLOT1.L"

```

```

LLOAD SLOT1.L,A$4000

```

```

0900      1          ttl "EOS+ Source Code, SLOT1.L"
0900      2      ;
0900      3      ;
0900      4      ; SLOT1.L
0900      5      ;
0900      6      ;
0001      7  HWSLOT    let 1
0010      8  HWSLOT16 let $10
00C1      9  HWSLOT16 let $C1
0900     10      ;
0000     11  LABEL     let 0
0900     12      ;
0900     13      ;
0900     14      ; This is the generic code that is assembled specifically
0900     15      ; for each of the seven slots in which an EPROM card may
0900     16      ; reside.
0900     17      ;
0900     18      ; Interface to process an ASEOS command.
0900     19      ;
0900     20          .if LABEL
0900     21  EPASEOS    sta ASAV
0900     22          .el
0900 85 18     23          sta ASAV
0902     24          .fi
0902     25      ;
0902 18      26          clc
0903 90 03    27          bcc >0                ; always taken
0905     28      ;
0905     29      ;
0905     30      ; Insert TESTROM verification code here.  When the CXRESET
0905     31      ; routine is entered, it calls TSTROMCD to test for a ROM
0905     32      ; card.  TSTROMCD calls TESTROM that looks for 0x38xx18 at
0905     33      ; 0xC305.  If found, C3ROMOFF is enabled.
0905     34      ;
0905 38      35          sec
0906 90 00    36          bcc *+2
0908     37          dfs !-1
0907 18      38          clc
0908     39      ;
0908     40      ;
0908 86 16     41  ^0      stx XSAV
090A 84 17    42          sty YSAV
090C     43      ;
090C A5 76    44          lda CURLIN+1
090E C9 FF    45          cmp #RUNMODE
0910 D0 03    46          bne >2
0912     47      ;
0912 A5 18     48  ^1      lda ASAV                ; recall A-reg
0914     49      ;
0914 60      50          rts
0915     51      ;
0915 A2 00     52  ^2      ldx #ZERO
0917     53      ;
0917 8E A0 02  54  ^3      stx ASPRNUM
091A     55      ;
091A 20 B7 00  56          jsr CHRGOT
091D F0 21     57          beq >4
091F     58      ;
091F 20 BE DE  59          jsr CHKCOM
0922     60      ;

```

```

0922 20 B7 00      61      jsr CHRGOT
0925 F0 19        62      beq >4
0927              63      ;
0927 C9 2C        64      cmp #COMMA&MSBCLR
0929 F0 15        65      beq >4
092B              66      ;
092B 20 E3 DF     67      jsr PTRGET
092E              68      ;
092E AE A0 02     69      ldx ASPRNUM
0931 E0 0C        70      cpx #MAXASNUM      ; too many parameters
0933 F0 DD        71      beq <1
0935              72      ;
0935 9D D0 02     73      sta ASPRADRS,X
0938              74      ;
0938 98           75      tya
0939 9D D1 02     76      sta ASPRADRS+1,X
093C              77      ;
093C E8           78      inx
093D E8           79      inx
093E              80      ;
093E D0 D7        81      bne <3      ; always taken
0940              82      ;
0940 4E A0 02     83      ^4      lsr ASPRNUM
0943 F0 CD        84      beq <1      ; no parameters
0945              85      ;
0945 A9 00        86      lda #EPONVAL
0947 8D 90 C0     87      sta EPSELC+HWSLOT16
094A              88      ;
094A              89      ;
094A              90      ; Initialize X-reg with this slot number and enter ASEOS.
094A              91      ;
094A A2 01        92      ldx #HWSLOT
094C              93      ;
094C 4C 16 1A     94      jmp ASEOS
094F              95      ;
094F              96      ;
094F              97      dfs $50-*)&NEGONE,NEGONE
0950              98      ;
0950              99      ;
0950            100      ; Exit for ASEOS and to RUN Applesoft files. If a ZipChip
0950            101      ; is present, flush cache, enable ZipChip, and return to
0950            102      ; the external caller or address in MEMJMP.
0950            103      ;
0950            104      .if LABEL
0950            105      ASEXIT  lda ZSTATUS
0950            106      .el
0950 AD 9A 02      107      lda ZSTATUS
0953            108      .fi
0953            109      ;
0953 30 14        110      bmi >2
0955            111      ;
0955 A0 00        112      ldy #ZERO
0957 A9 60        113      lda /PAGE60
0959            114      ;
0959 84 CE        115      sty GENPTR
095B 85 CF        116      sta GENPTR+1
095D            117      ;
095D B1 CE        118      ^1      lda (GENPTR),Y
095F            119      ;
095F C8           120      iny
0960 D0 FB        121      bne <1

```

```

0962          122 ;
0962 E6 CF      123      inc GENPTR+1
0964 10 F7      124      bpl <1
0966          125 ;
0966 20 E1 1C    126      jsr DOZCON
0969          127 ;
0969          128      .if HWCARD
0969 A9 80        129 ^2      lda #EPOFFVAL
096B          130      .el
096B          131 ^2      lda #QLOFFVAL
096B          132      .fi
096B          133 ;
096B 8D 90 C0    134      sta EPSELC+HWSLOT16
096E          135 ;
096E A5 18      136      lda ASAV
0970 A6 16      137      ldx XSAV
0972 A4 17      138      ldy YSAV
0974          139 ;
0974 6C A6 02    140      jmp (MEMJMP)
0977          141 ;
0977          142 ;
0977          143      dfs $80-*)&NEGONE,NEGONE
0980          144 ;
0980          145 ;
0980          146 ; Exit for BINEOS and to RUN binary files. If a ZipChip is
0980          147 ; present, flush cache, enable ZipChip, push DOSWARM onto
0980          148 ; the stack, load the address for EPBINEOS, and return to
0980          149 ; the external caller or address in MEMJMP.
0980          150 ;
0980          151 ; Load the Y-reg/A-reg with the address of EPBINEOS in
0980          152 ; order for the current file in memory to process more
0980          153 ; DCBs if other EPROM files need to be loaded into memory.
0980          154 ;
0980          155      .if LABEL
0980          156 BINEXIT  lda ZSTATUS
0980          157      .el
0980 AD 9A 02      158      lda ZSTATUS
0983          159      .fi
0983          160 ;
0983 30 14        161      bmi >2
0985          162 ;
0985 A0 00        163      ldy #ZERO
0987 A9 60        164      lda /PAGE60
0989          165 ;
0989 84 CE        166      sty GENPTR
098B 85 CF        167      sta GENPTR+1
098D          168 ;
098D B1 CE        169 ^1      lda (GENPTR),Y
098F          170 ;
098F C8          171      iny
0990 D0 FB        172      bne <1
0992          173 ;
0992 E6 CF        174      inc GENPTR+1
0994 10 F7        175      bpl <1
0996          176 ;
0996 20 E1 1C    177      jsr DOZCON
0999          178 ;
0999 A2 FF        179 ^2      ldx #NEGONE
099B 9A          180      txs
099C          181 ;
099C A9 03        182      lda /DOSWARM-1

```

```

099E 48          183          pha
099F          184          ;
099F A9 CF      185          lda #DOSWARM-1
09A1 48          186          pha
09A2          187          ;
09A2 A0 E0      188          ld y #EPBINEOS
09A4 A9 C1      189          lda #HWSLOT16
09A6          190          ;
09A6          191          ;
09A6 EA        192          nop
09A7 EA        193          nop
09A8          194          ;
09A8          195          ;
09A8          196          ; Exit from EOS with the assumption that a return to EOS
09A8          197          ; will be made by means of EPMAPEOS. The X-reg must be
09A8          198          ; used in order to turn this EPROM card OFF.
09A8          199          ;
09A8          200          .if LABEL
09A8          201          RTNEXIT:
09A8          202          .fi
09A8          203          ;
09A8          204          .if HWCARD
09A8 A2 80      205          ldx #EPOFFVAL
09AA          206          .el
09AA          207          ldx #QLOFFVAL
09AA          208          .fi
09AA          209          ;
09AA 8E 90 C0   210          stx EPSELC+HWSLOT16
09AD          211          ;
09AD A2 10      212          ldx #HWSLOT16
09AF          213          ;
09AF 6C A6 02   214          jmp (MEMJMP)
09B2          215          ;
09B2          216          ;
09B2          217          dfs $B8-*&NEGONE,NEGONE
09B8          218          ;
09B8          219          ;
09B8          220          ; Special entrance in order to turn this EPROM card OFF.
09B8          221          ;
09B8          222          .if LABEL
09B8          223          EPOFF:
09B8          224          .fi
09B8          225          ;
09B8          226          .if HWCARD
09B8 A9 80      227          lda #EPOFFVAL
09BA          228          .el
09BA          229          lda #QLOFFVAL
09BA          230          .fi
09BA          231          ;
09BA 8D 90 C0   232          sta EPSELC+HWSLOT16
09BD          233          ;
09BD 60         234          rts
09BE          235          ;
09BE          236          ;
09BE          237          dfs $C0-*&NEGONE,NEGONE
09C0          238          ;
09C0          239          ;
09C0          240          ; Return from DOS CMDUSER 1 command.
09C0          241          ;
09C0          242          .if LABEL
09C0          243          EPUSER1 lda #EPONVAL

```

```

09C0          244          .el
09C0 A9 00      245          lda #EPONVAL
09C2          246          .fi
09C2          247          ;
09C2 8D 90 C0   248          sta EPSELC+HWSLOT16
09C5          249          ;
09C5 4C 18 11   250          jmp USERRTN1
09C8          251          ;
09C8          252          ;
09C8          253          ; Return from DOS CMDUSER 2 command.
09C8          254          ;
09C8          255          .if LABEL
09C8          256 EPUSER2   lda #EPONVAL
09C8          257          .el
09C8 A9 00      258          lda #EPONVAL
09CA          259          .fi
09CA          260          ;
09CA 8D 90 C0   261          sta EPSELC+HWSLOT16
09CD          262          ;
09CD 4C 16 14   263          jmp USERRTN2
09D0          264          ;
09D0          265          ;
09D0          266          ; Entrance into the EOS+ card at the EPMAPEOS location by
09D0          267          ; using SLOTMAP from the EOS mapping function.
09D0          268          ;
09D0          269          .if LABEL
09D0          270 EPMAPEOS   lda #EPONVAL
09D0          271          .el
09D0 A9 00      272          lda #EPONVAL
09D2          273          .fi
09D2          274          ;
09D2 8D 90 C0   275          sta EPSELC+HWSLOT16
09D5          276          ;
09D5          277          ;
09D5          278          ; Initialize X-reg with this slot number and enter MAPEOS.
09D5          279          ;
09D5 A2 01      280          ldx #HWSLOT
09D7          281          ;
09D7 4C B0 10   282          jmp MAPEOS
09DA          283          ;
09DA          284          ;
09DA          285          dfs $E0-*&NEGONE,NEGONE
09E0          286          ;
09E0          287          ;
09E0          288          ; Interface entrance in order to process a BINEOS command
09E0          289          ; that is contained within a BINEOS DCB. The Y-reg and the
09E0          290          ; A-reg must contain the address of the eight byte DCB.
09E0          291          ; The X-reg is initialized with the slot number of this
09E0          292          ; EPROM card before calling BINEOS. BINEOS first calls
09E0          293          ; DOZCOFF and MOVEEPBM before processing the BINEOS DCB.
09E0          294          ; The X-reg must be used in order to turn this EPROM card
09E0          295          ; ON.
09E0          296          ;
09E0          297          .if LABEL
09E0          298 EPBINEOS   ldx #EPONVAL
09E0          299          .el
09E0 A2 00      300          ldx #EPONVAL
09E2          301          .fi
09E2          302          ;
09E2 8E 90 C0   303          stx EPSELC+HWSLOT16
09E5          304          ;

```



```

09E5          305 ;
09E5          306 ; Initialize X-reg with this slot number and then enter
09E5          307 ; BINEOS.
09E5          308 ;
09E5 A2 01     309         ldx #HWSLOT
09E7          310 ;
09E7 4C B6 1B  311         jmp BINEOS
09EA          312 ;
09EA          313 ;
09EA          314         dfs $F0-*)&NEGONE,NEGONE
09F0          315 ;
09F0          316 ;
09F0          317 ; Manual entrance into EOS for this EPROM card.  If this
09F0          318 ; EPROM card is not the highest priority EPROM card,
09F0          319 ; control will pass to the highest priority EPROM card
09F0          320 ; after the EPROM card mapping function has completed.
09F0          321 ; CTRL-N may be used in order to select the desired
09F0          322 ; EPROM card if there are mulitple EPROM cards in an
09F0          323 ; Apple computer.
09F0          324 ;
09F0          325         .if LABEL
09F0          326 EPEOS     lda #EPONVAL
09F0          327         .el
09F0 A9 00     328         lda #EPONVAL
09F2          329         .fi
09F2          330 ;
09F2 8D 90 C0  331         sta EPSELC+HWSLOT16
09F5          332 ;
09F5 4C 00 10  333         jmp EOS
09F8          334 ;
09F8          335 ;
09F8          336 ; This is the ASCII text that is compared to EPTEXT in
09F8          337 ; order to determine if the slot that is being tested
09F8          338 ; contains an EPROM card.
09F8          339 ;
09F8          340         .if LABEL
09F8          341 EPBINTXT asc "EPBINEOS"
09F8          342         .el
09F8 C5 D0 C2  343         asc "EPBINEOS"
09FB C9 CE C5
09FE CF D3
0A00          344         .fi
0A00          345 ;
0A00          346 ;
0A00          347         icl "SLOT2.L"

```

LLOAD SLOT2.L,A\$4000

```

0A00          1          ttl "EOS+ Source Code, SLOT2.L"
0A00          2          ;
0A00          3          ;
0A00          4          ; SLOT2.L
0A00          5          ;
0A00          6          ;
0002          7          HWSLOT    let 2
0020          8          HWSLOT16 let $20
00C2          9          HWSLOT16 let $C2
0A00         10          ;
0000         11          LABEL    let 0
0A00         12          ;
0A00         13          ;
0A00         14          ; This is the generic code that is assembled specifically
0A00         15          ; for each of the seven slots in which an EPROM card may
0A00         16          ; reside.
0A00         17          ;
0A00         18          ; Interface to process an ASEOS command.
0A00         19          ;
0A00         20          .if LABEL
0A00         21          EPASEOS   sta ASAV
0A00         22          .el
0A00 85 18     23          sta ASAV
0A02         24          .fi
0A02         25          ;
0A02 18       26          clc
0A03 90 03    27          bcc >0          ; always taken
0A05         28          ;
0A05         29          ;
0A05         30          ; Insert TESTROM verification code here. When the CXRESET
0A05         31          ; routine is entered, it calls TSTROMCD to test for a ROM
0A05         32          ; card. TSTROMCD calls TESTROM that looks for 0x38xx18 at
0A05         33          ; 0xC305. If found, C3ROMOFF is enabled.
0A05         34          ;
0A05 38       35          sec
0A06 90 00    36          bcc *+2
0A08         37          dfs !-1
0A07 18       38          clc
0A08         39          ;
0A08         40          ;
0A08 86 16    41          ^0          stx XSAV
0A0A 84 17    42          sty YSAV
0A0C         43          ;
0A0C A5 76    44          lda CURLIN+1
0A0E C9 FF    45          cmp #RUNMODE
0A10 D0 03    46          bne >2
0A12         47          ;
0A12 A5 18    48          ^1          lda ASAV          ; recall A-reg
0A14         49          ;
0A14 60       50          rts
0A15         51          ;
0A15 A2 00    52          ^2          ldx #ZERO
0A17         53          ;
0A17 8E A0 02 54          ^3          stx ASPRNUM
0A1A         55          ;
0A1A 20 B7 00 56          jsr CHRGOT
0A1D F0 21    57          beq >4
0A1F         58          ;
0A1F 20 BE DE 59          jsr CHKCOM
0A22         60          ;

```

```

0A22 20 B7 00      61      jsr CHRGOT
0A25 F0 19        62      beq >4
0A27              63      ;
0A27 C9 2C        64      cmp #COMMA&MSBCLR
0A29 F0 15        65      beq >4
0A2B              66      ;
0A2B 20 E3 DF     67      jsr PTRGET
0A2E              68      ;
0A2E AE A0 02     69      ldx ASPRNUM
0A31 E0 0C        70      cpx #MAXASNUM      ; too many parameters
0A33 F0 DD        71      beq <1
0A35              72      ;
0A35 9D D0 02     73      sta ASPRADRS,X
0A38              74      ;
0A38 98           75      tya
0A39 9D D1 02     76      sta ASPRADRS+1,X
0A3C              77      ;
0A3C E8           78      inx
0A3D E8           79      inx
0A3E              80      ;
0A3E D0 D7        81      bne <3      ; always taken
0A40              82      ;
0A40 4E A0 02     83      ^4      lsr ASPRNUM
0A43 F0 CD        84      beq <1      ; no parameters
0A45              85      ;
0A45 A9 00        86      lda #EPONVAL
0A47 8D A0 C0     87      sta EPSELC+HWSLOT16
0A4A              88      ;
0A4A              89      ;
0A4A              90      ; Initialize X-reg with this slot number and enter ASEOS.
0A4A              91      ;
0A4A A2 02        92      ldx #HWSLOT
0A4C              93      ;
0A4C 4C 16 1A     94      jmp ASEOS
0A4F              95      ;
0A4F              96      ;
0A4F              97      dfs $50-*)&NEGONE,NEGONE
0A50              98      ;
0A50              99      ;
0A50            100      ; Exit for ASEOS and to RUN Applesoft files. If a ZipChip
0A50            101      ; is present, flush cache, enable ZipChip, and return to
0A50            102      ; the external caller or address in MEMJMP.
0A50            103      ;
0A50            104      .if LABEL
0A50            105      ASEXIT  lda ZSTATUS
0A50            106      .el
0A50 AD 9A 02     107      lda ZSTATUS
0A53            108      .fi
0A53            109      ;
0A53 30 14        110      bmi >2
0A55            111      ;
0A55 A0 00        112      ldz #ZERO
0A57 A9 60        113      lda /PAGE60
0A59            114      ;
0A59 84 CE        115      sty GENPTR
0A5B 85 CF        116      sta GENPTR+1
0A5D            117      ;
0A5D B1 CE        118      ^1      lda (GENPTR),Y
0A5F            119      ;
0A5F C8           120      iny
0A60 D0 FB        121      bne <1

```

```

0A62          122 ;
0A62 E6 CF    123      inc GENPTR+1
0A64 10 F7    124      bpl <1
0A66          125 ;
0A66 20 E1 1C 126      jsr DOZCON
0A69          127 ;
0A69          128      .if HWCARD
0A69 A9 80    129 ^2    lda #EPOFFVAL
0A6B          130      .el
0A6B          131 ^2    lda #QLOFFVAL
0A6B          132      .fi
0A6B          133 ;
0A6B 8D A0 C0 134      sta EPSELC+HWSLOT16
0A6E          135 ;
0A6E A5 18    136      lda ASAV
0A70 A6 16    137      ldx XSAV
0A72 A4 17    138      ldy YSAV
0A74          139 ;
0A74 6C A6 02 140      jmp (MEMJMP)
0A77          141 ;
0A77          142 ;
0A77          143      dfs $80-*)&NEGONE,NEGONE
0A80          144 ;
0A80          145 ;
0A80          146 ; Exit for BINEOS and to RUN binary files. If a ZipChip is
0A80          147 ; present, flush cache, enable ZipChip, push DOSWARM onto
0A80          148 ; the stack, load the address for EPBINEOS, and return to
0A80          149 ; the external caller or address in MEMJMP.
0A80          150 ;
0A80          151 ; Load the Y-reg/A-reg with the address of EPBINEOS in
0A80          152 ; order for the current file in memory to process more
0A80          153 ; DCBs if other EPROM files need to be loaded into memory.
0A80          154 ;
0A80          155      .if LABEL
0A80          156 BINEXIT  lda ZSTATUS
0A80          157      .el
0A80 AD 9A 02 158      lda ZSTATUS
0A83          159      .fi
0A83          160 ;
0A83 30 14    161      bmi >2
0A85          162 ;
0A85 A0 00    163      ldy #ZERO
0A87 A9 60    164      lda /PAGE60
0A89          165 ;
0A89 84 CE    166      sty GENPTR
0A8B 85 CF    167      sta GENPTR+1
0A8D          168 ;
0A8D B1 CE    169 ^1    lda (GENPTR),Y
0A8F          170 ;
0A8F C8       171      iny
0A90 D0 FB    172      bne <1
0A92          173 ;
0A92 E6 CF    174      inc GENPTR+1
0A94 10 F7    175      bpl <1
0A96          176 ;
0A96 20 E1 1C 177      jsr DOZCON
0A99          178 ;
0A99 A2 FF    179 ^2    ldx #NEGONE
0A9B 9A       180      txs
0A9C          181 ;
0A9C A9 03    182      lda /DOSWARM-1

```

```

0A9E 48          183          pha
0A9F          184          ;
0A9F A9 CF      185          lda #DOSWARM-1
0AA1 48          186          pha
0AA2          187          ;
0AA2 A0 E0      188          ld y #EPBINEOS
0AA4 A9 C2      189          lda #HWSLOT16
0AA6          190          ;
0AA6          191          ;
0AA6 EA         192          nop
0AA7 EA         193          nop
0AA8          194          ;
0AA8          195          ;
0AA8          196          ; Exit from EOS with the assumption that a return to EOS
0AA8          197          ; will be made by means of EPMAPEOS. The X-reg must be
0AA8          198          ; used in order to turn this EPROM card OFF.
0AA8          199          ;
0AA8          200          .if LABEL
0AA8          201          RTNEXIT:
0AA8          202          .fi
0AA8          203          ;
0AA8          204          .if HWCARD
0AA8 A2 80      205          ldx #EPOFFVAL
0AAA          206          .el
0AAA          207          ldx #QLOFFVAL
0AAA          208          .fi
0AAA          209          ;
0AAA 8E A0 C0   210          stx EPSELC+HWSLOT16
0AAD          211          ;
0AAD A2 20      212          ldx #HWSLOT16
0AAF          213          ;
0AAF 6C A6 02   214          jmp (MEMJMP)
0AB2          215          ;
0AB2          216          ;
0AB2          217          dfs $B8-*&NEGONE,NEGONE
0AB8          218          ;
0AB8          219          ;
0AB8          220          ; Special entrance in order to turn this EPROM card OFF.
0AB8          221          ;
0AB8          222          .if LABEL
0AB8          223          EPOFF:
0AB8          224          .fi
0AB8          225          ;
0AB8          226          .if HWCARD
0AB8 A9 80      227          lda #EPOFFVAL
0ABA          228          .el
0ABA          229          lda #QLOFFVAL
0ABA          230          .fi
0ABA          231          ;
0ABA 8D A0 C0   232          sta EPSELC+HWSLOT16
0ABD          233          ;
0ABD 60         234          rts
0ABE          235          ;
0ABE          236          ;
0ABE          237          dfs $C0-*&NEGONE,NEGONE
0AC0          238          ;
0AC0          239          ;
0AC0          240          ; Return from DOS CMDUSER 1 command.
0AC0          241          ;
0AC0          242          .if LABEL
0AC0          243          EPUSER1 lda #EPONVAL

```

```

0AC0          244          .el
0AC0 A9 00     245          lda #EPONVAL
0AC2          246          .fi
0AC2          247          ;
0AC2 8D A0 C0  248          sta EPSELC+HWSLOT16
0AC5          249          ;
0AC5 4C 18 11  250          jmp USERRTN1
0AC8          251          ;
0AC8          252          ;
0AC8          253          ; Return from DOS CMDUSER 2 command.
0AC8          254          ;
0AC8          255          .if LABEL
0AC8          256 EPUSER2  lda #EPONVAL
0AC8          257          .el
0AC8 A9 00     258          lda #EPONVAL
0ACA          259          .fi
0ACA          260          ;
0ACA 8D A0 C0  261          sta EPSELC+HWSLOT16
0ACD          262          ;
0ACD 4C 16 14  263          jmp USERRTN2
0AD0          264          ;
0AD0          265          ;
0AD0          266          ; Entrance into the EOS+ card at the EPMAPEOS location by
0AD0          267          ; using SLOTMAP from the EOS mapping function.
0AD0          268          ;
0AD0          269          .if LABEL
0AD0          270 EPMAPEOS  lda #EPONVAL
0AD0          271          .el
0AD0 A9 00     272          lda #EPONVAL
0AD2          273          .fi
0AD2          274          ;
0AD2 8D A0 C0  275          sta EPSELC+HWSLOT16
0AD5          276          ;
0AD5          277          ;
0AD5          278          ; Initialize X-reg with this slot number and enter MAPEOS.
0AD5          279          ;
0AD5 A2 02     280          ldx #HWSLOT
0AD7          281          ;
0AD7 4C B0 10  282          jmp MAPEOS
0ADA          283          ;
0ADA          284          ;
0ADA          285          dfs $E0-*&NEGONE,NEGONE
0AE0          286          ;
0AE0          287          ;
0AE0          288          ; Interface entrance in order to process a BINEOS command
0AE0          289          ; that is contained within a BINEOS DCB. The Y-reg and the
0AE0          290          ; A-reg must contain the address of the eight byte DCB.
0AE0          291          ; The X-reg is initialized with the slot number of this
0AE0          292          ; EPROM card before calling BINEOS. BINEOS first calls
0AE0          293          ; DOZCOFF and MOVEEPBM before processing the BINEOS DCB.
0AE0          294          ; The X-reg must be used in order to turn this EPROM card
0AE0          295          ; ON.
0AE0          296          ;
0AE0          297          .if LABEL
0AE0          298 EPBINEOS  ldx #EPONVAL
0AE0          299          .el
0AE0 A2 00     300          ldx #EPONVAL
0AE2          301          .fi
0AE2          302          ;
0AE2 8E A0 C0  303          stx EPSELC+HWSLOT16
0AE5          304          ;

```

```

0AE5      305 ;
0AE5      306 ; Initialize X-reg with this slot number and then enter
0AE5      307 ; BINEOS.
0AE5      308 ;
0AE5 A2 02 309      ldx #HWSLOT
0AE7      310 ;
0AE7 4C B6 1B 311      jmp BINEOS
0AEA      312 ;
0AEA      313 ;
0AEA      314      dfs $F0-*)&NEGONE,NEGONE
0AF0      315 ;
0AF0      316 ;
0AF0      317 ; Manual entrance into EOS for this EPROM card.  If this
0AF0      318 ; EPROM card is not the highest priority EPROM card,
0AF0      319 ; control will pass to the highest priority EPROM card
0AF0      320 ; after the EPROM card mapping function has completed.
0AF0      321 ; CTRL-N may be used in order to select the desired
0AF0      322 ; EPROM card if there are mulitple EPROM cards in an
0AF0      323 ; Apple computer.
0AF0      324 ;
0AF0      325      .if LABEL
0AF0      326 EPEOS   lda #EPONVAL
0AF0      327      .el
0AF0 A9 00 328      lda #EPONVAL
0AF2      329      .fi
0AF2      330 ;
0AF2 8D A0 C0 331      sta EPSELC+HWSLOT16
0AF5      332 ;
0AF5 4C 00 10 333      jmp EOS
0AF8      334 ;
0AF8      335 ;
0AF8      336 ; This is the ASCII text that is compared to EPTEXT in
0AF8      337 ; order to determine if the slot that is being tested
0AF8      338 ; contains an EPROM card.
0AF8      339 ;
0AF8      340      .if LABEL
0AF8      341 EPBINTXT asc "EPBINEOS"
0AF8      342      .el
0AF8 C5 D0 C2 343      asc "EPBINEOS"
0AFB C9 CE C5
0AFE CF D3
0B00      344      .fi
0B00      345 ;
0B00      346 ;
0B00      347      icl "SLOT3.L"

```

LLOAD SLOT3.L,A\$4000

```

0B00          1          ttl "EOS+ Source Code, SLOT3.L"
0B00          2          ;
0B00          3          ;
0B00          4          ; SLOT3.L
0B00          5          ;
0B00          6          ;
0003          7          HWSLOT    let 3
0030          8          HWSLOT16  let $30
00C3          9          HWSLOTCX  let $C3
0B00         10          ;
0000         11          LABEL     let 0
0B00         12          ;
0B00         13          ;
0B00         14          ; This is the generic code that is assembled specifically
0B00         15          ; for each of the seven slots in which an EPROM card may
0B00         16          ; reside.
0B00         17          ;
0B00         18          ; Interface to process an ASEOS command.
0B00         19          ;
0B00         20          .if LABEL
0B00         21          EPASEOS   sta ASAV
0B00         22          .el
0B00 85 18     23          sta ASAV
0B02         24          .fi
0B02         25          ;
0B02 18       26          clc
0B03 90 03    27          bcc >0          ; always taken
0B05         28          ;
0B05         29          ;
0B05         30          ; Insert TESTROM verification code here.  When the CXRESET
0B05         31          ; routine is entered, it calls TSTROMCD to test for a ROM
0B05         32          ; card.  TSTROMCD calls TESTROM that looks for 0x38xx18 at
0B05         33          ; 0xC305.  If found, C3ROMOFF is enabled.
0B05         34          ;
0B05 38       35          sec
0B06 90 00    36          bcc *+2
0B08         37          dfs !-1
0B07 18       38          clc
0B08         39          ;
0B08         40          ;
0B08 86 16    41          ^0          stx XSAV
0B0A 84 17    42          sty YSAV
0B0C         43          ;
0B0C A5 76    44          lda CURLIN+1
0B0E C9 FF    45          cmp #RUNMODE
0B10 D0 03    46          bne >2
0B12         47          ;
0B12 A5 18    48          ^1          lda ASAV          ; recall A-reg
0B14         49          ;
0B14 60       50          rts
0B15         51          ;
0B15 A2 00    52          ^2          ldx #ZERO
0B17         53          ;
0B17 8E A0 02 54          ^3          stx ASPRNUM
0B1A         55          ;
0B1A 20 B7 00 56          jsr CHRGOT
0B1D F0 21    57          beq >4
0B1F         58          ;
0B1F 20 BE DE 59          jsr CHKCOM
0B22         60          ;

```



```

0B22 20 B7 00      61      jsr CHRGOT
0B25 F0 19         62      beq >4
0B27              63      ;
0B27 C9 2C         64      cmp #COMMA&MSBCLR
0B29 F0 15         65      beq >4
0B2B              66      ;
0B2B 20 E3 DF      67      jsr PTRGET
0B2E              68      ;
0B2E AE A0 02      69      ldx ASPRNUM
0B31 E0 0C         70      cpx #MAXASNUM      ; too many parameters
0B33 F0 DD         71      beq <1
0B35              72      ;
0B35 9D D0 02      73      sta ASPRADRS,X
0B38              74      ;
0B38 98           75      tya
0B39 9D D1 02      76      sta ASPRADRS+1,X
0B3C              77      ;
0B3C E8           78      inx
0B3D E8           79      inx
0B3E              80      ;
0B3E D0 D7         81      bne <3      ; always taken
0B40              82      ;
0B40 4E A0 02      83      ^4      lsr ASPRNUM
0B43 F0 CD         84      beq <1      ; no parameters
0B45              85      ;
0B45 A9 00         86      lda #EPONVAL
0B47 8D B0 C0      87      sta EPSELC+HWSLOT16
0B4A              88      ;
0B4A              89      ;
0B4A              90      ; Initialize X-reg with this slot number and enter ASEOS.
0B4A              91      ;
0B4A A2 03         92      ldx #HWSLOT
0B4C              93      ;
0B4C 4C 16 1A      94      jmp ASEOS
0B4F              95      ;
0B4F              96      ;
0B4F              97      dfs $50-*)&NEGONE,NEGONE
0B50              98      ;
0B50              99      ;
0B50            100      ; Exit for ASEOS and to RUN Applesoft files. If a ZipChip
0B50            101      ; is present, flush cache, enable ZipChip, and return to
0B50            102      ; the external caller or address in MEMJMP.
0B50            103      ;
0B50            104      .if LABEL
0B50            105      ASEXIT  lda ZSTATUS
0B50            106      .el
0B50 AD 9A 02      107      lda ZSTATUS
0B53            108      .fi
0B53            109      ;
0B53 30 14         110      bmi >2
0B55              111      ;
0B55 A0 00         112      ldy #ZERO
0B57 A9 60         113      lda /PAGE60
0B59              114      ;
0B59 84 CE         115      sty GENPTR
0B5B 85 CF         116      sta GENPTR+1
0B5D              117      ;
0B5D B1 CE         118      ^1      lda (GENPTR),Y
0B5F              119      ;
0B5F C8           120      iny
0B60 D0 FB         121      bne <1

```

```

0B62          122 ;
0B62 E6 CF    123      inc GENPTR+1
0B64 10 F7    124      bpl <1
0B66          125 ;
0B66 20 E1 1C 126      jsr DOZCON
0B69          127 ;
0B69          128      .if HWCARD
0B69 A9 80    129 ^2    lda #EPOFFVAL
0B6B          130      .el
0B6B          131 ^2    lda #QLOFFVAL
0B6B          132      .fi
0B6B          133 ;
0B6B 8D B0 C0 134      sta EPSELC+HWSLOT16
0B6E          135 ;
0B6E A5 18    136      lda ASAV
0B70 A6 16    137      ldx XSAV
0B72 A4 17    138      ldy YSAV
0B74          139 ;
0B74 6C A6 02 140      jmp (MEMJMP)
0B77          141 ;
0B77          142 ;
0B77          143      dfs $80-*)&NEGONE,NEGONE
0B80          144 ;
0B80          145 ;
0B80          146 ; Exit for BINEOS and to RUN binary files. If a ZipChip is
0B80          147 ; present, flush cache, enable ZipChip, push DOSWARM onto
0B80          148 ; the stack, load the address for EPBINEOS, and return to
0B80          149 ; the external caller or address in MEMJMP.
0B80          150 ;
0B80          151 ; Load the Y-reg/A-reg with the address of EPBINEOS in
0B80          152 ; order for the current file in memory to process more
0B80          153 ; DCBs if other EPROM files need to be loaded into memory.
0B80          154 ;
0B80          155      .if LABEL
0B80          156 BINEXIT  lda ZSTATUS
0B80          157      .el
0B80 AD 9A 02 158      lda ZSTATUS
0B83          159      .fi
0B83          160 ;
0B83 30 14    161      bmi >2
0B85          162 ;
0B85 A0 00    163      ldy #ZERO
0B87 A9 60    164      lda /PAGE60
0B89          165 ;
0B89 84 CE    166      sty GENPTR
0B8B 85 CF    167      sta GENPTR+1
0B8D          168 ;
0B8D B1 CE    169 ^1    lda (GENPTR),Y
0B8F          170 ;
0B8F C8       171      iny
0B90 D0 FB    172      bne <1
0B92          173 ;
0B92 E6 CF    174      inc GENPTR+1
0B94 10 F7    175      bpl <1
0B96          176 ;
0B96 20 E1 1C 177      jsr DOZCON
0B99          178 ;
0B99 A2 FF    179 ^2    ldx #NEGONE
0B9B 9A       180      txs
0B9C          181 ;
0B9C A9 03    182      lda /DOSWARM-1

```

```

0B9E 48          183          pha
0B9F          184          ;
0B9F A9 CF      185          lda #DOSWARM-1
0BA1 48          186          pha
0BA2          187          ;
0BA2 A0 E0      188          ldy #EPBINEOS
0BA4 A9 C3      189          lda #HWSLOT16
0BA6          190          ;
0BA6          191          ;
0BA6 EA        192          nop
0BA7 EA        193          nop
0BA8          194          ;
0BA8          195          ;
0BA8          196          ; Exit from EOS with the assumption that a return to EOS
0BA8          197          ; will be made by means of EPMAPEOS. The X-reg must be
0BA8          198          ; used in order to turn this EPROM card OFF.
0BA8          199          ;
0BA8          200          .if LABEL
0BA8          201          RTNEXIT:
0BA8          202          .fi
0BA8          203          ;
0BA8          204          .if HWCARD
0BA8 A2 80      205          ldx #EPOFFVAL
0BAA          206          .el
0BAA          207          ldx #QLOFFVAL
0BAA          208          .fi
0BAA          209          ;
0BAA 8E B0 C0   210          stx EPSELC+HWSLOT16
0BAD          211          ;
0BAD A2 30      212          ldx #HWSLOT16
0BAF          213          ;
0BAF 6C A6 02   214          jmp (MEMJMP)
0BB2          215          ;
0BB2          216          ;
0BB2          217          dfs $B8-*&NEGONE,NEGONE
0BB8          218          ;
0BB8          219          ;
0BB8          220          ; Special entrance in order to turn this EPROM card OFF.
0BB8          221          ;
0BB8          222          .if LABEL
0BB8          223          EPOFF:
0BB8          224          .fi
0BB8          225          ;
0BB8          226          .if HWCARD
0BB8 A9 80      227          lda #EPOFFVAL
0BBA          228          .el
0BBA          229          lda #QLOFFVAL
0BBA          230          .fi
0BBA          231          ;
0BBA 8D B0 C0   232          sta EPSELC+HWSLOT16
0BBD          233          ;
0BBD 60         234          rts
0BBE          235          ;
0BBE          236          ;
0BBE          237          dfs $C0-*&NEGONE,NEGONE
0BC0          238          ;
0BC0          239          ;
0BC0          240          ; Return from DOS CMDUSER 1 command.
0BC0          241          ;
0BC0          242          .if LABEL
0BC0          243          EPUSER1 lda #EPONVAL

```

```

0BC0          244          .el
0BC0 A9 00      245          lda #EPONVAL
0BC2          246          .fi
0BC2          247          ;
0BC2 8D B0 C0   248          sta EPSELC+HWSLOT16
0BC5          249          ;
0BC5 4C 18 11   250          jmp USERRTN1
0BC8          251          ;
0BC8          252          ;
0BC8          253          ; Return from DOS CMDUSER 2 command.
0BC8          254          ;
0BC8          255          .if LABEL
0BC8          256 EPUSER2   lda #EPONVAL
0BC8          257          .el
0BC8 A9 00      258          lda #EPONVAL
0BCA          259          .fi
0BCA          260          ;
0BCA 8D B0 C0   261          sta EPSELC+HWSLOT16
0BCD          262          ;
0BCD 4C 16 14   263          jmp USERRTN2
0BD0          264          ;
0BD0          265          ;
0BD0          266          ; Entrance into the EOS+ card at the EPMAPEOS location by
0BD0          267          ; using SLOTMAP from the EOS mapping function.
0BD0          268          ;
0BD0          269          .if LABEL
0BD0          270 EPMAPEOS   lda #EPONVAL
0BD0          271          .el
0BD0 A9 00      272          lda #EPONVAL
0BD2          273          .fi
0BD2          274          ;
0BD2 8D B0 C0   275          sta EPSELC+HWSLOT16
0BD5          276          ;
0BD5          277          ;
0BD5          278          ; Initialize X-reg with this slot number and enter MAPEOS.
0BD5          279          ;
0BD5 A2 03      280          ldx #HWSLOT
0BD7          281          ;
0BD7 4C B0 10   282          jmp MAPEOS
0BDA          283          ;
0BDA          284          ;
0BDA          285          dfs $E0-*&NEGONE,NEGONE
0BE0          286          ;
0BE0          287          ;
0BE0          288          ; Interface entrance in order to process a BINEOS command
0BE0          289          ; that is contained within a BINEOS DCB. The Y-reg and the
0BE0          290          ; A-reg must contain the address of the eight byte DCB.
0BE0          291          ; The X-reg is initialized with the slot number of this
0BE0          292          ; EPROM card before calling BINEOS. BINEOS first calls
0BE0          293          ; DOZCOFF and MOVEEPBM before processing the BINEOS DCB.
0BE0          294          ; The X-reg must be used in order to turn this EPROM card
0BE0          295          ; ON.
0BE0          296          ;
0BE0          297          .if LABEL
0BE0          298 EPBINEOS   ldx #EPONVAL
0BE0          299          .el
0BE0 A2 00      300          ldx #EPONVAL
0BE2          301          .fi
0BE2          302          ;
0BE2 8E B0 C0   303          stx EPSELC+HWSLOT16
0BE5          304          ;

```

```

0BE5          305 ;
0BE5          306 ; Initialize X-reg with this slot number and then enter
0BE5          307 ; BINEOS.
0BE5          308 ;
0BE5 A2 03    309         ldx #HWSLOT
0BE7          310 ;
0BE7 4C B6 1B 311         jmp BINEOS
0BEA          312 ;
0BEA          313 ;
0BEA          314         dfs $F0-*)&NEGONE,NEGONE
0BF0          315 ;
0BF0          316 ;
0BF0          317 ; Manual entrance into EOS for this EPROM card.  If this
0BF0          318 ; EPROM card is not the highest priority EPROM card,
0BF0          319 ; control will pass to the highest priority EPROM card
0BF0          320 ; after the EPROM card mapping function has completed.
0BF0          321 ; CTRL-N may be used in order to select the desired
0BF0          322 ; EPROM card if there are mulitple EPROM cards in an
0BF0          323 ; Apple computer.
0BF0          324 ;
0BF0          325         .if LABEL
0BF0          326 EPEOS    lda #EPONVAL
0BF0          327         .el
0BF0 A9 00    328         lda #EPONVAL
0BF2          329         .fi
0BF2          330 ;
0BF2 8D B0 C0 331         sta EPSELC+HWSLOT16
0BF5          332 ;
0BF5 4C 00 10 333         jmp EOS
0BF8          334 ;
0BF8          335 ;
0BF8          336 ; This is the ASCII text that is compared to EPTEXT in
0BF8          337 ; order to determine if the slot that is being tested
0BF8          338 ; contains an EPROM card.
0BF8          339 ;
0BF8          340         .if LABEL
0BF8          341 EPBINTXT asc "EPBINEOS"
0BF8          342         .el
0BF8 C5 D0 C2 343         asc "EPBINEOS"
0BFB C9 CE C5
0BFE CF D3
0C00          344         .fi
0C00          345 ;
0C00          346 ;
0C00          347         icl "SLOT4.L"

```

```

LLOAD SLOT4.L,A$4000

```

```

0C00          1          ttl "EOS+ Source Code, SLOT4.L"
0C00          2          ;
0C00          3          ;
0C00          4          ; SLOT4.L
0C00          5          ;
0C00          6          ;
0004          7  HWSLOT   let 4
0040          8  HWSLOT16 let $40
00C4          9  HWSLOT16 let $C4
0C00         10          ;
0001         11  LABEL    let 1
0C00         12          ;
0C00         13          ;
0C00         14          ; This is the generic code that is assembled specifically
0C00         15          ; for each of the seven slots in which an EPROM card may
0C00         16          ; reside.
0C00         17          ;
0C00         18          ; Interface to process an ASEOS command.
0C00         19          ;
0C00         20          .if LABEL
0C00 85 18     21  EPASEOS  sta ASAV
0C02         22          .el
0C02         23          sta ASAV
0C02         24          .fi
0C02         25          ;
0C02 18       26          clc
0C03 90 03    27          bcc >0                ; always taken
0C05         28          ;
0C05         29          ;
0C05         30          ; Insert TESTROM verification code here. When the CXRESET
0C05         31          ; routine is entered, it calls TSTROMCD to test for a ROM
0C05         32          ; card. TSTROMCD calls TESTROM that looks for 0x38xx18 at
0C05         33          ; 0xC305. If found, C3ROMOFF is enabled.
0C05         34          ;
0C05 38       35          sec
0C06 90 00    36          bcc *+2
0C08         37          dfs !-1
0C07 18       38          clc
0C08         39          ;
0C08         40          ;
0C08 86 16    41  ^0      stx XSAV
0C0A 84 17    42          sty YSAV
0C0C         43          ;
0C0C A5 76    44          lda CURLIN+1
0C0E C9 FF    45          cmp #RUNMODE
0C10 D0 03    46          bne >2
0C12         47          ;
0C12 A5 18    48  ^1      lda ASAV                ; recall A-reg
0C14         49          ;
0C14 60       50          rts
0C15         51          ;
0C15 A2 00    52  ^2      ldx #ZERO
0C17         53          ;
0C17 8E A0 02 54  ^3      stx ASPRNUM
0C1A         55          ;
0C1A 20 B7 00 56          jsr CHRGOT
0C1D F0 21    57          beq >4
0C1F         58          ;
0C1F 20 BE DE 59          jsr CHKCOM
0C22         60          ;

```

```

0C22 20 B7 00      61      jsr CHRGOT
0C25 F0 19         62      beq >4
0C27              63      ;
0C27 C9 2C         64      cmp #COMMA&MSBCLR
0C29 F0 15         65      beq >4
0C2B              66      ;
0C2B 20 E3 DF      67      jsr PTRGET
0C2E              68      ;
0C2E AE A0 02      69      ldx ASPRNUM
0C31 E0 0C         70      cpx #MAXASNUM      ; too many parameters
0C33 F0 DD         71      beq <1
0C35              72      ;
0C35 9D D0 02      73      sta ASPRADRS,X
0C38              74      ;
0C38 98           75      tya
0C39 9D D1 02      76      sta ASPRADRS+1,X
0C3C              77      ;
0C3C E8           78      inx
0C3D E8           79      inx
0C3E              80      ;
0C3E D0 D7         81      bne <3      ; always taken
0C40              82      ;
0C40 4E A0 02      83      ^4      lsr ASPRNUM
0C43 F0 CD         84      beq <1      ; no parameters
0C45              85      ;
0C45 A9 00         86      lda #EPONVAL
0C47 8D C0 C0      87      sta EPSELC+HWSLOT16
0C4A              88      ;
0C4A              89      ;
0C4A              90      ; Initialize X-reg with this slot number and enter ASEOS.
0C4A              91      ;
0C4A A2 04         92      ldx #HWSLOT
0C4C              93      ;
0C4C 4C 16 1A      94      jmp ASEOS
0C4F              95      ;
0C4F              96      ;
0C4F              97      dfs $50-*)&NEGONE,NEGONE
0C50              98      ;
0C50              99      ;
0C50             100      ; Exit for ASEOS and to RUN Applesoft files. If a ZipChip
0C50             101      ; is present, flush cache, enable ZipChip, and return to
0C50             102      ; the external caller or address in MEMJMP.
0C50             103      ;
0C50             104      .if LABEL
0C50 AD 9A 02      105      ASEXIT  lda ZSTATUS
0C53             106      .el
0C53             107      lda ZSTATUS
0C53             108      .fi
0C53             109      ;
0C53 30 14         110      bmi >2
0C55             111      ;
0C55 A0 00         112      ldy #ZERO
0C57 A9 60         113      lda /PAGE60
0C59             114      ;
0C59 84 CE         115      sty GENPTR
0C5B 85 CF         116      sta GENPTR+1
0C5D             117      ;
0C5D B1 CE         118      ^1      lda (GENPTR),Y
0C5F             119      ;
0C5F C8           120      iny
0C60 D0 FB         121      bne <1

```

```

0C62          122 ;
0C62 E6 CF    123      inc GENPTR+1
0C64 10 F7    124      bpl <1
0C66          125 ;
0C66 20 E1 1C 126      jsr DOZCON
0C69          127 ;
0C69          128      .if HWCARD
0C69 A9 80    129 ^2    lda #EPOFFVAL
0C6B          130      .el
0C6B          131 ^2    lda #QLOFFVAL
0C6B          132      .fi
0C6B          133 ;
0C6B 8D C0 C0 134      sta EPSELC+HWSLOT16
0C6E          135 ;
0C6E A5 18    136      lda ASAV
0C70 A6 16    137      ldx XSAV
0C72 A4 17    138      ldy YSAV
0C74          139 ;
0C74 6C A6 02 140      jmp (MEMJMP)
0C77          141 ;
0C77          142 ;
0C77          143      dfs $80-*)&NEGONE,NEGONE
0C80          144 ;
0C80          145 ;
0C80          146 ; Exit for BINEOS and to RUN binary files. If a ZipChip is
0C80          147 ; present, flush cache, enable ZipChip, push DOSWARM onto
0C80          148 ; the stack, load the address for EPBINEOS, and return to
0C80          149 ; the external caller or address in MEMJMP.
0C80          150 ;
0C80          151 ; Load the Y-reg/A-reg with the address of EPBINEOS in
0C80          152 ; order for the current file in memory to process more
0C80          153 ; DCBs if other EPROM files need to be loaded into memory.
0C80          154 ;
0C80          155      .if LABEL
0C80 AD 9A 02 156 BINEXIT lda ZSTATUS
0C83          157      .el
0C83          158      lda ZSTATUS
0C83          159      .fi
0C83          160 ;
0C83 30 14    161      bmi >2
0C85          162 ;
0C85 A0 00    163      ldy #ZERO
0C87 A9 60    164      lda /PAGE60
0C89          165 ;
0C89 84 CE    166      sty GENPTR
0C8B 85 CF    167      sta GENPTR+1
0C8D          168 ;
0C8D B1 CE    169 ^1    lda (GENPTR),Y
0C8F          170 ;
0C8F C8       171      iny
0C90 D0 FB    172      bne <1
0C92          173 ;
0C92 E6 CF    174      inc GENPTR+1
0C94 10 F7    175      bpl <1
0C96          176 ;
0C96 20 E1 1C 177      jsr DOZCON
0C99          178 ;
0C99 A2 FF    179 ^2    ldx #NEGONE
0C9B 9A       180      txs
0C9C          181 ;
0C9C A9 03    182      lda /DOSWARM-1

```



```

0C9E 48          183          pha
0C9F          184          ;
0C9F A9 CF      185          lda #DOSWARM-1
0CA1 48          186          pha
0CA2          187          ;
0CA2 A0 E0      188          ld y #EPBINEOS
0CA4 A9 C4      189          lda #HWSLOT16
0CA6          190          ;
0CA6          191          ;
0CA6 EA         192          nop
0CA7 EA         193          nop
0CA8          194          ;
0CA8          195          ;
0CA8          196          ; Exit from EOS with the assumption that a return to EOS
0CA8          197          ; will be made by means of EPMAPEOS. The X-reg must be
0CA8          198          ; used in order to turn this EPROM card OFF.
0CA8          199          ;
0CA8          200          .if LABEL
0CA8          201          RTNEXIT:
0CA8          202          .fi
0CA8          203          ;
0CA8          204          .if HWCARD
0CA8 A2 80      205          ldx #EPOFFVAL
0CAA          206          .el
0CAA          207          ldx #QLOFFVAL
0CAA          208          .fi
0CAA          209          ;
0CAA 8E C0 C0   210          stx EPSELC+HWSLOT16
0CAD          211          ;
0CAD A2 40      212          ldx #HWSLOT16
0CAF          213          ;
0CAF 6C A6 02   214          jmp (MEMJMP)
0CB2          215          ;
0CB2          216          ;
0CB2          217          dfs $B8-*&NEGONE,NEGONE
0CB8          218          ;
0CB8          219          ;
0CB8          220          ; Special entrance in order to turn this EPROM card OFF.
0CB8          221          ;
0CB8          222          .if LABEL
0CB8          223          EPOFF:
0CB8          224          .fi
0CB8          225          ;
0CB8          226          .if HWCARD
0CB8 A9 80      227          lda #EPOFFVAL
0CBA          228          .el
0CBA          229          lda #QLOFFVAL
0CBA          230          .fi
0CBA          231          ;
0CBA 8D C0 C0   232          sta EPSELC+HWSLOT16
0CBD          233          ;
0CBD 60         234          rts
0CBE          235          ;
0CBE          236          ;
0CBE          237          dfs $C0-*&NEGONE,NEGONE
0CC0          238          ;
0CC0          239          ;
0CC0          240          ; Return from DOS CMDUSER 1 command.
0CC0          241          ;
0CC0          242          .if LABEL
0CC0 A9 00      243          EPUSER1 lda #EPONVAL

```

```

0CC2      244      .el
0CC2      245      lda #EPONVAL
0CC2      246      .fi
0CC2      247      ;
0CC2 8D C0 C0 248      sta EPSELC+HWSLOT16
0CC5      249      ;
0CC5 4C 18 11 250      jmp USERRTN1
0CC8      251      ;
0CC8      252      ;
0CC8      253      ; Return from DOS CMDUSER 2 command.
0CC8      254      ;
0CC8      255      .if LABEL
0CC8 A9 00 256  EPUSER2  lda #EPONVAL
0CCA      257      .el
0CCA      258      lda #EPONVAL
0CCA      259      .fi
0CCA      260      ;
0CCA 8D C0 C0 261      sta EPSELC+HWSLOT16
0CCD      262      ;
0CCD 4C 16 14 263      jmp USERRTN2
0CD0      264      ;
0CD0      265      ;
0CD0      266      ; Entrance into the EOS+ card at the EPMAPEOS location by
0CD0      267      ; using SLOTMAP from the EOS mapping function.
0CD0      268      ;
0CD0      269      .if LABEL
0CD0 A9 00 270  EPMAPEOS  lda #EPONVAL
0CD2      271      .el
0CD2      272      lda #EPONVAL
0CD2      273      .fi
0CD2      274      ;
0CD2 8D C0 C0 275      sta EPSELC+HWSLOT16
0CD5      276      ;
0CD5      277      ;
0CD5      278      ; Initialize X-reg with this slot number and enter MAPEOS.
0CD5      279      ;
0CD5 A2 04 280      ldx #HWSLOT
0CD7      281      ;
0CD7 4C B0 10 282      jmp MAPEOS
0CDA      283      ;
0CDA      284      ;
0CDA      285      dfs $E0-*&NEGONE,NEGONE
0CE0      286      ;
0CE0      287      ;
0CE0      288      ; Interface entrance in order to process a BINEOS command
0CE0      289      ; that is contained within a BINEOS DCB. The Y-reg and the
0CE0      290      ; A-reg must contain the address of the eight byte DCB.
0CE0      291      ; The X-reg is initialized with the slot number of this
0CE0      292      ; EPROM card before calling BINEOS. BINEOS first calls
0CE0      293      ; DOZCOFF and MOVEEPBM before processing the BINEOS DCB.
0CE0      294      ; The X-reg must be used in order to turn this EPROM card
0CE0      295      ; ON.
0CE0      296      ;
0CE0      297      .if LABEL
0CE0 A2 00 298  EPBINEOS  ldx #EPONVAL
0CE2      299      .el
0CE2      300      ldx #EPONVAL
0CE2      301      .fi
0CE2      302      ;
0CE2 8E C0 C0 303      stx EPSELC+HWSLOT16
0CE5      304      ;

```

```

0CE5      305 ;
0CE5      306 ; Initialize X-reg with this slot number and then enter
0CE5      307 ; BINEOS.
0CE5      308 ;
0CE5 A2 04 309      ldx #HWSLOT
0CE7      310 ;
0CE7 4C B6 1B 311      jmp BINEOS
0CEA      312 ;
0CEA      313 ;
0CEA      314      dfs $F0-*)&NEGONE,NEGONE
0CF0      315 ;
0CF0      316 ;
0CF0      317 ; Manual entrance into EOS for this EPROM card.  If this
0CF0      318 ; EPROM card is not the highest priority EPROM card,
0CF0      319 ; control will pass to the highest priority EPROM card
0CF0      320 ; after the EPROM card mapping function has completed.
0CF0      321 ; CTRL-N may be used in order to select the desired
0CF0      322 ; EPROM card if there are mulitple EPROM cards in an
0CF0      323 ; Apple computer.
0CF0      324 ;
0CF0      325      .if LABEL
0CF0 A9 00 326 EPEOS   lda #EPONVAL
0CF2      327      .el
0CF2      328      lda #EPONVAL
0CF2      329      .fi
0CF2      330 ;
0CF2 8D C0 C0 331      sta EPSELC+HWSLOT16
0CF5      332 ;
0CF5 4C 00 10 333      jmp EOS
0CF8      334 ;
0CF8      335 ;
0CF8      336 ; This is the ASCII text that is compared to EPTEXT in
0CF8      337 ; order to determine if the slot that is being tested
0CF8      338 ; contains an EPROM card.
0CF8      339 ;
0CF8      340      .if LABEL
0CF8 C5 D0 C2 341 EPBINTXT asc "EPBINEOS"
0CFB C9 CE C5
0CFE CF D3
0D00      342      .el
0D00      343      asc "EPBINEOS"
0D00      344      .fi
0D00      345 ;
0D00      346 ;
0D00      347      icl "SLOT5.L"

```

LLOAD SLOT5.L,A\$4000

```

0D00          1          ttl "EOS+ Source Code, SLOT5.L"
0D00          2          ;
0D00          3          ;
0D00          4          ; SLOT5.L
0D00          5          ;
0D00          6          ;
0005          7  HWSLOT    let 5
0050          8  HWSLOT16 let $50
00C5          9  HWSLOT16 let $C5
0D00         10          ;
0000         11  LABEL     let 0
0D00         12          ;
0D00         13          ;
0D00         14          ; This is the generic code that is assembled specifically
0D00         15          ; for each of the seven slots in which an EPROM card may
0D00         16          ; reside.
0D00         17          ;
0D00         18          ; Interface to process an ASEOS command.
0D00         19          ;
0D00         20          .if LABEL
0D00         21  EPASEOS   sta ASAV
0D00         22          .el
0D00 85 18      23          sta ASAV
0D02         24          .fi
0D02         25          ;
0D02 18        26          clc
0D03 90 03     27          bcc >0                ; always taken
0D05         28          ;
0D05         29          ;
0D05         30          ; Insert TESTROM verification code here. When the CXRESET
0D05         31          ; routine is entered, it calls TSTROMCD to test for a ROM
0D05         32          ; card. TSTROMCD calls TESTROM that looks for 0x38xx18 at
0D05         33          ; 0xC305. If found, C3ROMOFF is enabled.
0D05         34          ;
0D05 38        35          sec
0D06 90 00     36          bcc *+2
0D08         37          dfs !-1
0D07 18        38          clc
0D08         39          ;
0D08         40          ;
0D08 86 16     41  ^0      stx XSAV
0D0A 84 17     42          sty YSAV
0D0C         43          ;
0D0C A5 76     44          lda CURLIN+1
0D0E C9 FF     45          cmp #RUNMODE
0D10 D0 03     46          bne >2
0D12         47          ;
0D12 A5 18     48  ^1      lda ASAV                ; recall A-reg
0D14         49          ;
0D14 60        50          rts
0D15         51          ;
0D15 A2 00     52  ^2      ldx #ZERO
0D17         53          ;
0D17 8E A0 02  54  ^3      stx ASPRNUM
0D1A         55          ;
0D1A 20 B7 00  56          jsr CHRGOT
0D1D F0 21     57          beq >4
0D1F         58          ;
0D1F 20 BE DE  59          jsr CHKCOM
0D22         60          ;

```

```

0D22 20 B7 00      61      jsr CHRGOT
0D25 F0 19         62      beq >4
0D27              63      ;
0D27 C9 2C         64      cmp #COMMA&MSBCLR
0D29 F0 15         65      beq >4
0D2B              66      ;
0D2B 20 E3 DF      67      jsr PTRGET
0D2E              68      ;
0D2E AE A0 02      69      ldx ASPRNUM
0D31 E0 0C         70      cpx #MAXASNUM      ; too many parameters
0D33 F0 DD         71      beq <1
0D35              72      ;
0D35 9D D0 02      73      sta ASPRADRS,X
0D38              74      ;
0D38 98           75      tya
0D39 9D D1 02      76      sta ASPRADRS+1,X
0D3C              77      ;
0D3C E8           78      inx
0D3D E8           79      inx
0D3E              80      ;
0D3E D0 D7         81      bne <3      ; always taken
0D40              82      ;
0D40 4E A0 02      83      ^4      lsr ASPRNUM
0D43 F0 CD         84      beq <1      ; no parameters
0D45              85      ;
0D45 A9 00         86      lda #EPONVAL
0D47 8D D0 C0      87      sta EPSELC+HWSLOT16
0D4A              88      ;
0D4A              89      ;
0D4A              90      ; Initialize X-reg with this slot number and enter ASEOS.
0D4A              91      ;
0D4A A2 05         92      ldx #HWSLOT
0D4C              93      ;
0D4C 4C 16 1A      94      jmp ASEOS
0D4F              95      ;
0D4F              96      ;
0D4F              97      dfs $50-*)&NEGONE,NEGONE
0D50              98      ;
0D50              99      ;
0D50             100      ; Exit for ASEOS and to RUN Applesoft files. If a ZipChip
0D50             101      ; is present, flush cache, enable ZipChip, and return to
0D50             102      ; the external caller or address in MEMJMP.
0D50             103      ;
0D50             104      .if LABEL
0D50             105      ASEXIT  lda ZSTATUS
0D50             106      .el
0D50 AD 9A 02      107      lda ZSTATUS
0D53             108      .fi
0D53             109      ;
0D53 30 14         110      bmi >2
0D55             111      ;
0D55 A0 00         112      ldy #ZERO
0D57 A9 60         113      lda /PAGE60
0D59             114      ;
0D59 84 CE         115      sty GENPTR
0D5B 85 CF         116      sta GENPTR+1
0D5D             117      ;
0D5D B1 CE         118      ^1      lda (GENPTR),Y
0D5F             119      ;
0D5F C8           120      iny
0D60 D0 FB         121      bne <1

```

```

0D62          122 ;
0D62 E6 CF    123      inc GENPTR+1
0D64 10 F7    124      bpl <1
0D66          125 ;
0D66 20 E1 1C 126      jsr DOZCON
0D69          127 ;
0D69          128      .if HWCARD
0D69 A9 80    129 ^2    lda #EPOFFVAL
0D6B          130      .el
0D6B          131 ^2    lda #QLOFFVAL
0D6B          132      .fi
0D6B          133 ;
0D6B 8D D0 C0 134      sta EPSELC+HWSLOT16
0D6E          135 ;
0D6E A5 18    136      lda ASAV
0D70 A6 16    137      ldx XSAV
0D72 A4 17    138      ldy YSAV
0D74          139 ;
0D74 6C A6 02 140      jmp (MEMJMP)
0D77          141 ;
0D77          142 ;
0D77          143      dfs $80-*)&NEGONE,NEGONE
0D80          144 ;
0D80          145 ;
0D80          146 ; Exit for BINEOS and to RUN binary files. If a ZipChip is
0D80          147 ; present, flush cache, enable ZipChip, push DOSWARM onto
0D80          148 ; the stack, load the address for EPBINEOS, and return to
0D80          149 ; the external caller or address in MEMJMP.
0D80          150 ;
0D80          151 ; Load the Y-reg/A-reg with the address of EPBINEOS in
0D80          152 ; order for the current file in memory to process more
0D80          153 ; DCBs if other EPROM files need to be loaded into memory.
0D80          154 ;
0D80          155      .if LABEL
0D80          156 BINEXIT  lda ZSTATUS
0D80          157      .el
0D80 AD 9A 02 158      lda ZSTATUS
0D83          159      .fi
0D83          160 ;
0D83 30 14    161      bmi >2
0D85          162 ;
0D85 A0 00    163      ldy #ZERO
0D87 A9 60    164      lda /PAGE60
0D89          165 ;
0D89 84 CE    166      sty GENPTR
0D8B 85 CF    167      sta GENPTR+1
0D8D          168 ;
0D8D B1 CE    169 ^1    lda (GENPTR),Y
0D8F          170 ;
0D8F C8       171      iny
0D90 D0 FB    172      bne <1
0D92          173 ;
0D92 E6 CF    174      inc GENPTR+1
0D94 10 F7    175      bpl <1
0D96          176 ;
0D96 20 E1 1C 177      jsr DOZCON
0D99          178 ;
0D99 A2 FF    179 ^2    ldx #NEGONE
0D9B 9A       180      txs
0D9C          181 ;
0D9C A9 03    182      lda /DOSWARM-1

```

```

0D9E 48          183          pha
0D9F          184          ;
0D9F A9 CF      185          lda #DOSWARM-1
0DA1 48          186          pha
0DA2          187          ;
0DA2 A0 E0      188          ld y #EPBINEOS
0DA4 A9 C5      189          lda #HWSLOT16
0DA6          190          ;
0DA6          191          ;
0DA6 EA         192          nop
0DA7 EA         193          nop
0DA8          194          ;
0DA8          195          ;
0DA8          196          ; Exit from EOS with the assumption that a return to EOS
0DA8          197          ; will be made by means of EPMAPEOS. The X-reg must be
0DA8          198          ; used in order to turn this EPROM card OFF.
0DA8          199          ;
0DA8          200          .if LABEL
0DA8          201          RTNEXIT:
0DA8          202          .fi
0DA8          203          ;
0DA8          204          .if HWCARD
0DA8 A2 80      205          ldx #EPOFFVAL
0DAA          206          .el
0DAA          207          ldx #QLOFFVAL
0DAA          208          .fi
0DAA          209          ;
0DAA 8E D0 C0   210          stx EPSELC+HWSLOT16
0DAD          211          ;
0DAD A2 50      212          ldx #HWSLOT16
0DAF          213          ;
0DAF 6C A6 02   214          jmp (MEMJMP)
0DB2          215          ;
0DB2          216          ;
0DB2          217          dfs $B8-*&NEGONE,NEGONE
0DB8          218          ;
0DB8          219          ;
0DB8          220          ; Special entrance in order to turn this EPROM card OFF.
0DB8          221          ;
0DB8          222          .if LABEL
0DB8          223          EPOFF:
0DB8          224          .fi
0DB8          225          ;
0DB8          226          .if HWCARD
0DB8 A9 80      227          lda #EPOFFVAL
0DBA          228          .el
0DBA          229          lda #QLOFFVAL
0DBA          230          .fi
0DBA          231          ;
0DBA 8D D0 C0   232          sta EPSELC+HWSLOT16
0DBD          233          ;
0DBD 60         234          rts
0DBE          235          ;
0DBE          236          ;
0DBE          237          dfs $C0-*&NEGONE,NEGONE
0DC0          238          ;
0DC0          239          ;
0DC0          240          ; Return from DOS CMDUSER 1 command.
0DC0          241          ;
0DC0          242          .if LABEL
0DC0          243          EPUSER1 lda #EPONVAL

```

```

0DC0          244          .el
0DC0 A9 00      245          lda #EPONVAL
0DC2          246          .fi
0DC2          247          ;
0DC2 8D D0 C0   248          sta EPSELC+HWSLOT16
0DC5          249          ;
0DC5 4C 18 11   250          jmp USERRTN1
0DC8          251          ;
0DC8          252          ;
0DC8          253          ; Return from DOS CMDUSER 2 command.
0DC8          254          ;
0DC8          255          .if LABEL
0DC8          256 EPUSER2   lda #EPONVAL
0DC8          257          .el
0DC8 A9 00      258          lda #EPONVAL
0DCA          259          .fi
0DCA          260          ;
0DCA 8D D0 C0   261          sta EPSELC+HWSLOT16
0DCD          262          ;
0DCD 4C 16 14   263          jmp USERRTN2
0DD0          264          ;
0DD0          265          ;
0DD0          266          ; Entrance into the EOS+ card at the EPMAPEOS location by
0DD0          267          ; using SLOTMAP from the EOS mapping function.
0DD0          268          ;
0DD0          269          .if LABEL
0DD0          270 EPMAPEOS   lda #EPONVAL
0DD0          271          .el
0DD0 A9 00      272          lda #EPONVAL
0DD2          273          .fi
0DD2          274          ;
0DD2 8D D0 C0   275          sta EPSELC+HWSLOT16
0DD5          276          ;
0DD5          277          ;
0DD5          278          ; Initialize X-reg with this slot number and enter MAPEOS.
0DD5          279          ;
0DD5 A2 05      280          ldx #HWSLOT
0DD7          281          ;
0DD7 4C B0 10   282          jmp MAPEOS
0DDA          283          ;
0DDA          284          ;
0DDA          285          dfs $E0-*&NEGONE,NEGONE
0DE0          286          ;
0DE0          287          ;
0DE0          288          ; Interface entrance in order to process a BINEOS command
0DE0          289          ; that is contained within a BINEOS DCB. The Y-reg and the
0DE0          290          ; A-reg must contain the address of the eight byte DCB.
0DE0          291          ; The X-reg is initialized with the slot number of this
0DE0          292          ; EPROM card before calling BINEOS. BINEOS first calls
0DE0          293          ; DOZCOFF and MOVEEPBM before processing the BINEOS DCB.
0DE0          294          ; The X-reg must be used in order to turn this EPROM card
0DE0          295          ; ON.
0DE0          296          ;
0DE0          297          .if LABEL
0DE0          298 EPBINEOS   ldx #EPONVAL
0DE0          299          .el
0DE0 A2 00      300          ldx #EPONVAL
0DE2          301          .fi
0DE2          302          ;
0DE2 8E D0 C0   303          stx EPSELC+HWSLOT16
0DE5          304          ;

```



```

0DE5      305 ;
0DE5      306 ; Initialize X-reg with this slot number and then enter
0DE5      307 ; BINEOS.
0DE5      308 ;
0DE5 A2 05      309      ldx #HWSLOT
0DE7      310 ;
0DE7 4C B6 1B   311      jmp BINEOS
0DEA      312 ;
0DEA      313 ;
0DEA      314      dfs $F0-*&NEGONE,NEGONE
0DF0      315 ;
0DF0      316 ;
0DF0      317 ; Manual entrance into EOS for this EPROM card.  If this
0DF0      318 ; EPROM card is not the highest priority EPROM card,
0DF0      319 ; control will pass to the highest priority EPROM card
0DF0      320 ; after the EPROM card mapping function has completed.
0DF0      321 ; CTRL-N may be used in order to select the desired
0DF0      322 ; EPROM card if there are mulitple EPROM cards in an
0DF0      323 ; Apple computer.
0DF0      324 ;
0DF0      325      .if LABEL
0DF0      326 EPEOS      lda #EPONVAL
0DF0      327      .el
0DF0 A9 00      328      lda #EPONVAL
0DF2      329      .fi
0DF2      330 ;
0DF2 8D D0 C0   331      sta EPSELC+HWSLOT16
0DF5      332 ;
0DF5 4C 00 10   333      jmp EOS
0DF8      334 ;
0DF8      335 ;
0DF8      336 ; This is the ASCII text that is compared to EPTEXT in
0DF8      337 ; order to determine if the slot that is being tested
0DF8      338 ; contains an EPROM card.
0DF8      339 ;
0DF8      340      .if LABEL
0DF8      341 EPBINTXT asc "EPBINEOS"
0DF8      342      .el
0DF8 C5 D0 C2   343      asc "EPBINEOS"
0DFB C9 CE C5
0DFE CF D3
0E00      344      .fi
0E00      345 ;
0E00      346 ;
0E00      347      icl "SLOT6.L"

```

```

LLOAD SLOT6.L,A$4000

```

```

0E00          1          ttl "EOS+ Source Code, SLOT6.L"
0E00          2          ;
0E00          3          ;
0E00          4          ; SLOT6.L
0E00          5          ;
0E00          6          ;
0006          7  HWSLOT    let 6
0060          8  HWSLOT16  let $60
00C6          9  HWSLOTCX  let $C6
0E00         10          ;
0000         11  LABEL     let 0
0E00         12          ;
0E00         13          ;
0E00         14          ; This is the generic code that is assembled specifically
0E00         15          ; for each of the seven slots in which an EPROM card may
0E00         16          ; reside.
0E00         17          ;
0E00         18          ; Interface to process an ASEOS command.
0E00         19          ;
0E00         20          .if LABEL
0E00         21  EPASEOS   sta ASAV
0E00         22          .el
0E00 85 18      23          sta ASAV
0E02         24          .fi
0E02         25          ;
0E02 18        26          clc
0E03 90 03      27          bcc >0                      ; always taken
0E05         28          ;
0E05         29          ;
0E05         30          ; Insert TESTROM verification code here.  When the CXRESET
0E05         31          ; routine is entered, it calls TSTROMCD to test for a ROM
0E05         32          ; card.  TSTROMCD calls TESTROM that looks for 0x38xx18 at
0E05         33          ; 0xC305.  If found, C3ROMOFF is enabled.
0E05         34          ;
0E05 38        35          sec
0E06 90 00      36          bcc *+2
0E08         37          dfs !-1
0E07 18        38          clc
0E08         39          ;
0E08         40          ;
0E08 86 16      41  ^0      stx XSAV
0E0A 84 17      42          sty YSAV
0E0C         43          ;
0E0C A5 76      44          lda CURLIN+1
0E0E C9 FF      45          cmp #RUNMODE
0E10 D0 03      46          bne >2
0E12         47          ;
0E12 A5 18      48  ^1      lda ASAV                      ; recall A-reg
0E14         49          ;
0E14 60        50          rts
0E15         51          ;
0E15 A2 00      52  ^2      ldx #ZERO
0E17         53          ;
0E17 8E A0 02   54  ^3      stx ASPRNUM
0E1A         55          ;
0E1A 20 B7 00   56          jsr CHRGOT
0E1D F0 21      57          beq >4
0E1F         58          ;
0E1F 20 BE DE   59          jsr CHKCOM
0E22         60          ;

```

```

0E22 20 B7 00      61      jsr CHRGOT
0E25 F0 19        62      beq >4
0E27              63      ;
0E27 C9 2C        64      cmp #COMMA&MSBCLR
0E29 F0 15        65      beq >4
0E2B              66      ;
0E2B 20 E3 DF     67      jsr PTRGET
0E2E              68      ;
0E2E AE A0 02     69      ldx ASPRNUM
0E31 E0 0C        70      cpx #MAXASNUM      ; too many parameters
0E33 F0 DD        71      beq <1
0E35              72      ;
0E35 9D D0 02     73      sta ASPRADRS,X
0E38              74      ;
0E38 98          75      tya
0E39 9D D1 02     76      sta ASPRADRS+1,X
0E3C              77      ;
0E3C E8          78      inx
0E3D E8          79      inx
0E3E              80      ;
0E3E D0 D7       81      bne <3      ; always taken
0E40              82      ;
0E40 4E A0 02     83      ^4      lsr ASPRNUM
0E43 F0 CD        84      beq <1      ; no parameters
0E45              85      ;
0E45 A9 00        86      lda #EPONVAL
0E47 8D E0 C0     87      sta EPSELC+HWSLOT16
0E4A              88      ;
0E4A              89      ;
0E4A              90      ; Initialize X-reg with this slot number and enter ASEOS.
0E4A              91      ;
0E4A A2 06        92      ldx #HWSLOT
0E4C              93      ;
0E4C 4C 16 1A     94      jmp ASEOS
0E4F              95      ;
0E4F              96      ;
0E4F              97      dfs $50-*)&NEGONE,NEGONE
0E50              98      ;
0E50              99      ;
0E50            100      ; Exit for ASEOS and to RUN Applesoft files. If a ZipChip
0E50            101      ; is present, flush cache, enable ZipChip, and return to
0E50            102      ; the external caller or address in MEMJMP.
0E50            103      ;
0E50            104      .if LABEL
0E50            105      ASEXIT  lda ZSTATUS
0E50            106      .el
0E50 AD 9A 02     107      lda ZSTATUS
0E53            108      .fi
0E53            109      ;
0E53 30 14       110      bmi >2
0E55            111      ;
0E55 A0 00       112      ldy #ZERO
0E57 A9 60       113      lda /PAGE60
0E59            114      ;
0E59 84 CE       115      sty GENPTR
0E5B 85 CF       116      sta GENPTR+1
0E5D            117      ;
0E5D B1 CE       118      ^1      lda (GENPTR),Y
0E5F            119      ;
0E5F C8          120      iny
0E60 D0 FB       121      bne <1

```

```

0E62          122 ;
0E62 E6 CF    123      inc GENPTR+1
0E64 10 F7    124      bpl <1
0E66          125 ;
0E66 20 E1 1C 126      jsr DOZCON
0E69          127 ;
0E69          128      .if HWCARD
0E69 A9 80    129 ^2    lda #EPOFFVAL
0E6B          130      .el
0E6B          131 ^2    lda #QLOFFVAL
0E6B          132      .fi
0E6B          133 ;
0E6B 8D E0 C0 134      sta EPSELC+HWSLOT16
0E6E          135 ;
0E6E A5 18    136      lda ASAV
0E70 A6 16    137      ldx XSAV
0E72 A4 17    138      ldy YSAV
0E74          139 ;
0E74 6C A6 02 140      jmp (MEMJMP)
0E77          141 ;
0E77          142 ;
0E77          143      dfs $80-*)&NEGONE,NEGONE
0E80          144 ;
0E80          145 ;
0E80          146 ; Exit for BINEOS and to RUN binary files. If a ZipChip is
0E80          147 ; present, flush cache, enable ZipChip, push DOSWARM onto
0E80          148 ; the stack, load the address for EPBINEOS, and return to
0E80          149 ; the external caller or address in MEMJMP.
0E80          150 ;
0E80          151 ; Load the Y-reg/A-reg with the address of EPBINEOS in
0E80          152 ; order for the current file in memory to process more
0E80          153 ; DCBs if other EPROM files need to be loaded into memory.
0E80          154 ;
0E80          155      .if LABEL
0E80          156 BINEXIT  lda ZSTATUS
0E80          157      .el
0E80 AD 9A 02 158      lda ZSTATUS
0E83          159      .fi
0E83          160 ;
0E83 30 14    161      bmi >2
0E85          162 ;
0E85 A0 00    163      ldy #ZERO
0E87 A9 60    164      lda /PAGE60
0E89          165 ;
0E89 84 CE    166      sty GENPTR
0E8B 85 CF    167      sta GENPTR+1
0E8D          168 ;
0E8D B1 CE    169 ^1    lda (GENPTR),Y
0E8F          170 ;
0E8F C8       171      iny
0E90 D0 FB    172      bne <1
0E92          173 ;
0E92 E6 CF    174      inc GENPTR+1
0E94 10 F7    175      bpl <1
0E96          176 ;
0E96 20 E1 1C 177      jsr DOZCON
0E99          178 ;
0E99 A2 FF    179 ^2    ldx #NEGONE
0E9B 9A       180      txs
0E9C          181 ;
0E9C A9 03    182      lda /DOSWARM-1

```

```

0E9E 48          183          pha
0E9F          184          ;
0E9F A9 CF      185          lda #DOSWARM-1
0EA1 48          186          pha
0EA2          187          ;
0EA2 A0 E0      188          ld y #EPBINEOS
0EA4 A9 C6      189          lda #HWSLOT16
0EA6          190          ;
0EA6          191          ;
0EA6 EA         192          nop
0EA7 EA         193          nop
0EA8          194          ;
0EA8          195          ;
0EA8          196          ; Exit from EOS with the assumption that a return to EOS
0EA8          197          ; will be made by means of EPMAPEOS. The X-reg must be
0EA8          198          ; used in order to turn this EPROM card OFF.
0EA8          199          ;
0EA8          200          .if LABEL
0EA8          201          RTNEXIT:
0EA8          202          .fi
0EA8          203          ;
0EA8          204          .if HWCARD
0EA8 A2 80      205          ldx #EPOFFVAL
0EAA          206          .el
0EAA          207          ldx #QLOFFVAL
0EAA          208          .fi
0EAA          209          ;
0EAA 8E E0 C0   210          stx EPSELC+HWSLOT16
0EAD          211          ;
0EAD A2 60      212          ldx #HWSLOT16
0EAF          213          ;
0EAF 6C A6 02   214          jmp (MEMJMP)
0EB2          215          ;
0EB2          216          ;
0EB2          217          dfs $B8-*&NEGONE,NEGONE
0EB8          218          ;
0EB8          219          ;
0EB8          220          ; Special entrance in order to turn this EPROM card OFF.
0EB8          221          ;
0EB8          222          .if LABEL
0EB8          223          EPOFF:
0EB8          224          .fi
0EB8          225          ;
0EB8          226          .if HWCARD
0EB8 A9 80      227          lda #EPOFFVAL
0EBA          228          .el
0EBA          229          lda #QLOFFVAL
0EBA          230          .fi
0EBA          231          ;
0EBA 8D E0 C0   232          sta EPSELC+HWSLOT16
0EBD          233          ;
0EBD 60         234          rts
0EBE          235          ;
0EBE          236          ;
0EBE          237          dfs $C0-*&NEGONE,NEGONE
0EC0          238          ;
0EC0          239          ;
0EC0          240          ; Return from DOS CMDUSER 1 command.
0EC0          241          ;
0EC0          242          .if LABEL
0EC0          243          EPUSER1 lda #EPONVAL

```

```

0EC0          244          .el
0EC0 A9 00      245          lda #EPONVAL
0EC2          246          .fi
0EC2          247          ;
0EC2 8D E0 C0   248          sta EPSELC+HWSLOT16
0EC5          249          ;
0EC5 4C 18 11   250          jmp USERRTN1
0EC8          251          ;
0EC8          252          ;
0EC8          253          ; Return from DOS CMDUSER 2 command.
0EC8          254          ;
0EC8          255          .if LABEL
0EC8          256 EPUSER2   lda #EPONVAL
0EC8          257          .el
0EC8 A9 00      258          lda #EPONVAL
0ECA          259          .fi
0ECA          260          ;
0ECA 8D E0 C0   261          sta EPSELC+HWSLOT16
0ECD          262          ;
0ECD 4C 16 14   263          jmp USERRTN2
0ED0          264          ;
0ED0          265          ;
0ED0          266          ; Entrance into the EOS+ card at the EPMAPEOS location by
0ED0          267          ; using SLOTMAP from the EOS mapping function.
0ED0          268          ;
0ED0          269          .if LABEL
0ED0          270 EPMAPEOS   lda #EPONVAL
0ED0          271          .el
0ED0 A9 00      272          lda #EPONVAL
0ED2          273          .fi
0ED2          274          ;
0ED2 8D E0 C0   275          sta EPSELC+HWSLOT16
0ED5          276          ;
0ED5          277          ;
0ED5          278          ; Initialize X-reg with this slot number and enter MAPEOS.
0ED5          279          ;
0ED5 A2 06      280          ldx #HWSLOT
0ED7          281          ;
0ED7 4C B0 10   282          jmp MAPEOS
0EDA          283          ;
0EDA          284          ;
0EDA          285          dfs $E0-*&NEGONE,NEGONE
0EE0          286          ;
0EE0          287          ;
0EE0          288          ; Interface entrance in order to process a BINEOS command
0EE0          289          ; that is contained within a BINEOS DCB. The Y-reg and the
0EE0          290          ; A-reg must contain the address of the eight byte DCB.
0EE0          291          ; The X-reg is initialized with the slot number of this
0EE0          292          ; EPROM card before calling BINEOS. BINEOS first calls
0EE0          293          ; DOZCOFF and MOVEEPBM before processing the BINEOS DCB.
0EE0          294          ; The X-reg must be used in order to turn this EPROM card
0EE0          295          ; ON.
0EE0          296          ;
0EE0          297          .if LABEL
0EE0          298 EPBINEOS   ldx #EPONVAL
0EE0          299          .el
0EE0 A2 00      300          ldx #EPONVAL
0EE2          301          .fi
0EE2          302          ;
0EE2 8E E0 C0   303          stx EPSELC+HWSLOT16
0EE5          304          ;

```

```

0EE5      305 ;
0EE5      306 ; Initialize X-reg with this slot number and then enter
0EE5      307 ; BINEOS.
0EE5      308 ;
0EE5 A2 06 309      ldx #HWSLOT
0EE7      310 ;
0EE7 4C B6 1B 311      jmp BINEOS
0EEA      312 ;
0EEA      313 ;
0EEA      314      dfs $F0-*)&NEGONE,NEGONE
0EF0      315 ;
0EF0      316 ;
0EF0      317 ; Manual entrance into EOS for this EPROM card.  If this
0EF0      318 ; EPROM card is not the highest priority EPROM card,
0EF0      319 ; control will pass to the highest priority EPROM card
0EF0      320 ; after the EPROM card mapping function has completed.
0EF0      321 ; CTRL-N may be used in order to select the desired
0EF0      322 ; EPROM card if there are mulitple EPROM cards in an
0EF0      323 ; Apple computer.
0EF0      324 ;
0EF0      325      .if LABEL
0EF0      326 EPEOS    lda #EPONVAL
0EF0      327      .el
0EF0 A9 00 328      lda #EPONVAL
0EF2      329      .fi
0EF2      330 ;
0EF2 8D E0 C0 331      sta EPSELC+HWSLOT16
0EF5      332 ;
0EF5 4C 00 10 333      jmp EOS
0EF8      334 ;
0EF8      335 ;
0EF8      336 ; This is the ASCII text that is compared to EPTEXT in
0EF8      337 ; order to determine if the slot that is being tested
0EF8      338 ; contains an EPROM card.
0EF8      339 ;
0EF8      340      .if LABEL
0EF8      341 EPBINTXT asc "EPBINEOS"
0EF8      342      .el
0EF8 C5 D0 C2 343      asc "EPBINEOS"
0EFB C9 CE C5
0EFE CF D3
0F00      344      .fi
0F00      345 ;
0F00      346 ;
0F00      347      icl "SLOT7.L"

```

LLOAD SLOT7.L,A\$4000

```

0F00      1          ttl "EOS+ Source Code, SLOT7.L"
0F00      2      ;
0F00      3      ;
0F00      4      ; SLOT7.L
0F00      5      ;
0F00      6      ;
0007      7      HWSLOT    let 7
0070      8      HWSLOT16 let $70
00C7      9      HWSLOT16 let $C7
0F00     10      ;
0000     11      LABEL     let 0
0F00     12      ;
0F00     13      ;
0F00     14      ; This is the generic code that is assembled specifically
0F00     15      ; for each of the seven slots in which an EPROM card may
0F00     16      ; reside.
0F00     17      ;
0F00     18      ; Interface to process an ASEOS command.
0F00     19      ;
0F00     20          .if LABEL
0F00     21      EPASEOS    sta ASAV
0F00     22          .el
0F00 85 18     23          sta ASAV
0F02     24          .fi
0F02     25      ;
0F02 18      26          clc
0F03 90 03     27          bcc >0                ; always taken
0F05     28      ;
0F05     29      ;
0F05     30      ; Insert TESTROM verification code here. When the CXRESET
0F05     31      ; routine is entered, it calls TSTROMCD to test for a ROM
0F05     32      ; card. TSTROMCD calls TESTROM that looks for 0x38xx18 at
0F05     33      ; 0xC305. If found, C3ROMOFF is enabled.
0F05     34      ;
0F05 38      35          sec
0F06 90 00     36          bcc *+2
0F08     37          dfs !-1
0F07 18      38          clc
0F08     39      ;
0F08     40      ;
0F08 86 16     41      ^0      stx XSAV
0F0A 84 17     42          sty YSAV
0F0C     43      ;
0F0C A5 76     44          lda CURLIN+1
0F0E C9 FF     45          cmp #RUNMODE
0F10 D0 03     46          bne >2
0F12     47      ;
0F12 A5 18     48      ^1      lda ASAV                ; recall A-reg
0F14     49      ;
0F14 60      50          rts
0F15     51      ;
0F15 A2 00     52      ^2      ldx #ZERO
0F17     53      ;
0F17 8E A0 02  54      ^3      stx ASPRNUM
0F1A     55      ;
0F1A 20 B7 00  56          jsr CHRGOT
0F1D F0 21     57          beq >4
0F1F     58      ;
0F1F 20 BE DE  59          jsr CHKCOM
0F22     60      ;

```



```

0F22 20 B7 00      61      jsr CHRGOT
0F25 F0 19        62      beq >4
0F27              63      ;
0F27 C9 2C        64      cmp #COMMA&MSBCLR
0F29 F0 15        65      beq >4
0F2B              66      ;
0F2B 20 E3 DF     67      jsr PTRGET
0F2E              68      ;
0F2E AE A0 02     69      ldx ASPRNUM
0F31 E0 0C        70      cpx #MAXASNUM      ; too many parameters
0F33 F0 DD        71      beq <1
0F35              72      ;
0F35 9D D0 02     73      sta ASPRADRS,X
0F38              74      ;
0F38 98          75      tya
0F39 9D D1 02     76      sta ASPRADRS+1,X
0F3C              77      ;
0F3C E8          78      inx
0F3D E8          79      inx
0F3E              80      ;
0F3E D0 D7       81      bne <3      ; always taken
0F40              82      ;
0F40 4E A0 02     83      ^4      lsr ASPRNUM
0F43 F0 CD        84      beq <1      ; no parameters
0F45              85      ;
0F45 A9 00        86      lda #EPONVAL
0F47 8D F0 C0     87      sta EPSELC+HWSLOT16
0F4A              88      ;
0F4A              89      ;
0F4A              90      ; Initialize X-reg with this slot number and enter ASEOS.
0F4A              91      ;
0F4A A2 07        92      ldx #HWSLOT
0F4C              93      ;
0F4C 4C 16 1A     94      jmp ASEOS
0F4F              95      ;
0F4F              96      ;
0F4F              97      dfs $50-*)&NEGONE,NEGONE
0F50              98      ;
0F50              99      ;
0F50            100      ; Exit for ASEOS and to RUN Applesoft files. If a ZipChip
0F50            101      ; is present, flush cache, enable ZipChip, and return to
0F50            102      ; the external caller or address in MEMJMP.
0F50            103      ;
0F50            104      .if LABEL
0F50            105      ASEXIT  lda ZSTATUS
0F50            106      .el
0F50 AD 9A 02     107      lda ZSTATUS
0F53            108      .fi
0F53            109      ;
0F53 30 14       110      bmi >2
0F55            111      ;
0F55 A0 00       112      ldy #ZERO
0F57 A9 60       113      lda /PAGE60
0F59            114      ;
0F59 84 CE       115      sty GENPTR
0F5B 85 CF       116      sta GENPTR+1
0F5D            117      ;
0F5D B1 CE       118      ^1      lda (GENPTR),Y
0F5F            119      ;
0F5F C8          120      iny
0F60 D0 FB       121      bne <1

```

```

0F62          122 ;
0F62 E6 CF    123      inc GENPTR+1
0F64 10 F7    124      bpl <1
0F66          125 ;
0F66 20 E1 1C 126      jsr DOZCON
0F69          127 ;
0F69          128      .if HWCARD
0F69 A9 80    129 ^2    lda #EPOFFVAL
0F6B          130      .el
0F6B          131 ^2    lda #QLOFFVAL
0F6B          132      .fi
0F6B          133 ;
0F6B 8D F0 C0 134      sta EPSELC+HWSLOT16
0F6E          135 ;
0F6E A5 18    136      lda ASAV
0F70 A6 16    137      ldx XSAV
0F72 A4 17    138      ldy YSAV
0F74          139 ;
0F74 6C A6 02 140      jmp (MEMJMP)
0F77          141 ;
0F77          142 ;
0F77          143      dfs $80-*)&NEGONE,NEGONE
0F80          144 ;
0F80          145 ;
0F80          146 ; Exit for BINEOS and to RUN binary files. If a ZipChip is
0F80          147 ; present, flush cache, enable ZipChip, push DOSWARM onto
0F80          148 ; the stack, load the address for EPBINEOS, and return to
0F80          149 ; the external caller or address in MEMJMP.
0F80          150 ;
0F80          151 ; Load the Y-reg/A-reg with the address of EPBINEOS in
0F80          152 ; order for the current file in memory to process more
0F80          153 ; DCBs if other EPROM files need to be loaded into memory.
0F80          154 ;
0F80          155      .if LABEL
0F80          156 BINEXIT  lda ZSTATUS
0F80          157      .el
0F80 AD 9A 02 158      lda ZSTATUS
0F83          159      .fi
0F83          160 ;
0F83 30 14    161      bmi >2
0F85          162 ;
0F85 A0 00    163      ldy #ZERO
0F87 A9 60    164      lda /PAGE60
0F89          165 ;
0F89 84 CE    166      sty GENPTR
0F8B 85 CF    167      sta GENPTR+1
0F8D          168 ;
0F8D B1 CE    169 ^1    lda (GENPTR),Y
0F8F          170 ;
0F8F C8       171      iny
0F90 D0 FB    172      bne <1
0F92          173 ;
0F92 E6 CF    174      inc GENPTR+1
0F94 10 F7    175      bpl <1
0F96          176 ;
0F96 20 E1 1C 177      jsr DOZCON
0F99          178 ;
0F99 A2 FF    179 ^2    ldx #NEGONE
0F9B 9A       180      txs
0F9C          181 ;
0F9C A9 03    182      lda /DOSWARM-1

```

```

0F9E 48          183          pha
0F9F          184          ;
0F9F A9 CF      185          lda #DOSWARM-1
0FA1 48          186          pha
0FA2          187          ;
0FA2 A0 E0      188          ld y #EPBINEOS
0FA4 A9 C7      189          lda #HWSLOT16
0FA6          190          ;
0FA6          191          ;
0FA6 EA        192          nop
0FA7 EA        193          nop
0FA8          194          ;
0FA8          195          ;
0FA8          196          ; Exit from EOS with the assumption that a return to EOS
0FA8          197          ; will be made by means of EPMAPEOS. The X-reg must be
0FA8          198          ; used in order to turn this EPROM card OFF.
0FA8          199          ;
0FA8          200          .if LABEL
0FA8          201          RTNEXIT:
0FA8          202          .fi
0FA8          203          ;
0FA8          204          .if HWCARD
0FA8 A2 80      205          ldx #EPOFFVAL
0FAA          206          .el
0FAA          207          ldx #QLOFFVAL
0FAA          208          .fi
0FAA          209          ;
0FAA 8E F0 C0   210          stx EPSELC+HWSLOT16
0FAD          211          ;
0FAD A2 70      212          ldx #HWSLOT16
0FAF          213          ;
0FAF 6C A6 02   214          jmp (MEMJMP)
0FB2          215          ;
0FB2          216          ;
0FB2          217          dfs $B8-*&NEGONE,NEGONE
0FB8          218          ;
0FB8          219          ;
0FB8          220          ; Special entrance in order to turn this EPROM card OFF.
0FB8          221          ;
0FB8          222          .if LABEL
0FB8          223          EPOFF:
0FB8          224          .fi
0FB8          225          ;
0FB8          226          .if HWCARD
0FB8 A9 80      227          lda #EPOFFVAL
0FBA          228          .el
0FBA          229          lda #QLOFFVAL
0FBA          230          .fi
0FBA          231          ;
0FBA 8D F0 C0   232          sta EPSELC+HWSLOT16
0FBD          233          ;
0FBD 60         234          rts
0FBE          235          ;
0FBE          236          ;
0FBE          237          dfs $C0-*&NEGONE,NEGONE
0FC0          238          ;
0FC0          239          ;
0FC0          240          ; Return from DOS CMDUSER 1 command.
0FC0          241          ;
0FC0          242          .if LABEL
0FC0          243          EPUSER1 lda #EPONVAL

```

```

0FC0          244          .el
0FC0 A9 00      245          lda #EPONVAL
0FC2          246          .fi
0FC2          247          ;
0FC2 8D F0 C0   248          sta EPSELC+HWSLOT16
0FC5          249          ;
0FC5 4C 18 11   250          jmp USERRTN1
0FC8          251          ;
0FC8          252          ;
0FC8          253          ; Return from DOS CMDUSER 2 command.
0FC8          254          ;
0FC8          255          .if LABEL
0FC8          256 EPUSER2   lda #EPONVAL
0FC8          257          .el
0FC8 A9 00      258          lda #EPONVAL
0FCA          259          .fi
0FCA          260          ;
0FCA 8D F0 C0   261          sta EPSELC+HWSLOT16
0FCD          262          ;
0FCD 4C 16 14   263          jmp USERRTN2
0FD0          264          ;
0FD0          265          ;
0FD0          266          ; Entrance into the EOS+ card at the EPMAPEOS location by
0FD0          267          ; using SLOTMAP from the EOS mapping function.
0FD0          268          ;
0FD0          269          .if LABEL
0FD0          270 EPMAPEOS   lda #EPONVAL
0FD0          271          .el
0FD0 A9 00      272          lda #EPONVAL
0FD2          273          .fi
0FD2          274          ;
0FD2 8D F0 C0   275          sta EPSELC+HWSLOT16
0FD5          276          ;
0FD5          277          ;
0FD5          278          ; Initialize X-reg with this slot number and enter MAPEOS.
0FD5          279          ;
0FD5 A2 07      280          ldx #HWSLOT
0FD7          281          ;
0FD7 4C B0 10   282          jmp MAPEOS
0FDA          283          ;
0FDA          284          ;
0FDA          285          dfs $E0-*&NEGONE,NEGONE
0FE0          286          ;
0FE0          287          ;
0FE0          288          ; Interface entrance in order to process a BINEOS command
0FE0          289          ; that is contained within a BINEOS DCB. The Y-reg and the
0FE0          290          ; A-reg must contain the address of the eight byte DCB.
0FE0          291          ; The X-reg is initialized with the slot number of this
0FE0          292          ; EPROM card before calling BINEOS. BINEOS first calls
0FE0          293          ; DOZCOFF and MOVEEPBM before processing the BINEOS DCB.
0FE0          294          ; The X-reg must be used in order to turn this EPROM card
0FE0          295          ; ON.
0FE0          296          ;
0FE0          297          .if LABEL
0FE0          298 EPBINEOS   ldx #EPONVAL
0FE0          299          .el
0FE0 A2 00      300          ldx #EPONVAL
0FE2          301          .fi
0FE2          302          ;
0FE2 8E F0 C0   303          stx EPSELC+HWSLOT16
0FE5          304          ;

```

```

0FE5          305 ;
0FE5          306 ; Initialize X-reg with this slot number and then enter
0FE5          307 ; BINEOS.
0FE5          308 ;
0FE5 A2 07    309         ldx #HWSLOT
0FE7          310 ;
0FE7 4C B6 1B 311         jmp BINEOS
0FEA          312 ;
0FEA          313 ;
0FEA          314         dfs $F0-*)&NEGONE,NEGONE
0FF0          315 ;
0FF0          316 ;
0FF0          317 ; Manual entrance into EOS for this EPROM card.  If this
0FF0          318 ; EPROM card is not the highest priority EPROM card,
0FF0          319 ; control will pass to the highest priority EPROM card
0FF0          320 ; after the EPROM card mapping function has completed.
0FF0          321 ; CTRL-N may be used in order to select the desired
0FF0          322 ; EPROM card if there are mulitple EPROM cards in an
0FF0          323 ; Apple computer.
0FF0          324 ;
0FF0          325         .if LABEL
0FF0          326 EPEOS    lda #EPONVAL
0FF0          327         .el
0FF0 A9 00    328         lda #EPONVAL
0FF2          329         .fi
0FF2          330 ;
0FF2 8D F0 C0 331         sta EPSELC+HWSLOT16
0FF5          332 ;
0FF5 4C 00 10 333         jmp EOS
0FF8          334 ;
0FF8          335 ;
0FF8          336 ; This is the ASCII text that is compared to EPTEXT in
0FF8          337 ; order to determine if the slot that is being tested
0FF8          338 ; contains an EPROM card.
0FF8          339 ;
0FF8          340         .if LABEL
0FF8          341 EPBINTXT asc "EPBINEOS"
0FF8          342         .el
0FF8 C5 D0 C2 343         asc "EPBINEOS"
0FFB C9 CE C5
0FFE CF D3
1000          344         .fi
1000          345 ;
1000          346 ;
1000          347         icl "MENU.L"

```

LLOAD MENU.L,A\$4000

```

1000      1          ttl "EOS+ Source Code, MENU.L"
1000      2      ;
1000      3      ;
1000      4      ; MENU.L
1000      5      ;
1000      6      ;
1000      7          .if DEBUG
1000      8          .el
1000      9          phs PAGEE8
1000     10          .fi
1000     11      ;
1000     12      ;
1000     13      ; This is the entrance for an EPROM card after a RESET has
1000     14      ; been pressed.
1000     15      ;
1000 D8     16 EOS      cld
1001     17      ;
1001     18      ;
1001     19      ; Initialize stack pointer and general state of computer.
1001     20      ;
1001 A2 FF   21          ldx #NEGONE
1003 9A     22          txs
1004     23      ;
1004 8E FB 04 24          stx XMODE
1007     25      ;
1007 8D 00 C0 26          sta STR80OFF
100A 8D 02 C0 27          sta RAMRDOFF
100D 8D 04 C0 28          sta RAMWROFF
1010 8D 08 C0 29          sta AUXZPOFF
1013 8D 0C C0 30          sta VID80OFF
1016 8D 0E C0 31          sta ALTCHOFF
1019     32      ;
1019     33      ;
1019     34      ; Determine if this is an Apple ][ or an Apple //e. Set
1019     35      ; APPLTYPE to 0x00 for Apple ][ and 0xFF for an Apple //e.
1019     36      ; The Apple ][ will fail either test because it does not
1019     37      ; have this particular hardware. Run test for 32 times
1019     38      ; in order to avoid random false positives.
1019     39      ;
1019 A2 00    40          ldx #ZERO
101B A0 20   41          ldy #TESTCNT
101D     42      ;
101D 8D 07 C0 43      ^1      sta CXROMON          ; enable internal CX ROM
1020     44      ;
1020 AD 15 C0 45          lda RDCXROM          ; read which ROM is enabled
1023 10 0C    46          bpl >2          ; BPL is wrong, so Apple ][
1025     47      ;
1025 8D 06 C0 48          sta CXROMOFF        ; enable slot ROM
1028     49      ;
1028 AD 15 C0 50          lda RDCXROM
102B 30 04    51          bmi >2          ; BMI is wrong, Apple ][
102D     52      ;
102D 88       53          dey
102E D0 ED    54          bne <1
1030     55      ;
1030 CA       56          dex
1031     57      ;
1031 8E 93 02  58      ^2      stx APPLTYPE
1034 8D 06 C0 59          sta CXROMOFF
1037     60      ;

```

```

1037      61 ;
1037      62 ; Set initial video and Language Card.
1037      63 ;
1037 AD 56 C0      64      lda HIRESOFF
103A AD 54 C0      65      lda PAGE1ON
103D AD 51 C0      66      lda TEXTON
1040      67 ;
1040 A9 00      68      lda #ZERO
1042 85 22      69      sta WNDTOP
1044 85 20      70      sta WNDLFT
1046      71 ;
1046 A9 18      72      lda #24
1048 85 23      73      sta WNDBTM
104A      74 ;
104A A9 28      75      lda #40
104C 85 21      76      sta WNDWDTH
104E      77 ;
104E AD 82 C0      78      lda ROM2WP
1051      79 ;
1051 AD FF CF      80      lda CLRROM
1054      81 ;
1054      82 ;
1054      83 ; If there is a ZipChip then disable it and turn it off.
1054      84 ; The annunciators are set by these two routines.
1054      85 ;
1054 20 A5 1C      86      jsr DOZCOFF
1057 20 FC 1C      87      jsr DOZCRSET
105A      88 ;
105A AD 9A 02      89      lda ZSTATUS
105D 49 10      90      eor #ZCSTAT      ; toggle ZipChip OFF
105F 8D 9A 02      91      sta ZSTATUS
1062      92 ;
1062      93 ;
1062      94 ; Initialize the page-zero pointers in order to discover
1062      95 ; in which slot this EPROM card resides.
1062      96 ;
1062 A0 00      97      ldy #ZERO
1064      98 ;
1064 8C 94 02      99      sty EPNMBR
1067 8C 95 02     100      sty EPBANK
106A      101 ;
106A A9 C1      102      lda /PAGEC1
106C      103 ;
106C 84 2A      104      sty SRCPTR
106E 85 2B      105      sta SRCPTR+1
1070      106 ;
1070 A9 E1      107      lda /PAGEE1
1072      108 ;
1072 84 2E      109      sty DSTPTR
1074 85 2F      110      sta DSTPTR+1
1076      111 ;
1076 8D 0B C0     112      sta C3ROMON      ; enable slot 3 ROM
1079      113 ;
1079 A2 01      114      ldx #1      ; begin with Slot 1
107B      115 ;
107B      116 ;
107B      117 ; Test for an EPROM card by comparing the EPTEXT in the
107B      118 ; Slot Page and in the EPROM page.
107B      119 ;
107B A0 F8      120 ^3      ldy #EPBINTXT
107D      121 ;

```

```

107D B9 44 25      122  ^4      lda EPTEXT-EPBINTXT&NEGONE,Y
1080              123  ;
1080 D1 2A         124      cmp (SRCPTR),Y      ; compare slot page
1082 D0 09         125      bne >5
1084              126  ;
1084 D1 2E         127      cmp (DSTPTR),Y      ; compare EPROM page
1086 D0 05         128      bne >5
1088              129  ;
1088 C8            130      iny
1089 D0 F2         131      bne <4
108B              132  ;
108B F0 10         133      beq >6              ; always taken
108D              134  ;
108D              135  ;
108D              136  ; Go to the next slot to test.
108D              137  ;
108D 2C FF CF      138  ^5      bit CLRROM
1090              139  ;
1090 E6 2B         140      inc SRCPTR+1        ; next slot page
1092 E6 2F         141      inc DSTPTR+1        ; next EPROM page
1094              142  ;
1094 E8            143      inx
1095              144  ;
1095 E0 08         145      cpx #8
1097 D0 E2         146      bne <3
1099              147  ;
1099              148  ;
1099              149  ; A horrible situation -- the hardware must be broken.
1099              150  ;
1099              151      .if DEBUG
1099 A2 04         152      ldx #4
109B EA           153      nop
109C EA           154      nop
109D              155      .el
109D              156      brk
109D              157      .fi
109D              158  ;
109D              159  ;
109D              160  ; Found this EPROM card. Build the SLOTMAP for the
109D              161  ; computer.
109D              162  ;
109D 2C FF CF      163  ^6      bit CLRROM
10A0              164  ;
10A0 8E 90 02      165      stx PRISLOT
10A3              166  ;
10A3 20 E7 22      167      jsr BUILDMAP
10A6              168  ;
10A6 AE 90 02      169      ldx PRISLOT
10A9 E0 03         170      cpx #SLOT3
10AB F0 03         171      beq MAPEOS
10AD              172  ;
10AD 8D 0A C0      173      sta C3ROMOFF        ; disable slot 3 ROM
10B0              174  ;
10B0              175  ;
10B0              176  ; This is the general entrance for an EPROM card.
10B0              177  ; Install the EP interface management routines.
10B0              178  ;
10B0 8E 91 02      179  MAPEOS  stx EPSLOT
10B3              180  ;
10B3 20 62 23      181      jsr MOVEEPBM
10B6              182  ;

```



```

10B6      183 ;
10B6      184 ; If the ESC key is pressed then exit to the monitor.
10B6      185 ;
10B6 AD 00 C0 186      lda KEY
10B9 C9 9B 187      cmp #ESCAPE
10BB D0 07 188      bne >1
10BD      189 ;
10BD A0 65 190      ldy #MONITOR
10BF A9 FF 191      lda /MONITOR
10C1      192 ;
10C1 4C 24 15 193      jmp EXITBIN
10C4      194 ;
10C4      195 ;
10C4      196 ; Initialize the screen.
10C4      197 ;
10C4 2C 10 C0 198 ^1      bit CLRKEY
10C7      199 ;
10C7 20 C8 20 200      jsr PRINT
10CA 51 201      byt NORMCMD
10CB 52 202      byt INITCMD
10CC 53 203      byt VIDCMD
10CD 54 204      byt KBDCMD
10CE 55 205      byt HOMECMD
10CF 50 206      byt RTNCMD
10D0      207 ;
10D0 A9 8D 208      lda #RETURN
10D2 8D 00 02 209      sta INPUT
10D5      210 ;
10D5      211 ;
10D5      212 ; Initialize the SDV values and ring the bell.
10D5      213 ;
10D5 A9 07 214      lda #7
10D7 85 EB 215      sta MSLOT
10D9      216 ;
10D9 A9 01 217      lda #1
10DB 85 EC 218      sta DRIVE
10DD      219 ;
10DD A9 00 220      lda #ZERO
10DF 85 ED 221      sta VOLUME
10E1      222 ;
10E1 20 32 1F 223      jsr EOSBELL
10E4      224 ;
10E4      225 ;
10E4      226 ; Check private slot variables for power up bytes. Load
10E4      227 ; DOS if in power up state.
10E4      228 ;
10E4      229      .if DEBUG
10E4 4C 35 11 230      jmp MAIN
10E7      231      .fi
10E7      232 ;
10E7 18 233      clc
10E8      234 ;
10E8 A9 78 235      lda #PWRUP3
10EA 6D 90 02 236      adc PRISLOT
10ED 85 2A 237      sta SRCPTR
10EF      238 ;
10EF A9 07 239      lda /PWRUP3
10F1 85 2B 240      sta SRCPTR+1
10F3      241 ;
10F3 A2 03 242      ldx #SYNC.L-1
10F5 A0 00 243      ldy #ZERO

```

```

10F7          244 ;
10F7 B1 2A    245 ^2      lda (SRCPTR),Y
10F9 DD 76 26 246      cmp SYNCBYTES,X
10FC D0 07    247      bne >3
10FE          248 ;
10FE C6 2B    249      dec SRCPTR+1
1100          250 ;
1100 CA       251      dex
1101 10 F4    252      bpl <2
1103          253 ;
1103 30 30    254      bmi MAIN          ; always taken
1105          255 ;
1105          256 ;
1105          257 ; If this is an Apple ][, load DOS 4.5.05L into memory.
1105          258 ; If this is an Apple //e, load DOS 4.5.06H into memory.
1105          259 ;
1105 2C 93 02  260 ^3      bit APPLTYPE
1108 30 06     261      bmi >4
110A          262 ;
110A 20 52 21  263      jsr LOADOSL
110D          264 ;
110D 4C 13 11  265      jmp >5
1110          266 ;
1110 20 6D 21  267 ^4      jsr LOADOSH
1113          268 ;
1113          269 ;
1113          270 ; Initialize DOS.  EPUSER1 will direct program flow to
1113          271 ; USERRTN1 before MAIN.
1113          272 ;
1113 A0 C0      273 ^5      ldy #EPUSER1
1115          274 ;
1115 4C FE 13   275      jmp BHNDLR2
1118          276 ;
1118          277 ;
1118          278 ; Install power up bytes into private slot variables.
1118          279 ;
1118 20 B8 20   280 USERRTN1 jsr CLRUSER
111B          281 ;
111B 18        282      clc
111C          283 ;
111C A9 78     284      lda #PWRUP3
111E 6D 90 02  285      adc PRISLOT
1121 85 2A     286      sta SRCPTR
1123          287 ;
1123 A9 07     288      lda /PWRUP3
1125 85 2B     289      sta SRCPTR+1
1127          290 ;
1127 A2 03     291      ldx #SYNC.L-1
1129 A0 00     292      ldy #ZERO
112B          293 ;
112B BD 76 26  294 ^6      lda SYNCBYTES,X
112E 91 2A     295      sta (SRCPTR),Y
1130          296 ;
1130 C6 2B     297      dec SRCPTR+1
1132          298 ;
1132 CA       299      dex
1133 10 F6     300      bpl <6
1135          301 ;
1135          302 ;
1135          303 ; EOS main menu display.
1135          304 ;

```

```

1135 20 24 21    305  MAIN      jsr DOSHOOK
1138              306  ;
1138 20 C8 20    307              jsr PRINT
113B 55          308              byt HOMECMD
113C 8D          309              byt RETURN
113D 02          310              hex 02
113E C5 CF D3    311              asc "EOS+ Main Menu Selection for Slot "
1141 AB A0 CD
1144 E1 E9 EE
1147 A0 CD E5
114A EE F5 A0
114D D3 E5 EC
1150 E5 E3 F4
1153 E9 EF EE
1156 A0 E6 EF
1159 F2 A0 D3
115C EC EF F4
115F A0
1160 50          312              byt RTNCMD
1161              313  ;
1161 AD 91 02    314              lda EPSLOT
1164 20 32 21    315              jsr DOPRHEX
1167              316  ;
1167 20 C8 20    317              jsr PRINT
116A 8D 8D        318              byt RETURN,RETURN
116C C1 A0 C4     319              asc "A DOS 4.5.05L"
116F CF D3 A0
1172 B4 AE B5
1175 AE B0 B5
1178 CC
1179 13          320              hex 13
117A CE A0 C3     321              asc "N Copy ROM->RAM"
117D EF F0 F9
1180 A0 D2 CF
1183 CD AD BE
1186 D2 C1 CD
1189 8D          322              byt RETURN
118A C2 A0 C4     323              asc "B DOS 4.5.06H"
118D CF D3 A0
1190 B4 AE B5
1193 AE B0 B6
1196 C8
1197 13          324              hex 13
1198 CF A0 D2     325              asc "O Run HELLO on SDV"
119B F5 EE A0
119E C8 C5 CC
11A1 CC CF A0
11A4 EF EE A0
11A7 D3 C4 D6
11AA 8D          326              byt RETURN
11AB C3 A0 C3     327              asc "C Coldstart DOS"
11AE EF EC E4
11B1 F3 F4 E1
11B4 F2 F4 A0
11B7 C4 CF D3
11BA 13          328              hex 13
11BB D0 A0 C3     329              asc "P CATALOG this SDV"
11BE C1 D4 C1
11C1 CC CF C7
11C4 A0 F4 E8
11C7 E9 F3 A0

```

```
11CA D3 C4 D6
11CD 8D          330      byt RETURN
11CE C4 A0 D7    331      asc "D Warmstart DOS"
11D1 E1 F2 ED
11D4 F3 F4 E1
11D7 F2 F4 A0
11DA C4 CF D3
11DD 13          332      hex 13
11DE D1 A0 C2    333      asc "Q BigMac"
11E1 E9 E7 CD
11E4 E1 E3
11E6 8D          334      byt RETURN
11E7 C5 A0 C2    335      asc "E Boot the Slot S"
11EA EF EF F4
11ED A0 F4 E8
11F0 E5 A0 D3
11F3 EC EF F4
11F6 A0 D3
11F8 13          336      hex 13
11F9 D2 A0 D3    337      asc "R Scan Disk"
11FC E3 E1 EE
11FF A0 C4 E9
1202 F3 EB
1204 8D          338      byt RETURN
1205 C6 A0 C8    339      asc "F Hook the Slot S"
1208 EF EF EB
120B A0 F4 E8
120E E5 A0 D3
1211 EC EF F4
1214 A0 D3
1216 13          340      hex 13
1217 D3 A0 C1    341      asc "S Applesoft List"
121A F0 F0 EC
121D E5 F3 EF
1220 E6 F4 A0
1223 CC E9 F3
1226 F4
1227 8D          342      byt RETURN
1228 C7 A0 D5    343      asc "G Unhook a Slot S"
122B EE E8 EF
122E EF EB A0
1231 E1 A0 D3
1234 EC EF F4
1237 A0 D3
1239 13          344      hex 13
123A D4 A0 C2    345      asc "T Binary Install"
123D E9 EE E1
1240 F2 F9 A0
1243 C9 EE F3
1246 F4 E1 EC
1249 EC
124A 8D          346      byt RETURN
124B C8 A0 D3    347      asc "H Sourceror"
124E EF F5 F2
1251 E3 E5 F2
1254 EF F2
1256 13          348      hex 13
1257 D5 A0 D6    349      asc "U VTOC Manager"
125A D4 CF C3
125D A0 CD E1
1260 EE E1 E7
```

```

1263 E5 F2
1265 8D          350      byt RETURN
1266 C9 A0 C5    351      asc "I EPROM Burner"
1269 D0 D2 CF
126C CD A0 C2
126F F5 F2 EE
1272 E5 F2
1274 13          352      hex 13
1275 D6 A0 D6    353      asc "V Volume Manager"
1278 EF EC F5
127B ED E5 A0
127E CD E1 EE
1281 E1 E7 E5
1284 F2
1285 8D          354      byt RETURN
1286 CA A0 C6    355      asc "J FID"
1289 C9 C4
128B 13          356      hex 13
128C D7 A0 D6    357      asc "W Volume Duplicate"
128F EF EC F5
1292 ED E5 A0
1295 C4 F5 F0
1298 EC E9 E3
129B E1 F4 E5
129E 8D          358      byt RETURN
129F CB A0 C1    359      asc "K ADT2"
12A2 C4 D4 B2
12A5 13          360      hex 13
12A6 D8 A0 C4    361      asc "X Disk Window"
12A9 E9 F3 EB
12AC A0 D7 E9
12AF EE E4 EF
12B2 F7
12B3 8D          362      byt RETURN
12B4 CC A0 CC    363      asc "L Lisa80"
12B7 E9 F3 E1
12BA B8 B0
12BC 13          364      hex 13
12BD D9 A0 D2    365      asc "Y Real Time Clock"
12C0 E5 E1 EC
12C3 A0 D4 E9
12C6 ED E5 A0
12C9 C3 EC EF
12CC E3 EB
12CE 8D          366      byt RETURN
12CF CD A0 D2    367      asc "M RamDisk Config"
12D2 E1 ED C4
12D5 E9 F3 EB
12D8 A0 C3 EF
12DB EE E6 E9
12DE E7
12DF 13          368      hex 13
12E0 DA A0 DA    369      asc "Z ZipChip Config"
12E3 E9 F0 C3
12E6 E8 E9 F0
12E9 A0 C3 EF
12EC EE E6 E9
12EF E7
12F0 8D 8D      370      byt RETURN,RETURN
12F2 A0 A0 D3    371      asc "  S=  D=  V="
12F5 BD A0 A0

```

```

12F8 C4 BD A0
12FB A0 A0 D6
12FE BD
12FF 13          372      hex 13
1300          373      ;
1300          374      .if HWCARD
1300 B0 B0 AD    375      asc "00-0F EPROM Catalog"
1303 B0 C6 A0
1306 C5 D0 D2
1309 CF CD A0
130C C3 E1 F4
130F E1 EC EF
1312 E7
1313          376      .el
1313          377      asc "00-07 EPROM Catalog"
1313          378      .fi
1313          379      ;
1313 8D 8D      380      byt RETURN,RETURN
1315 DE C3 A0   381      asc "^C Configure SDV"
1318 C3 EF EE
131B E6 E9 E7
131E F5 F2 E5
1321 A0 D3 C4
1324 D6
1325 13          382      hex 13
1326 D2 D4 CE   383      asc "RTN Toggle ZipChip"
1329 A0 D4 EF
132C E7 E7 EC
132F E5 A0 DA
1332 E9 F0 C3
1335 E8 E9 F0
1338 8D          384      byt RETURN
1339 DE D3 A0   385      asc "^S Skip EPROM Card"
133C D3 EB E9
133F F0 A0 C5
1342 D0 D2 CF
1345 CD A0 C3
1348 E1 F2 E4
134B 50          386      byt RTNCMD
134C          387      ;
134C          388      ;
134C 20 68 20   389      MAIN2      jsr PRTSDV
134F          390      ;
134F          391      ;
134F 20 C8 20   392      MAIN3      jsr PRINT
1352 17 74      393      hex 1774
1354 AD BE A0   394      asc "-> 0"
1357 CF
1358 1B          395      hex 1B
1359 50          396      byt RTNCMD
135A          397      ;
135A AD 9A 02   398      lda ZSTATUS
135D 30 0D      399      bmi >1
135F          400      ;
135F D0 0B      401      bne >1
1361          402      ;
1361 20 C8 20   403      jsr PRINT
1364 EE A0 BC    404      asc "n <- "
1367 AD A0
1369 50          405      byt RTNCMD
136A          406      ;

```

```

136A 90 09      407      bcc SELC          ; always taken
136C           408      ;
136C 20 C8 20   409      ^1      jsr PRINT
136F E6 E6 A0   410      asc "ff <-"
1372 BC AD
1374 50         411      byt RTNCMD
1375           412      ;
1375           413      ;
1375           414      ; EOS main menu selection.
1375           415      ;
1375 20 C8 20   416      SELC      jsr PRINT
1378 00 76      417      hex 0076
137A C5 EE F4   418      asc "Enter Selection:  "
137D E5 F2 A0
1380 D3 E5 EC
1383 E5 E3 F4
1386 E9 EF EE
1389 BA A0 A0
138C 57         419      byt EOLCMD
138D 50         420      byt RTNCMD
138E           421      ;
138E 20 42 1F   422      jsr RDKEY
1391           423      ;
1391 C9 83      424      cmp #CTRLC
1393 D0 06      425      bne >2
1395           426      ;
1395 20 00 20   427      jsr EDITS DV
1398           428      ;
1398 4C 4C 13   429      jmp MAIN2
139B           430      ;
139B           431      ;
139B           432      ; Confirm that SLOTMAP contains a valid value.  If not, do
139B           433      ; not process SLOTMAP.
139B           434      ;
139B C9 93      435      ^2      cmp #CTRLS
139D D0 0D      436      bne >3
139F           437      ;
139F AD 92 02   438      lda SLOTMAP
13A2 C9 02      439      cmp #2          ; value if card in slot 1
13A4 90 37      440      bcc SELCERR
13A6           441      ;
13A6 20 A0 20   442      jsr NEXTMAP
13A9           443      ;
13A9 4C 27 15   444      jmp EXITRTN
13AC           445      ;
13AC C9 8D      446      ^3      cmp #RETURN
13AE D0 0C      447      bne >4
13B0           448      ;
13B0 AD 9A 02   449      lda ZSTATUS
13B3 30 28      450      bmi SELCERR
13B5           451      ;
13B5 49 10      452      eor #ZCSTAT
13B7 8D 9A 02   453      sta ZSTATUS
13BA           454      ;
13BA 10 93      455      bpl MAIN3          ; always taken
13BC           456      ;
13BC C9 B0      457      ^4      cmp #"0"
13BE D0 15      458      bne >5
13C0           459      ;
13C0 20 3B 21   460      jsr DOCOUT
13C3 20 42 1F   461      jsr RDKEY

```

```

13C6      462 ;
13C6 20 5F 1F 463      jsr GETHEX
13C9 B0 12    464      bcs SELCERR
13CB      465 ;
13CB 20 3B 21 466      jsr DOCOUT
13CE      467 ;
13CE 8A      468      txa
13CF 20 69 15 469      jsr EOSCAT
13D2      470 ;
13D2 4C 35 11 471      jmp MAIN
13D5      472 ;
13D5 C9 C1    473 ^5      cmp #"A"
13D7 90 04    474      bcc SELCERR
13D9      475 ;
13D9 C9 DB    476      cmp #"Z"+1
13DB 90 05    477      bcc >6
13DD      478 ;
13DD 20 32 1F 479 SELCERR jsr EOSBELL
13E0 F0 93    480      beq SELC      ; always taken
13E2      481 ;
13E2 20 3B 21 482 ^6      jsr DOCOUT
13E5      483 ;
13E5 29 1F    484      and #MENUMASK
13E7      485 ;
13E7 0A      486      asl
13E8 A8      487      tay
13E9      488 ;
13E9 B9 22 25 489      lda CMDTBL-1,Y
13EC 48      490      pha
13ED      491 ;
13ED B9 21 25 492      lda CMDTBL-2,Y
13F0 48      493      pha
13F1      494 ;
13F1 60      495      rts
13F2      496 ;
13F2      497 ;
13F2      498 ; EOS main menu selection handlers.
13F2      499 ;
13F2      500 ; EPUSER2 will direct program flow to USERRTN2 before
13F2      501 ; DHNDLR.
13F2      502 ;
13F2 20 52 21 503 AHNDLR  jsr LOADOSL
13F5      504 ;
13F5 A0 C8    505      ldy #EPUSER2
13F7 D0 05    506      bne BHNDLR2      ; always taken
13F9      507 ;
13F9      508 ;
13F9 20 6D 21 509 BHNDLR  jsr LOADOSH
13FC      510 ;
13FC A0 C8    511      ldy #EPUSER2
13FE      512 ;
13FE AD 91 02 513 BHNDLR2  lda EPSLOT
1401 09 C0    514      ora /PAGEC0
1403      515 ;
1403 20 BA 20 516      jsr SETUSER
1406      517 ;
1406 AC F8 BF 518      ldy INITDOS
1409 AD F9 BF 519      lda INITDOS+1
140C      520 ;
140C 4C 27 15 521      jmp EXITRTN
140F      522 ;

```



```

140F          523 ;
140F A0 D3    524 CHNDLR    ldy #DOSCOLD
1411 A9 03    525          lda /DOSCOLD
1413          526 ;
1413 4C 24 15 527          jmp EXITBIN
1416          528 ;
1416          529 ;
1416 20 B8 20 530 USERRTN2 jsr CLRUSER
1419          531 ;
1419 20 39 21 532 DHNDLR    jsr DOCROUT
141C          533 ;
141C A0 D0    534          ldy #DOSWARM
141E A9 03    535          lda /DOSWARM
1420          536 ;
1420 4C 24 15 537          jmp EXITBIN
1423          538 ;
1423          539 ;
1423 A5 EB    540 EHNDLR    lda MSLOT
1425 85 18    541          sta ASAV
1427          542 ;
1427 A0 95    543          ldy #OUTPORT
1429 A9 FE    544          lda /OUTPORT
142B          545 ;
142B 4C 21 15 546          jmp EXITAS
142E          547 ;
142E          548 ;
142E A0 10    549 FHNDLR    ldy #HOOKSLT
1430          550 ;
1430 2C 00 00 551          bit *-*
1433          552          dfs !-2
1431          553 ;
1431          554 ;
1431 A0 18    555 GHNDLR    ldy #UHOOKSLT
1433          556 ;
1433 A5 EB    557          lda MSLOT
1435 09 C0    558          ora /PAGEC0
1437          559 ;
1437 20 49 21 560          jsr JSRMEM
143A 20 49 1F 561          jsr GETKEY
143D          562 ;
143D 4C 35 11 563          jmp MAIN
1440          564 ;
1440          565 ;
1440 A2 16    566 HHNDLR    ldx #LSDLEN
1442          567 ;
1442 A0 6A    568          ldy #LSDCB
1444 A9 24    569          lda /LSDCB
1446          570 ;
1446 4C 0A 15 571          jmp DOEOSDCB
1449          572 ;
1449          573 ;
1449 A2 14    574 IHNDLR    ldx #EBDLLEN
144B          575 ;
144B A0 80    576          ldy #EBDCB
144D A9 24    577          lda /EBDCB
144F          578 ;
144F 4C 0A 15 579          jmp DOEOSDCB
1452          580 ;
1452          581 ;
1452 A2 0B    582 JHNDLR    ldx #FDDLLEN
1454          583 ;

```

```

1454 A0 27      584      ldy #FDDCB
1456 A9 24      585      lda /FDDCB
1458           586      ;
1458 4C 0A 15    587      jmp DOEOSDCB
145B           588      ;
145B           589      ;
145B A2 0C      590 KHNDLR ldx #ADDLEN
145D           591      ;
145D A0 32      592      ldy #ADDCB
145F A9 24      593      lda /ADDCB
1461           594      ;
1461 4C 0A 15    595      jmp DOEOSDCB
1464           596      ;
1464           597      ;
1464 A2 12      598 LHNDLR ldx #LLDLEN
1466           599      ;
1466 A0 FF      600      ldy #LLDCB
1468 A9 23      601      lda /LLDCB
146A           602      ;
146A 4C 0A 15    603      jmp DOEOSDCB
146D           604      ;
146D           605      ;
146D A2 16      606 MHNDLR ldx #RDDLEN
146F           607      ;
146F A0 11      608      ldy #RDDCB
1471 A9 24      609      lda /RDDCB
1473           610      ;
1473 4C 0A 15    611      jmp DOEOSDCB
1476           612      ;
1476           613      ;
1476 20 B2 21    614 NHNDLR jsr COPYROM
1479           615      ;
1479 4C DD 13    616      jmp SELCERR
147C           617      ;
147C           618      ;
147C A2 01      619 OHNDLR ldx #1          ; HELOTEXT
147E           620      ;
147E 2C 00 00    621      bit *-*
1481           622      dfs !-2
147F           623      ;
147F           624      ;
147F A2 00      625 PHNDLR ldx #ZERO      ; CATTEXT
1481           626      ;
1481 20 51 15    627      jsr DOEXEC
1484           628      ;
1484 A2 02      629      ldx #2          ; SLTTEXT
1486 20 51 15    630      jsr DOEXEC
1489           631      ;
1489 A5 EB      632      lda MSLOT
148B 20 32 21    633      jsr DOPRHEX
148E           634      ;
148E A2 03      635      ldx #3          ; DRVTEXT
1490 20 51 15    636      jsr DOEXEC
1493           637      ;
1493 A5 EC      638      lda DRIVE
1495 20 2E 21    639      jsr DOPRBYTE
1498           640      ;
1498 A2 04      641      ldx #4          ; VOLTEXT
149A 20 51 15    642      jsr DOEXEC
149D           643      ;
149D A5 ED      644      lda VOLUME

```

```

149F 20 2E 21      645      jsr DOPRBYTE
14A2              646      ;
14A2 20 39 21      647      jsr DOCROUT
14A5 20 49 1F      648      jsr GETKEY
14A8              649      ;
14A8 4C 35 11      650      jmp MAIN
14AB              651      ;
14AB              652      ;
14AB A2 13          653      QHNDLR    ldx #BMDLEN
14AD              654      ;
14AD A0 04          655      ldy #BMDCB
14AF A9 25          656      lda /BMDCB
14B1              657      ;
14B1 4C 0A 15      658      jmp DOEOSDCB
14B4              659      ;
14B4              660      ;
14B4 A2 0C          661      RHNDLR    ldx #SDDLEN
14B6              662      ;
14B6 A0 17          663      ldy #SDDCB
14B8 A9 25          664      lda /SDDCB
14BA              665      ;
14BA 4C 0A 15      666      jmp DOEOSDCB
14BD              667      ;
14BD              668      ;
14BD A2 1B          669      SHNDLR    ldx #AFDLEN
14BF              670      ;
14BF A0 4F          671      ldy #AFDCB
14C1 A9 24          672      lda /AFDCB
14C3              673      ;
14C3 4C 0A 15      674      jmp DOEOSDCB
14C6              675      ;
14C6              676      ;
14C6 A2 1B          677      THNDLR    ldx #BFDLEN
14C8              678      ;
14C8 A0 E9          679      ldy #BFD CB
14CA A9 24          680      lda /BFD CB
14CC              681      ;
14CC 4C 0A 15      682      jmp DOEOSDCB
14CF              683      ;
14CF              684      ;
14CF A2 14          685      UHNDLR    ldx #VTDLEN
14D1              686      ;
14D1 A0 94          687      ldy #VTDCB
14D3 A9 24          688      lda /VTDCB
14D5              689      ;
14D5 4C 0A 15      690      jmp DOEOSDCB
14D8              691      ;
14D8              692      ;
14D8 A2 16          693      VHNDLR    ldx #VODLEN
14DA              694      ;
14DA A0 A8          695      ldy #VODCB
14DC A9 24          696      lda /VODCB
14DE              697      ;
14DE 4C 0A 15      698      jmp DOEOSDCB
14E1              699      ;
14E1              700      ;
14E1 A2 18          701      WHNDLR    ldx #VDDLEN
14E3              702      ;
14E3 A0 BE          703      ldy #VDDCB
14E5 A9 24          704      lda /VDDCB
14E7              705      ;

```

```

14E7 4C 0A 15      706      jmp DOEOSDCB
14EA              707      ;
14EA              708      ;
14EA A2 13        709      XHNDLR   ldx #DWDLEN
14EC              710      ;
14EC A0 D6        711      ldy #DWDCB
14EE A9 24        712      lda /DWDCB
14F0              713      ;
14F0 4C 0A 15      714      jmp DOEOSDCB
14F3              715      ;
14F3              716      ;
14F3 A2 11        717      YHNDLR   ldx #SCDLEN
14F5              718      ;
14F5 A0 3E        719      ldy #SCDCB
14F7 A9 24        720      lda /SCDCB
14F9              721      ;
14F9 4C 0A 15      722      jmp DOEOSDCB
14FC              723      ;
14FC              724      ;
14FC AD 9A 02     725      ZHNDLR   lda ZSTATUS
14FF              726      ;
14FF              727      .if DEBUG
14FF EA           728      nop
1500 EA           729      nop
1501              730      .el
1501              731      bmi >1
1501              732      .fi
1501              733      ;
1501 20 9D 1D      734      jsr ZCONFIG
1504              735      ;
1504 4C 35 11      736      jmp MAIN
1507              737      ;
1507 4C DD 13      738      ^1      jmp SELCERR
150A              739      ;
150A              740      ;
150A              741      ; Save the address of the selected DCB and copy the DCB
150A              742      ; contents to the DCBBUFR. The X-reg contains the number
150A              743      ; of bytes to copy. Call BINEOS2 to process the DCB.
150A              744      ;
150A              745      ; A RUN DCB will exit through EXITBIN and a LOAD DCB will
150A              746      ; return to MAIN.
150A              747      ;
150A 84 EE        748      DOEOSDCB sty CMDPTR
150C 85 EF        749      sta CMDPTR+1
150E              750      ;
150E A0 00        751      ldy #ZERO
1510              752      ;
1510 B1 EE        753      ^1      lda (CMDPTR),Y
1512 99 D0 02     754      sta DCBBUFR,Y
1515              755      ;
1515 C8           756      iny
1516              757      ;
1516 CA           758      dex
1517 D0 F7        759      bne <1
1519              760      ;
1519 A9 01        761      lda #INTERNAL
151B 20 D2 1B     762      jsr BINEOS2
151E              763      ;
151E 4C 35 11     764      jmp MAIN
1521              765      ;
1521              766      ;

```

```

1521          767 ; EXIT for all ASEOS utilization other than RUNMODE and
1521          768 ; OUTPORT.
1521          769 ;
1521 A2 50      770 EXITAS    ldx #ASEXIT
1523          771 ;
1523 2C 00 00   772          bit *-*
1526          773          dfs !-2
1524          774 ;
1524          775 ;
1524          776 ; EXIT for all external BINEOS utilization and RUNMODE.
1524          777 ;
1524 A2 80      778 EXITBIN   ldx #BINEXIT
1526          779 ;
1526 2C 00 00   780          bit *-*
1529          781          dfs !-2
1527          782 ;
1527          783 ;
1527          784 ; EXIT for the routines that return to EOS using USERRTN
1527          785 ; or the mapping function.
1527          786 ;
1527 A2 A8      787 EXITRTN   ldx #RTNEXIT
1529          788 ;
1529 8C A6 02    789          sty MEMJMP
152C 8D A7 02   790          sta MEMJMP+1
152F          791 ;
152F AD 91 02   792          lda EPSLOT
1532 09 C0      793          ora /PAGEC0
1534          794 ;
1534 8E A8 02    795          stx SLOTJMP
1537 8D A9 02   796          sta SLOTJMP+1
153A          797 ;
153A 2C FF CF   798          bit CLRROM
153D          799 ;
153D 6C A8 02   800          jmp (SLOTJMP)
1540          801 ;
1540          802 ;
1540 A9 FF      803 EXECTEXT  lda #RUNMODE
1542 85 76      804          sta CURLIN+1
1544          805 ;
1544 20 98 22    806          jsr SELCBANK
1547          807 ;
1547 A4 2A      808          ldy SRCPTR
1549 A5 2B      809          lda SRCPTR+1
154B          810 ;
154B 20 5F 15   811          jsr DOEXEC2
154E          812 ;
154E 4C 19 14   813          jmp DHNDLR
1551          814 ;
1551          815 ;
1551 BC 11 26    816 DOEXEC    ldy TEXTTBLL,X
1554 BD 16 26    817          lda TEXTTBLH,X
1557          818 ;
1557 A2 00      819          ldx #ZERO
1559 8E 94 02    820          stx EPNMBR
155C 8E 95 02    821          stx EPBANK
155F          822 ;
155F          823 ;
155F 84 FA      824 DOEXEC2   sty EXECPTR
1561 85 FB      825          sta EXECPTR+1
1563          826 ;
1563 20 24 21    827          jsr DOSHOOK

```

```
1566          828 ;
1566 4C 65 01 829      jmp EPEXEC
1569          830 ;
1569          831 ;
1569          832      icl "CAT.L"

LLOAD CAT.L,A$4000
```

```

1569          1          ttl "EOS+ Source Code, CAT.L"
1569          2          ;
1569          3          ;
1569          4          ; CAT.L
1569          5          ;
1569          6          ;
1569 8D 94 02      7  EOSCAT      sta  EPNMBR
156C          8          ;
156C AD 94 02      9  EOSCAT2    lda  EPNMBR
156F          10         ;
156F          11         .if HWCARD
156F 29 0F        12         and #EPMASK
1571          13         .el
1571          14         and #QLMASK
1571          15         .fi
1571          16         ;
1571 8D 94 02      17         sta  EPNMBR
1574          18         ;
1574 20 19 16      19         jsr  CATHDR
1577          20         ;
1577 20 D2 21      21         jsr  INITCAT
157A 90 75        22         bcc  >3
157C          23         ;
157C 20 C8 20      24         jsr  PRINT
157F 00 64        25         hex  0064
1581 58          26         byt  EOPCMD
1582 00 69        27         hex  0069
1584 59          28         byt  CNTRCMD
1585 D4 E8 E5      29         asc  "There is no EPROM"
1588 F2 E5 A0
158B E9 F3 A0
158E EE EF A0
1591 C5 D0 D2
1594 CF CD
1596 8D 8D        30         byt  RETURN,RETURN
1598 59          31         byt  CNTRCMD
1599 EF F2 A0      32         asc  "or EPROM Catalog"
159C C5 D0 D2
159F CF CD A0
15A2 C3 E1 F4
15A5 E1 EC EF
15A8 E7
15A9 8D 8D        33         byt  RETURN,RETURN
15AB 59          34         byt  CNTRCMD
15AC E9 EE A0      35         asc  "in this Socket."
15AF F4 E8 E9
15B2 F3 A0 D3
15B5 EF E3 EB
15B8 E5 F4 AE
15BB 00 76        36         hex  0076
15BD 59          37         byt  CNTRCMD
15BE D0 F2 E5      38         asc  "Press any key for next EPROM"
15C1 F3 F3 A0
15C4 E1 EE F9
15C7 A0 EB E5
15CA F9 A0 E6
15CD EF F2 A0
15D0 EE E5 F8
15D3 F4 A0 C5
15D6 D0 D2 CF

```

```

15D9 CD
15DA 50          39      byt RTNCMD
15DB          40      ;
15DB 20 49 1F    41      jsr GETKEY
15DE          42      ;
15DE C9 D1      43      cmp #"Q"
15E0 F0 29      44      beq >5
15E2          45      ;
15E2 C9 8B      46      cmp #UARROW
15E4 D0 06      47      bne >2
15E6          48      ;
15E6 CE 94 02   49      dec EPNMBR
15E9          50      ;
15E9 4C 6C 15   51      ^1    jmp EOSCAT2
15EC          52      ;
15EC EE 94 02   53      ^2    inc EPNMBR
15EF D0 F8      54      bne <1          ; always taken
15F1          55      ;
15F1 20 B4 16   56      ^3    jsr CHKCAT
15F4          57      ;
15F4 A9 00      58      lda #ZERO
15F6 8D 9D 02   59      sta NUMSELC
15F9          60      ;
15F9 20 5D 16   61      ^4    jsr CATFTR
15FC 20 DE 16   62      jsr SHOWCAT
15FF          63      ;
15FF A9 FF      64      lda #NEGONE
1601 8D D1 02   65      sta FIRSTIME
1604          66      ;
1604 20 3A 17   67      jsr SELCFILE
1607 90 F0      68      bcc <4
1609          69      ;
1609 D0 DE      70      bne <1
160B          71      ;
160B 20 C8 20   72      ^5    jsr PRINT
160E 52          73      byt INITCMD
160F 50          74      byt RTNCMD
1610          75      ;
1610 A9 00      76      lda #ZERO
1612 8D 94 02   77      sta EPNMBR
1615 8D 95 02   78      sta EPBANK
1618          79      ;
1618 60          80      rts
1619          81      ;
1619          82      ;
1619 20 C8 20   83      CATHDR jsr PRINT
161C 52          84      byt INITCMD
161D 55          85      byt HOMECMD
161E 07          86      hex 07
161F C5 D0 D2   87      asc "EPROM Card - EPROM Catalog"
1622 CF CD A0
1625 C3 E1 F2
1628 E4 A0 AD
162B A0 C5 D0
162E D2 CF CD
1631 A0 C3 E1
1634 F4 E1 EC
1637 EF E7
1639 8D          88      byt RETURN
163A 09          89      hex 09
163B D3 EC EF   90      asc "Slot "

```



```

163E F4 A0
1640 50          91      byt RTNCMD
1641          92      ;
1641 AD 91 02    93      lda EPSLOT
1644 20 32 21    94      jsr DOPRHEX
1647          95      ;
1647 20 C8 20    96      jsr PRINT
164A 17          97      hex 17
164B C5 D0 D2    98      asc "EPROM "
164E CF CD A0
1651 50          99      byt RTNCMD
1652          100     ;
1652 AD 94 02    101     lda EPNMBR
1655 20 32 21    102     jsr DOPRHEX
1658          103     ;
1658 A9 04       104     lda #4
165A 85 22       105     sta WNDTOP
165C          106     ;
165C 60          107     rts
165D          108     ;
165D          109     ;
165D 20 C8 20    110     CATFTR jsr PRINT
1660 52          111     byt INITCMD
1661 00 75       112     hex 0075
1663 58          113     byt EOPCMD
1664 05          114     hex 05
1665 D2 D4 CE    115     asc "RTN - File Info"
1668 A0 AD A0
166B C6 E9 EC
166E E5 A0 C9
1671 EE E6 EF
1674 1D          116     hex 1D
1675 A8 CC A9    117     asc "(L)oad"
1678 EF E1 E4
167B 8D          118     byt RETURN
167C 05          119     hex 05
167D D3 D0 C3    120     asc "SPC - Next EPROM"
1680 A0 AD A0
1683 CE E5 F8
1686 F4 A0 C5
1689 D0 D2 CF
168C CD
168D 1D          121     hex 1D
168E A8 D2 A9    122     asc "(R)un"
1691 F5 EE
1693 8D          123     byt RETURN
1694 05          124     hex 05
1695          125     ;
1695          126     .if HWCARD
1695 B0 AD C6     127     asc "0-F - Select EPROM"
1698 A0 AD A0
169B D3 E5 EC
169E E5 E3 F4
16A1 A0 C5 D0
16A4 D2 CF CD
16A7          128     .el
16A7          129     asc "0-7 - Select EPROM"
16A7          130     .fi
16A7          131     ;
16A7 1D          132     hex 1D
16A8 A8 D1 A9    133     asc "(Q)uit"

```

```

16AB F5 E9 F4
16AE 50          134      byt RTNCMD
16AF          135      ;
16AF A9 15      136      lda #21
16B1 85 23      137      sta WNDBTM
16B3          138      ;
16B3 60         139      rts
16B4          140      ;
16B4          141      ;
16B4 A9 00      142      CHKCAT   lda #ZERO
16B6 8D D3 02   143      sta NUMNTRYS
16B9 8D D4 02   144      sta LSTOPNTY
16BC 8D D5 02   145      sta NTRYSTRT
16BF          146      ;
16BF 20 EB 21   147      ^1      jsr GETENTRY
16C2          148      ;
16C2 AD B1 02   149      lda FILETYPE
16C5 F0 05      150      beq >2
16C7          151      ;
16C7 EE D3 02   152      inc NUMNTRYS
16CA D0 F3      153      bne <1
16CC          154      ;
16CC AD D3 02   155      ^2      lda NUMNTRYS
16CF C9 09      156      cmp #9
16D1 90 07      157      bcc >3
16D3          158      ;
16D3 E9 08      159      sbc #8
16D5 8D D4 02   160      sta LSTOPNTY
16D8          161      ;
16D8 A9 08      162      lda #8
16DA          163      ;
16DA 8D D6 02   164      ^3      sta NTRYEND
16DD          165      ;
16DD 60         166      rts
16DE          167      ;
16DE          168      ;
16DE 20 C8 20   169      SHOWCAT  jsr PRINT
16E1 00 64      170      hex 0064
16E3 58         171      byt EOPCMD
16E4 50         172      byt RTNCMD
16E5          173      ;
16E5 A9 00      174      lda #ZERO
16E7 8D D0 02   175      sta NUMSCRN
16EA          176      ;
16EA AD D5 02   177      lda NTRYSTRT
16ED 20 11 18   178      jsr GETFILE
16F0          179      ;
16F0 20 EB 21   180      ^1      jsr GETENTRY
16F3          181      ;
16F3 A9 05      182      lda #INDENT
16F5 85 24      183      sta CH
16F7          184      ;
16F7 A0 07      185      ldy #7
16F9          186      ;
16F9 AD B1 02   187      lda FILETYPE
16FC          188      ;
16FC 0A         189      ^2      asl
16FD B0 03      190      bcs >3
16FF          191      ;
16FF 88         192      dey
1700 D0 FA      193      bne <2

```

```

1702      194 ;
1702 B9 8E 25 195 ^3      lda PARMTYPE,Y
1705 20 3B 21 196      jsr DOCOUT
1708      197 ;
1708 20 C8 20 198      jsr PRINT
170B A0 B0 F8 199      asc " 0x"
170E 50      200      byt RTNCMD
170F      201 ;
170F AD B1 02 202      lda FILETYPE
1712 20 2E 21 203      jsr DOPRBYTE
1715      204 ;
1715 20 36 21 205      jsr DOSPACE
1718 20 36 21 206      jsr DOSPACE
171B      207 ;
171B A0 00      208      ldy #ZERO
171D      209 ;
171D B9 B8 02 210 ^4      lda FILENAME,Y
1720 20 3B 21 211      jsr DOCOUT
1723      212 ;
1723 C8      213      iny
1724      214 ;
1724 C0 18      215      cpy #NAME SIZE
1726 D0 F5      216      bne <4
1728      217 ;
1728 20 39 21 218      jsr DOCROUT
172B 20 39 21 219      jsr DOCROUT
172E      220 ;
172E EE D0 02 221      inc NUMSCRN
1731      222 ;
1731 AD D0 02 223      lda NUMSCRN
1734 CD D6 02 224      cmp NTRYEND
1737 D0 B7      225      bne <1
1739      226 ;
1739 60      227      rts
173A      228 ;
173A      229 ;
173A 2C D1 02 230 SELCFILE bit FIRSTIME
173D 30 07      231      bmi >1
173F      232 ;
173F A9 0C      233      lda #INDENT+7
1741 85 24      234      sta CH
1743      235 ;
1743 20 36 21 236      jsr DOSPACE
1746      237 ;
1746 A9 00      238 ^1      lda #ZERO
1748 8D D1 02 239      sta FIRSTIME
174B      240 ;
174B AD 9D 02 241      lda NUMSELC
174E 0A      242      asl
174F 69 04      243      adc #4
1751 85 25      244      sta CV
1753      245 ;
1753 20 20 21 246      jsr DOVTAB
1756      247 ;
1756 A9 0C      248      lda #INDENT+7
1758 85 24      249      sta CH
175A      250 ;
175A A9 BE      251      lda #">"
175C 20 3B 21 252      jsr DOCOUT
175F      253 ;
175F 20 49 1F 254 SELCFIL2 jsr GETKEY

```

```

1762          255 ;
1762 C9 88      256      cmp #LARROW
1764 F0 04      257      beq >3
1766          258 ;
1766 C9 8B      259      cmp #UARROW
1768 D0 12      260      bne >4
176A          261 ;
176A CE 9D 02   262 ^3      dec NUMSELC
176D 10 CB      263      bpl SELCFILE
176F          264 ;
176F EE 9D 02   265      inc NUMSELC
1772          266 ;
1772 AD D5 02   267      lda NTRYSTRT
1775 F0 C3      268      beq SELCFILE
1777          269 ;
1777 CE D5 02   270      dec NTRYSTRT
177A 10 50      271      bpl >7          ; always taken
177C          272 ;
177C C9 95      273 ^4      cmp #RARROW
177E F0 04      274      beq >5
1780          275 ;
1780 C9 8A      276      cmp #DARROW
1782 D0 1B      277      bne >6
1784          278 ;
1784 EE 9D 02   279 ^5      inc NUMSELC
1787          280 ;
1787 AD 9D 02   281      lda NUMSELC
178A CD D6 02   282      cmp NTRYEND
178D D0 AB      283      bne SELCFILE
178F          284 ;
178F CE 9D 02   285      dec NUMSELC
1792          286 ;
1792 AD D5 02   287      lda NTRYSTRT
1795 CD D4 02   288      cmp LSTOPNTY
1798 F0 A0      289      beq SELCFILE
179A          290 ;
179A EE D5 02   291      inc NTRYSTRT
179D D0 2D      292      bne >7          ; always taken
179F          293 ;
179F C9 8D      294 ^6      cmp #RETURN
17A1 D0 2D      295      bne CKCATCMD
17A3          296 ;
17A3 20 0A 18   297      jsr GETFILE2
17A6 20 24 18   298      jsr SHOWFILE
17A9          299 ;
17A9 20 C8 20   300      jsr PRINT
17AC 00 76      301      hex 0076
17AE 59         302      byt CNTRCMD
17AF D0 F2 E5   303      asc "Press any key to Continue"
17B2 F3 F3 A0
17B5 E1 EE F9
17B8 A0 EB E5
17BB F9 A0 F4
17BE EF A0 C3
17C1 EF EE F4
17C4 E9 EE F5
17C7 E5
17C8 50         304      byt RTNCMD
17C9          305 ;
17C9 20 49 1F   306      jsr GETKEY
17CC          307 ;

```

```

17CC 18          308 ^7      clc
17CD           309 ;
17CD 60          310      rts
17CE           311 ;
17CE 38          312 ^8      sec
17CF           313 ;
17CF 60          314      rts
17D0           315 ;
17D0           316 ;
17D0 C9 D1       317 CKCATCMD cmp #"Q"
17D2 F0 FA       318      beq <8
17D4           319 ;
17D4 20 5F 1F    320      jsr GETHEX
17D7 B0 07       321      bcs >1
17D9           322 ;
17D9 8E 94 02    323      stx EPNMBR
17DC           324 ;
17DC A9 01       325      lda #1
17DE D0 EE       326      bne <8                ; always taken
17E0           327 ;
17E0 C9 A0       328 ^1      cmp #SPACE
17E2 D0 05       329      bne >2
17E4           330 ;
17E4 EE 94 02    331      inc EPNMBR
17E7 D0 E5       332      bne <8                ; always taken
17E9           333 ;
17E9 C9 CC       334 ^2      cmp #"L"
17EB D0 04       335      bne >3
17ED           336 ;
17ED A9 00       337      lda #ZERO
17EF F0 06       338      beq >4                ; always taken
17F1           339 ;
17F1 C9 D2       340 ^3      cmp #"R"
17F3 D0 0D       341      bne >5
17F5           342 ;
17F5 A9 FF       343      lda #RUNMODE
17F7           344 ;
17F7 8D 9F 02    345 ^4      sta RUNFLAG
17FA           346 ;
17FA 20 0A 18    347      jsr GETFILE2
17FD           348 ;
17FD 2C B1 02    349      bit FILETYPE
1800 50 03       350      bvc >6
1802           351 ;
1802 4C 5F 17    352 ^5      jmp SELCFIL2
1805           353 ;
1805 20 EF 18    354 ^6      jsr RUNLOAD
1808           355 ;
1808 18          356      clc
1809           357 ;
1809 60          358      rts
180A           359 ;
180A           360 ;
180A           361 ; Read the selected file's parameters from the Catalog.
180A           362 ;
180A 38          363 GETFILE2 sec
180B           364 ;
180B AD 9D 02    365      lda NUMSELC
180E 6D D5 02    366      adc NTRYSTRT
1811           367 ;
1811           368 ;

```

```

1811 48          369  GETFILE pha
1812          370  ;
1812 20 D2 21    371          jsr INITCAT
1815          372  ;
1815 68          373          pla
1816 F0 0B       374          beq >2
1818          375  ;
1818 8D D2 02    376          sta FILECNT
181B          377  ;
181B 20 EB 21    378 ^1      jsr GETENTRY
181E          379  ;
181E CE D2 02    380          dec FILECNT
1821 D0 F8       381          bne <1
1823          382  ;
1823 60          383 ^2      rts
1824          384  ;
1824          385  ;
1824 A9 18       386 SHOWFILE lda #24
1826 85 23       387          sta WNDBTM
1828          388  ;
1828 20 C8 20    389          jsr PRINT
182B 00 64       390          hex 0064
182D 58          391          byt EOPCMD
182E 8D          392          byt RETURN
182F C6 E9 EC    393          asc "File Name - "
1832 E5 A0 CE
1835 E1 ED E5
1838 A0 AD A0
183B 50          394          byt RTNCMD
183C          395  ;
183C A0 00       396          ldy #ZERO
183E          397  ;
183E B9 B8 02    398 ^1      lda FILENAME,Y
1841 20 3B 21    399          jsr DOCOUT
1844          400  ;
1844 C8          401          iny
1845          402  ;
1845 CC 9E 02    403          cpy FLENGTH
1848 D0 F4       404          bne <1
184A          405  ;
184A 20 C8 20    406          jsr PRINT
184D 8D 8D       407          byt RETURN,RETURN
184F C6 E9 EC    408          asc "File Type - "
1852 E5 A0 D4
1855 F9 F0 E5
1858 A0 AD A0
185B 50          409          byt RTNCMD
185C          410  ;
185C A9 07       411          lda #7
185E 8D D8 02    412          sta INDEX
1861          413  ;
1861 AD B1 02    414          lda FILETYPE
1864 8D D7 02    415          sta FILTYPE
1867          416  ;
1867 0E D7 02    417 ^2      asl FILTYPE
186A 90 17       418          bcc >5
186C          419  ;
186C AE D8 02    420          ldx INDEX
186F          421  ;
186F BC AA 25    422          ldy TYPETBL,X
1872          423  ;

```

```

1872 B9 B2 25      424 ^3      lda TYPTEXTS,Y
1875 F0 06          425      beq >4
1877              426      ;
1877 20 3B 21      427      jsr DOCOUT
187A              428      ;
187A C8            429      iny
187B D0 F5          430      bne <3
187D              431      ;
187D 20 C8 20      432 ^4      jsr PRINT
1880 8D            433      byt RETURN
1881 0C            434      hex 0C
1882 50            435      byt RTNCMD
1883              436      ;
1883 CE D8 02      437 ^5      dec INDEX
1886 10 DF          438      bpl <2
1888              439      ;
1888 20 C8 20      440      jsr PRINT
188B 8D            441      byt RETURN
188C C6 E9 EC      442      asc "File Size - 0x"
188F E5 A0 D3
1892 E9 FA E5
1895 A0 AD A0
1898 B0 F8
189A 50            443      byt RTNCMD
189B              444      ;
189B AE B4 02      445      ldx LENVAL
189E AD B5 02      446      lda LENVAL+1
18A1              447      ;
18A1 20 28 21      448      jsr DOPRNTAX
18A4              449      ;
18A4 20 C8 20      450      jsr PRINT
18A7 8D 8D 8D      451      byt RETURN,RETURN,RETURN
18AA C6 F2 EF      452      asc "From EPROM Offset - 0x"
18AD ED A0 C5
18B0 D0 D2 CF
18B3 CD A0 CF
18B6 E6 E6 F3
18B9 E5 F4 A0
18BC AD A0 B0
18BF F8
18C0 50            453      byt RTNCMD
18C1              454      ;
18C1 AE B2 02      455      ldx SRCVAL
18C4 AD B3 02      456      lda SRCVAL+1
18C7              457      ;
18C7 20 28 21      458      jsr DOPRNTAX
18CA              459      ;
18CA 20 C8 20      460      jsr PRINT
18CD 8D 8D          461      byt RETURN,RETURN
18CF D4 EF A0      462      asc "To Memory Address - 0x"
18D2 CD E5 ED
18D5 EF F2 F9
18D8 A0 C1 E4
18DB E4 F2 E5
18DE F3 F3 A0
18E1 AD A0 B0
18E4 F8
18E5 50            463      byt RTNCMD
18E6              464      ;
18E6 AE B6 02      465      ldx DSTVAL
18E9 AD B7 02      466      lda DSTVAL+1

```

```

18EC          467 ;
18EC 4C 28 21 468      jmp DOPRNTAX
18EF          469 ;
18EF          470 ;
18EF          471 ; Use FILETYPE and RUNFLAG to LOAD or RUN the file.
18EF          472 ;
18EF A0 0E    473 RUNLOAD ldy #14
18F1          474 ;
18F1 AD B1 02 475      lda FILETYPE
18F4 29 1F    476      and #%00011111
18F6 F0 1F    477      beq >3
18F8          478 ;
18F8 0A       479 ^1     asl
18F9 B0 04    480      bcs >2
18FB          481 ;
18FB 88       482      dey
18FC 88       483      dey
18FD          484 ;
18FD D0 F9    485      bne <1
18FF          486 ;
18FF 2C 9F 02 487 ^2     bit RUNFLAG
1902 10 14    488      bpl LOADFILE
1904          489 ;
1904 8C 97 02 490      sty TEMPVAL
1907          491 ;
1907 20 C8 20 492      jsr PRINT
190A 52       493      byt INITCMD
190B 50       494      byt RTNCMD
190C          495 ;
190C AC 97 02 496      ldy TEMPVAL
190F          497 ;
190F          498 ;
190F B9 A1 25 499 RUNFILE lda RUNTBL+1,Y
1912 48       500      pha
1913          501 ;
1913 B9 A0 25 502      lda RUNTBL,Y
1916 48       503      pha
1917          504 ;
1917 60       505 ^3     rts
1918          506 ;
1918          507 ;
1918 B9 97 25 508 LOADFILE lda LOADTBL+1,Y
191B 48       509      pha
191C          510 ;
191C B9 96 25 511      lda LOADTBL,Y
191F 48       512      pha
1920          513 ;
1920 60       514      rts
1921          515 ;
1921          516 ;
1921 20 0A 18 517 LOADTEXT jsr GETFILE2
1924          518 ;
1924 A0 00     519      ldy #PAGE09
1926 A9 09     520      lda /PAGE09
1928          521 ;
1928 8C B6 02 522 ^1     sty DSTVAL
192B 8D B7 02 523      sta DSTVAL+1
192E          524 ;
192E 20 24 18 525      jsr SHOWFILE
1931          526 ;
1931 20 C8 20 527      jsr PRINT

```



```

1934 73          528          hex 73
1935 59          529          byt CNTRCMD
1936 C9 F3 A0    530          asc "Is this Memory Address okay?"
1939 F4 E8 E9
193C F3 A0 CD
193F E5 ED EF
1942 F2 F9 A0
1945 C1 E4 E4
1948 F2 E5 F3
194B F3 A0 EF
194E EB E1 F9
1951 BF
1952 76          531          hex 76
1953 59          532          byt CNTRCMD
1954 C5 D3 C3    533          asc "ESC      (Y)es      (N)o"
1957 A0 A0 A0
195A A0 A8 D9
195D A9 E5 F3
1960 A0 A0 A0
1963 A0 A8 CE
1966 A9 EF
1968 50          534          byt RTNCMD
1969          535          ;
1969 20 49 1F    536          ^2      jsr GETKEY
196C          537          ;
196C C9 9B       538          cmp #ESCAPE
196E F0 54       539          beq >8
1970          540          ;
1970 C9 D9       541          cmp #"Y"
1972 F0 4D       542          beq >7
1974          543          ;
1974 C9 CE       544          cmp #"N"
1976 F0 02       545          beq >3
1978          546          ;
1978 D0 EF       547          bne <2          ; always taken
197A          548          ;
197A 20 C8 20    549          ^3      jsr PRINT
197D 00 73       550          hex 0073
197F 58          551          byt EOPCMD
1980 C5 EE F4    552          asc "Enter Memory Address:  0x"
1983 E5 F2 A0
1986 CD E5 ED
1989 EF F2 F9
198C A0 C1 E4
198F E4 F2 E5
1992 F3 F3 BA
1995 A0 A0 B0
1998 F8
1999 50          553          byt RTNCMD
199A          554          ;
199A A9 00       555          lda #ZERO
199C 85 2E       556          sta DSTPTR
199E 85 2F       557          sta DSTPTR+1
19A0          558          ;
19A0 20 7E 1F    559          ^4      jsr GETNUM
19A3 B0 D5       560          bcs <3
19A5          561          ;
19A5 F0 F9       562          beq <4
19A7          563          ;
19A7 20 7E 1F    564          ^5      jsr GETNUM
19AA B0 F4       565          bcs <4

```

```

19AC          566 ;
19AC F0 F9    567      beq <5
19AE          568 ;
19AE 20 7E 1F 569 ^6      jsr GETNUM
19B1 B0 F4    570      bcs <5
19B3          571 ;
19B3 F0 F9    572      beq <6
19B5          573 ;
19B5 20 7E 1F 574      jsr GETNUM
19B8 B0 F4    575      bcs <6
19BA          576 ;
19BA A4 2E    577      ldy DSTPTR
19BC A5 2F    578      lda DSTPTR+1
19BE          579 ;
19BE 4C 28 19 580      jmp <1
19C1          581 ;
19C1 20 B2 22 582 ^7      jsr READBLK
19C4          583 ;
19C4 60       584 ^8      rts
19C5          585 ;
19C5          586 ;
19C5 A0 01    587 EXECAS   ldy #STARTAS
19C7 A9 08    588          lda /STARTAS
19C9          589 ;
19C9 8C B6 02 590          sty DSTVAL
19CC 8D B7 02 591          sta DSTVAL+1
19CF          592 ;
19CF 20 B2 22 593          jsr READBLK
19D2          594 ;
19D2 18       595          clc
19D3          596 ;
19D3 A9 01    597          lda #STARTAS
19D5 85 67    598          sta ASPGMST
19D7          599 ;
19D7 6D B4 02 600          adc LENVAL
19DA          601 ;
19DA 85 69    602          sta ASVARS
19DC 85 AF    603          sta ASPEND
19DE          604 ;
19DE A9 08    605          lda /STARTAS
19E0 85 68    606          sta ASPGMST+1
19E2          607 ;
19E2 6D B5 02 608          adc LENVAL+1
19E5          609 ;
19E5 85 6A    610          sta ASVARS+1
19E7 85 B0    611          sta ASPEND+1
19E9          612 ;
19E9 2C 9F 02 613          bit RUNFLAG
19EC 10 D6    614          bpl <8
19EE          615 ;
19EE A0 66    616          ldy #RUNAS
19F0 A9 D5    617          lda /RUNAS
19F2          618 ;
19F2 4C 21 15 619          jmp EXITAS
19F5          620 ;
19F5          621 ;
19F5 AD 89 C0 622 EXECBIN1  lda ROM1WE
19F8 AD 89 C0 623          lda ROM1WE
19FB          624 ;
19FB 4C 04 1A 625          jmp EXECBIN0
19FE          626 ;

```

```
19FE          627 ;
19FE AD 81 C0 628 EXECBIN2 lda ROM2WE
1A01 AD 81 C0 629          lda ROM2WE
1A04          630 ;
1A04          631 ;
1A04 20 B2 22 632 EXECBIN0 jsr READBLK
1A07          633 ;
1A07 2C 9F 02 634          bit RUNFLAG
1A0A 10 09     635          bpl >1
1A0C          636 ;
1A0C AC B6 02 637          ldy DSTVAL
1A0F AD B7 02 638          lda DSTVAL+1
1A12          639 ;
1A12 4C 24 15 640          jmp EXITBIN
1A15          641 ;
1A15 60       642 ^1      rts
1A16          643 ;
1A16          644 ;
1A16          645          icl "ASEOS.L"
```

```
LLOAD ASEOS.L,A$4000
```

```

1A16          1          ttl "EOS+ Source Code, ASEOS.L"
1A16          2          ;
1A16          3          ;
1A16          4          ; ASEOS.L
1A16          5          ;
1A16          6          ;
1A16          7          ; Entry for the external user of ASEOS. Pop the stack for
1A16          8          ; the address of the external routine that called ASEOS.
1A16          9          ;
1A16 8E A5 02   10 ASEOS      stx SLOT
1A19          11          ;
1A19 20 A5 1C   12          jsr DOZCOFF
1A1C          13          ;
1A1C AE A5 02   14          ldx SLOT
1A1F 20 62 23   15          jsr MOVEEPPBM
1A22          16          ;
1A22          17          ;
1A22          18          ; Begin ASEOS processing.
1A22          19          ;
1A22 A2 00      20          ldx #ASPCMD-ASPRADRS
1A24          21          ;
1A24 20 9E 1B   22          jsr GETASVAL          ; value in X-reg
1A27 D0 11      23          bne >1          ; value in A-reg must be zero
1A29          24          ;
1A29 AD A0 02   25          lda ASPRNUM
1A2C A0 FF      26          ldy #RUNMODE          ; run flag
1A2E          27          ;
1A2E E0 01      28          cpx #LOADCMD
1A30 F0 2F      29          beq >5
1A32          30          ;
1A32 E0 02      31          cpx #RUNCMD
1A34 F0 2C      32          beq >6
1A36          33          ;
1A36 E0 03      34          cpx #CATCMD
1A38 F0 3F      35          beq >8
1A3A          36          ;
1A3A          37          ;
1A3A          38          ; Unknown Command error.
1A3A          39          ;
1A3A A9 01      40 ^1          lda #ERR01
1A3C          41          ;
1A3C 2C 00 00   42          bit *-*
1A3F          43          dfs !-2
1A3D          44          ;
1A3D          45          ;
1A3D          46          ; Wrong Number of Parameters error.
1A3D          47          ;
1A3D A9 02      48 ^2          lda #ERR02
1A3F          49          ;
1A3F 2C 00 00   50          bit *-*
1A42          51          dfs !-2
1A40          52          ;
1A40          53          ;
1A40          54          ; Search Range Invalid error.
1A40          55          ;
1A40 A9 03      56 ^3          lda #ERR03
1A42          57          ;
1A42 2C 00 00   58          bit *-*
1A45          59          dfs !-2
1A43          60          ;

```

```

1A43      61 ;
1A43      62 ; File Not Found error.
1A43      63 ;
1A43 A9 04 64 ^4      lda #ERR04
1A45      65 ;
1A45 2C 00 00 66      bit *-*
1A48      67      dfs !-2
1A46      68 ;
1A46      69 ;
1A46      70 ; Return no error.
1A46      71 ;
1A46 A9 00 72 ^0      lda #ERR00
1A48      73 ;
1A48      74 ;
1A48      75 ; Exit ASEOS through EXITAS. All RUN commands are handled
1A48      76 ; by RUNFILE.
1A48      77 ;
1A48 8D A1 02 78      sta ASSTATUS
1A4B      79 ;
1A4B A2 02 80      ldx #ASPSTAT-ASPRADRS
1A4D 20 A9 1B 81      jsr SETASPTR
1A50      82 ;
1A50 AE A1 02 83      ldx ASSTATUS
1A53 20 94 1B 84      jsr SAVPARM2
1A56      85 ;
1A56 18 86      clc
1A57      87 ;
1A57 68 88      pla
1A58 69 01 89      adc #1
1A5A A8 90      tay
1A5B      91 ;
1A5B 68 92      pla
1A5C 69 00 93      adc #ZERO
1A5E      94 ;
1A5E 4C 21 15 95      jmp EXITAS
1A61      96 ;
1A61      97 ;
1A61      98 ; Command #1 processing, Load file.
1A61      99 ;
1A61 C8 100 ^5      iny
1A62      101 ;
1A62      102 ;
1A62      103 ; Command #2 processing, Run file. Check ASPRNUM.
1A62      104 ;
1A62 8C 9F 02 105 ^6      sty RUNFLAG
1A65      106 ;
1A65 C9 04 107      cmp #ASPNUM4
1A67 F0 04 108      beq >7
1A69      109 ;
1A69 C9 05 110      cmp #ASPNUM5
1A6B D0 D0 111      bne <2
1A6D      112 ;
1A6D      113 ;
1A6D      114 ; Get EPROM search range.
1A6D      115 ;
1A6D 20 66 1B 116 ^7      jsr GETEPRNG
1A70 B0 CE 117      bcs <3
1A72      118 ;
1A72      119 ;
1A72      120 ; Find the requested file and process.
1A72      121 ;

```

```

1A72 20 8B 1A 122      jsr DOASFILE
1A75 90 CF      123      bcc <0
1A77          124      ;
1A77 B0 CA      125      bcs <4                ; always taken
1A79          126      ;
1A79          127      ;
1A79          128      ; Command #3 processing, read Catalog.  Check ASPRNUM.
1A79          129      ;
1A79 C9 05      130      ^8      cmp #ASPNUM5
1A7B F0 04      131      beq >9
1A7D          132      ;
1A7D C9 06      133      cmp #ASPNUM6
1A7F D0 BC      134      bne <2
1A81          135      ;
1A81          136      ;
1A81          137      ; Get EPROM search range.
1A81          138      ;
1A81 20 66 1B   139      ^9      jsr GETEPRNG
1A84 B0 BA      140      bcs <3
1A86          141      ;
1A86 20 B3 1A   142      jsr DOASCAT
1A89 90 BB      143      bcc <0                ; always taken
1A8B          144      ;
1A8B          145      ;
1A8B          146      ; Load or run the requested EPROM file.  Applesoft string
1A8B          147      ; variables require three bytes:  length, low address, high
1A8B          148      ; address.  The first value returned in A-reg is length.
1A8B          149      ; X-reg contains the low address.  The third value is the
1A8B          150      ; high address of the string.
1A8B          151      ;
1A8B A2 06      152      DOASFILE ldx #ASPFIL-ASPRADRS
1A8D 20 9E 1B   153      jsr GETASVAL
1A90          154      ;
1A90 8D A3 02   155      sta FILELEN
1A93          156      ;
1A93 A0 02      157      ldy #2
1A95          158      ;
1A95 B1 CE      159      lda (GENPTR),Y        ; high address
1A97          160      ;
1A97 20 46 22   161      jsr FINDFILE
1A9A B0 16      162      bcs >2
1A9C          163      ;
1A9C          164      ;
1A9C          165      ; Found a matching filename.
1A9C          166      ;
1A9C AE A0 02   167      ldx ASPRNUM
1A9F E0 04      168      cpx #ASPNUM4
1AA1 F0 0B      169      beq >1
1AA3          170      ;
1AA3          171      ;
1AA3          172      ; Get optional run/load address.
1AA3          173      ;
1AA3 A2 08      174      ldx #ASPADR-ASPRADRS
1AA5 20 9E 1B   175      jsr GETASVAL
1AA8          176      ;
1AA8 8E B6 02   177      stx DSTVAL
1AAB 8D B7 02   178      sta DSTVAL+1
1AAE          179      ;
1AAE          180      ;
1AAE          181      ; If FILETYPE is zero, RUNLOAD will return, otherwise if
1AAE          182      ; the RUNFLAG is LOAD, a routine from LOADTBL will LOAD

```

```

1AAE      183 ; the file and return. If the RUNFLAG is RUN, a routine
1AAE      184 ; from RUNTBL will LOAD the file and exit EOS+ via EXITAS.
1AAE      185 ;
1AAE 20 EF 18 186 ^1      jsr RUNLOAD
1AB1      187 ;
1AB1 18      188      clc
1AB2      189 ;
1AB2 60      190 ^2      rts
1AB3      191 ;
1AB3      192 ;
1AB3      193 ; Read the requested EPROM catalog.
1AB3      194 ;
1AB3 AD 98 02 195 DOASCAT  lda EPSTRT
1AB6 8D 94 02 196      sta EPNMBR
1AB9      197 ;
1AB9 20 D2 21 198 ^1      jsr INITCAT
1ABC B0 0D      199      bcs >3
1ABE      200 ;
1ABE      201 ;
1ABE      202 ; Get a Catalog entry.
1ABE      203 ;
1ABE 20 EB 21 204 ^2      jsr GETENTRY
1AC1      205 ;
1AC1 AD B1 02 206      lda FILETYPE
1AC4 F0 05      207      beq >3
1AC6      208 ;
1AC6      209 ;
1AC6      210 ; Record the entry.
1AC6      211 ;
1AC6 20 D7 1A 212      jsr SAVFILE
1AC9 90 F3      213      bcc <2                ; always taken
1ACB      214 ;
1ACB      215 ;
1ACB      216 ; Check next EPROM.
1ACB      217 ;
1ACB EE 94 02 218 ^3      inc EPNMBR
1ACE      219 ;
1ACE AD 99 02 220      lda EPEND
1AD1 CD 94 02 221      cmp EPNMBR
1AD4 B0 E3      222      bcs <1
1AD6      223 ;
1AD6 60      224      rts
1AD7      225 ;
1AD7      226 ;
1AD7      227 ; Record the entry.
1AD7      228 ;
1AD7 A2 06      229 SAVFILE  ldx #ASPNUM-ASPRADRS
1AD9 20 A9 1B 230      jsr SETASPTR
1ADC      231 ;
1ADC 18      232      clc
1ADD      233 ;
1ADD C8      234      iny
1ADE      235 ;
1ADE B1 CE      236      lda (GENPTR),Y
1AE0 69 01      237      adc #1
1AE2 91 CE      238      sta (GENPTR),Y
1AE4      239 ;
1AE4 88      240      dey
1AE5      241 ;
1AE5 B1 CE      242      lda (GENPTR),Y
1AE7 69 00      243      adc #ZERO

```

```

1AE9 91 CE      244      sta (GENPTR),Y
1AEB           245      ;
1AEB           246      ;
1AEB           247      ; Get memory space for the filename.
1AEB           248      ;
1AEB AD 9E 02   249      lda FLENGTH
1AEE 85 08      250      sta AREG
1AF0           251      ;
1AF0 A0 D5      252      ldy #STRINI
1AF2 A9 E3      253      lda /STRINI
1AF4           254      ;
1AF4 20 49 21   255      jsr JSRMEM
1AF7           256      ;
1AF7           257      ;
1AF7           258      ; Record the filename length and address.
1AF7           259      ;
1AF7 A2 08      260      ldx #ASPFILS-ASPRADRS
1AF9 20 A9 1B   261      jsr SETASPTR
1AFC           262      ;
1AFC A0 02      263      ldy #2
1AFE           264      ;
1AFE B9 9D 00   265      ^1  lda DSCTMP,Y
1B01 91 CE      266      sta (GENPTR),Y
1B03           267      ;
1B03 88         268      dey
1B04 10 F8      269      bpl <1
1B06           270      ;
1B06 18         271      clc
1B07           272      ;
1B07 A5 CE      273      lda GENPTR
1B09 69 03      274      adc #3
1B0B 8D D8 02   275      sta ASPFILES
1B0E           276      ;
1B0E A5 CF      277      lda GENPTR+1
1B10 69 00      278      adc #ZERO
1B12 8D D9 02   279      sta ASPFILES+1
1B15           280      ;
1B15           281      ;
1B15           282      ; Copy the filename.
1B15           283      ;
1B15 A0 00      284      ldy #ZERO
1B17           285      ;
1B17 B9 B8 02   286      ^2  lda FILENAME,Y
1B1A 29 7F      287      and #MSBCLR
1B1C 91 9E      288      sta (DSCTMP+1),Y
1B1E           289      ;
1B1E C8         290      iny
1B1F           291      ;
1B1F C4 9D      292      cpy DSCTMP
1B21 D0 F4      293      bne <2
1B23           294      ;
1B23 AE A0 02   295      ldx ASPRNUM
1B26 E0 05      296      cpx #ASPNUM5
1B28 F0 3A      297      beq RTNCLC
1B2A           298      ;
1B2A           299      ;
1B2A           300      ; Initialize GENPTR with the ASPPARMS array.
1B2A           301      ;
1B2A A2 0A      302      ldx #ASPPARMS-ASPRADRS
1B2C 20 A9 1B   303      jsr SETASPTR
1B2F           304      ;

```



```

1B2F          305 ;
1B2F          306 ; Save EPNMBR.
1B2F          307 ;
1B2F AE 94 02 308         ldx EPNMBR
1B32 20 94 1B 309         jsr SAVPARM2
1B35          310 ;
1B35          311 ;
1B35          312 ; Save file type.
1B35          313 ;
1B35 AE B1 02 314         ldx FILETYPE
1B38 20 94 1B 315         jsr SAVPARM2
1B3B          316 ;
1B3B          317 ;
1B3B          318 ; Save EPROM offset.
1B3B          319 ;
1B3B AE B2 02 320         ldx SRCVAL
1B3E AD B3 02 321         lda SRCVAL+1
1B41          322 ;
1B41 20 96 1B 323         jsr SAVPARM
1B44          324 ;
1B44          325 ;
1B44          326 ; Save file length.
1B44          327 ;
1B44 AE B4 02 328         ldx LENVAL
1B47 AD B5 02 329         lda LENVAL+1
1B4A          330 ;
1B4A 20 96 1B 331         jsr SAVPARM
1B4D          332 ;
1B4D          333 ;
1B4D          334 ; Save memory address.
1B4D          335 ;
1B4D AE B6 02 336         ldx DSTVAL
1B50 AD B7 02 337         lda DSTVAL+1
1B53          338 ;
1B53 20 96 1B 339         jsr SAVPARM
1B56          340 ;
1B56          341 ;
1B56          342 ; Point to next array entry.
1B56          343 ;
1B56 18       344         clc
1B57          345 ;
1B57 98       346         tya
1B58 65 CE    347         adc GENPTR
1B5A 8D DA 02 348         sta ASPPARMS
1B5D          349 ;
1B5D A5 CF    350         lda GENPTR+1
1B5F 69 00    351         adc #ZERO
1B61 8D DB 02 352         sta ASPPARMS+1
1B64          353 ;
1B64 18       354 RTNCLC   clc
1B65          355 ;
1B65 60       356         rts
1B66          357 ;
1B66          358 ;
1B66          359 ; Get the EPROM search range in X-reg and verify.
1B66          360 ;
1B66 A2 04    361 GETEPRNG ldx #ASPSRCH-ASPRADRS
1B68          362 ;
1B68 20 9E 1B 363         jsr GETASVAL
1B6B D0 25    364         bne >2           ; value in A-reg must be zero
1B6D          365 ;

```

```

1B6D 8E A2 02    366  GETRANGE stx EPSEARCH
1B70              367  ;
1B70 8A          368              txa
1B71              369  ;
1B71              370              .if HWCARD
1B71 C9 10       371              cmp #EPMASK+1
1B73              372              .el
1B73              373              cmp #QLMASK+1
1B73              374              .fi
1B73              375  ;
1B73 90 12       376              bcc >1
1B75              377  ;
1B75 29 0F       378              and #EPMASK
1B77              379  ;
1B77              380              .if HWCARD
1B77 C9 10       381              cmp #EPMASK+1
1B79              382              .el
1B79              383              cmp #QLMASK+1
1B79              384              .fi
1B79              385  ;
1B79 B0 18       386              bcs >3
1B7B              387  ;
1B7B AA          388              tax
1B7C              389  ;
1B7C AD A2 02    390              lda EPSEARCH
1B7F              391  ;
1B7F 4A          392              lsr
1B80 4A          393              lsr
1B81 4A          394              lsr
1B82 4A          395              lsr
1B83              396  ;
1B83              397              .if HWCARD
1B83 C9 10       398              cmp #EPMASK+1
1B85              399              .el
1B85              400              cmp #QLMASK+1
1B85              401              .fi
1B85              402  ;
1B85 B0 0C       403              bcs >3
1B87              404  ;
1B87 8E 98 02    405  ^1          stx EPSTRT
1B8A 8D 99 02    406              sta EPEND
1B8D              407  ;
1B8D CD 98 02    408              cmp EPSTRT
1B90 B0 D2       409              bcs RTNCLC
1B92              410  ;
1B92 38          411  ^2          sec
1B93              412  ;
1B93 60          413  ^3          rts
1B94              414  ;
1B94              415  ;
1B94              416  ; Save Applesoft value in ASPPARMS array.
1B94              417  ;
1B94 A9 00       418  SAVPARM2 lda #ZERO
1B96              419  ;
1B96 91 CE       420  SAVPARM  sta (GENPTR),Y
1B98              421  ;
1B98 C8          422              iny
1B99              423  ;
1B99 8A          424              txa
1B9A 91 CE       425              sta (GENPTR),Y
1B9C              426  ;

```

```
1B9C C8          427          iny
1B9D             428          ;
1B9D 60          429          rts
1B9E             430          ;
1B9E             431          ;
1B9E             432          ; Get the requested Applesoft value.
1B9E             433          ;
1B9E 20 A9 1B    434  GETASVAL jsr SETASPTR
1BA1             435          ;
1BA1 C8          436          iny
1BA2             437          ;
1BA2 B1 CE       438          lda (GENPTR),Y
1BA4 AA          439          tax
1BA5             440          ;
1BA5 88          441          dey
1BA6             442          ;
1BA6 B1 CE       443          lda (GENPTR),Y
1BA8             444          ;
1BA8 60          445          rts
1BA9             446          ;
1BA9             447          ;
1BA9             448          ; Set GENPTR to the requested address.
1BA9             449          ;
1BA9 BC D0 02    450  SETASPTR ldy ASPRADRS,X
1BAC BD D1 02    451          lda ASPRADRS+1,X
1BAF             452          ;
1BAF 84 CE       453          sty GENPTR
1BB1 85 CF       454          sta GENPTR+1
1BB3             455          ;
1BB3 A0 00       456          ldy #ZERO
1BB5             457          ;
1BB5 60          458          rts
1BB6             459          ;
1BB6             460          ;
1BB6             461          icl "BINEOS.L"
```

LLOAD BINEOS.L,A\$4000

```

1BB6          1          ttl "EOS+ Source Code, BINEOS.L"
1BB6          2          ;
1BB6          3          ;
1BB6          4          ; BINEOS.L
1BB6          5          ;
1BB6          6          ;
1BB6          7          ; Entry for the external user of BINEOS. Pop the stack for
1BB6          8          ; the address of the external routine that called BINEOS.
1BB6          9          ;
1BB6 8E A5 02   10 BINEOS   stx SLOT
1BB9          11          ;
1BB9 84 EE     12          sty CMDPTR
1BBB 85 EF     13          sta CMDPTR+1
1BBD          14          ;
1BBD 20 A5 1C   15          jsr DOZCOFF
1BC0          16          ;
1BC0 AE A5 02   17          ldx SLOT
1BC3 20 62 23   18          jsr MOVEEPBM
1BC6          19          ;
1BC6 A0 07     20          ldy #DCBSIZE-1
1BC8          21          ;
1BC8 B1 EE     22 ^1      lda (CMDPTR),Y
1BCA 99 D0 02   23          sta DCBBUFR,Y
1BCD          24          ;
1BCD 88        25          dey
1BCE 10 F8     26          bpl <1
1BD0          27          ;
1BD0 A9 02     28          lda #EXTERNAL
1BD2          29          ;
1BD2          30          ;
1BD2          31          ; Initialize Y-reg with RUNMODE and process DCB command.
1BD2          32          ; Assume CMDPTR contains the address of DCBBUFR.
1BD2          33          ;
1BD2 8D 96 02   34 BINEOS2  sta RTNTYPE
1BD5          35          ;
1BD5 A0 FF     36          ldy #RUNMODE
1BD7          37          ;
1BD7 AD D0 02   38          lda DCBCMD
1BDA          39          ;
1BDA C9 01     40          cmp #1                ; load file
1BDC F0 30     41          beq BINLOAD
1BDE          42          ;
1BDE C9 02     43          cmp #2                ; run file
1BE0 F0 2D     44          beq BINRUN
1BE2          45          ;
1BE2 C9 03     46          cmp #3                ; catalog EPROM
1BE4 F0 5F     47          beq BINCAT
1BE6          48          ;
1BE6          49          ;
1BE6          50          ; Return Unknown Command error.
1BE6          51          ;
1BE6 A9 01     52          lda #ERR01
1BE8          53          ;
1BE8 2C 00 00   54          bit *-*
1BEB          55          dfs !-2
1BE9          56          ;
1BE9          57          ;
1BE9          58          ; Return Filename Length Invalid error.
1BE9          59          ;
1BE9 A9 02     60 ^2      lda #ERR02

```

```

1BEB      61 ;
1BEB 2C 00 00 62          bit *-*
1BEE      63          dfs !-2
1BEC      64 ;
1BEC      65 ;
1BEC      66 ; Return Search Range Invalid error.
1BEC      67 ;
1BEC A9 03    68 ^3      lda #ERR03
1BEE      69 ;
1BEE 2C 00 00 70          bit *-*
1BF1      71          dfs !-2
1BEF      72 ;
1BEF      73 ;
1BEF      74 ; Buffer/Filename Address error.
1BEF      75 ;
1BEF A9 04    76 ^4      lda #ERR04
1BF1      77 ;
1BF1 2C 00 00 78          bit *-*
1BF4      79          dfs !-2
1BF2      80 ;
1BF2      81 ;
1BF2      82 ; Requested File Not Found error.
1BF2      83 ;
1BF2 A9 05    84 ^5      lda #ERR05
1BF4      85 ;
1BF4 2C 00 00 86          bit *-*
1BF7      87          dfs !-2
1BF5      88 ;
1BF5      89 ;
1BF5      90 ; Exit BINEOS through EXITBIN for an external caller or
1BF5      91 ; return to MAIN for an internal caller for a LOAD DCB.
1BF5      92 ; All RUN DCBs are handled by RUNFILE.
1BF5      93 ;
1BF5 A9 00    94 BINDONE  lda #ERR00
1BF7      95 ;
1BF7 A0 04    96          ldy #DCBSTAT-DCBCMD
1BF9      97 ;
1BF9 91 EE    98          sta (CMDPTR),Y
1BFB      99 ;
1BFB AD 96 02 100         lda RTNTYPE
1BFE C9 01    101         cmp #INTERNAL
1C00 F0 0B    102         beq >1
1C02      103 ;
1C02 18      104         clc
1C03      105 ;
1C03 68      106         pla
1C04 69 01    107         adc #1
1C06 A8      108         tay
1C07      109 ;
1C07 68      110         pla
1C08 69 00    111         adc #ZERO
1C0A      112 ;
1C0A 4C 24 15 113         jmp EXITBIN
1C0D      114 ;
1C0D 60      115 ^1      rts
1C0E      116 ;
1C0E      117 ;
1C0E      118 ; Command #1 processing, Load file.
1C0E      119 ;
1C0E C8      120 BINLOAD  iny
1C0F      121 ;

```

```

1C0F      122 ;
1C0F      123 ; Command 2 processing, Run file.
1C0F      124 ;
1C0F 8C 9F 02 125 BINRUN    sty RUNFLAG
1C12      126 ;
1C12      127 ;
1C12      128 ; Get the filename length and verify.
1C12      129 ;
1C12 AD D5 02 130          lda DCBFLEN
1C15 C9 18    131          cmp #NAME SIZE
1C17 B0 D0    132          bcs <2
1C19      133 ;
1C19 8D A3 02 134          sta FILELEN
1C1C      135 ;
1C1C      136 ;
1C1C      137 ; Get EPROM search range.
1C1C      138 ;
1C1C AE D1 02 139          ldx DCBEPN
1C1F      140 ;
1C1F 20 6D 1B 141          jsr GETRANGE
1C22 B0 C8    142          bcs <3
1C24      143 ;
1C24      144 ;
1C24      145 ; Find the requested file and process.
1C24      146 ;
1C24 AE D6 02 147          ldx DCBFADR
1C27 AD D7 02 148          lda DCBFADR+1
1C2A F0 C3    149          beq <4
1C2C      150 ;
1C2C 20 46 22 151          jsr FINDFILE
1C2F B0 C1    152          bcs <5
1C31      153 ;
1C31      154 ;
1C31      155 ; Get alternate run/load address.
1C31      156 ;
1C31 AE D2 02 157          ldx DCBFALT
1C34 AD D3 02 158          lda DCBFALT+1
1C37 F0 06    159          beq >1
1C39      160 ;
1C39 8E B6 02 161          stx DSTVAL
1C3C 8D B7 02 162          sta DSTVAL+1
1C3F      163 ;
1C3F      164 ;
1C3F      165 ; If FILETYPE is zero, RUNLOAD will return, otherwise if
1C3F      166 ; the RUNFLAG is LOAD, a routine from LOADTBL will LOAD
1C3F      167 ; the file and return. If the RUNFLAG is RUN, a routine
1C3F      168 ; from RUNTBL will LOAD the file and exit EOS+ via EXITBIN
1C3F      169 ; with Y-reg/A-reg initialized from DSTVAL and DSTVAL+1.
1C3F      170 ;
1C3F 20 EF 18 171 ^1      jsr RUNLOAD
1C42      172 ;
1C42 4C F5 1B 173          jmp BINDONE
1C45      174 ;
1C45      175 ;
1C45      176 ; Command 3 processing, read Catalog.
1C45      177 ;
1C45      178 ; Initialize number of entries.
1C45      179 ;
1C45 A9 00    180 BINCAT    lda #ZERO
1C47 8D D5 02 181          sta DCBFLEN
1C4A      182 ;

```

```

1C4A      183 ;
1C4A      184 ; Get EPROM search range.
1C4A      185 ;
1C4A AE D1 02 186         ldx DCBEPN
1C4D      187 ;
1C4D 20 6D 1B 188         jsr GETRANGE
1C50 B0 9A      189         bcs <3
1C52      190 ;
1C52      191 ;
1C52      192 ; Get buffer address.
1C52      193 ;
1C52 AE D6 02 194         ldx DCBFADR
1C55 AD D7 02 195         lda DCBFADR+1
1C58 F0 95      196         beq <4
1C5A      197 ;
1C5A 86 CE      198         stx GENPTR
1C5C 85 CF      199         sta GENPTR+1
1C5E      200 ;
1C5E      201 ;
1C5E      202 ; Read the Catalog.
1C5E      203 ;
1C5E AD 98 02 204         lda EPSTRT
1C61 8D 94 02 205         sta EPNMBR
1C64      206 ;
1C64 20 D2 21 207 ^1      jsr INITCAT
1C67 B0 27      208         bcs >5
1C69      209 ;
1C69      210 ;
1C69      211 ; Get a Catalog entry.
1C69      212 ;
1C69 20 EB 21 213 ^2      jsr GETENTRY
1C6C      214 ;
1C6C AD B1 02 215         lda FILETYPE
1C6F F0 1F      216         beq >5
1C71      217 ;
1C71      218 ;
1C71      219 ; Copy the entry.
1C71      220 ;
1C71 A0 1F      221         ldy #ENTRYLEN-1
1C73      222 ;
1C73 B9 B0 02 223 ^3      lda FILEENTRY,Y
1C76 91 CE      224         sta (GENPTR),Y
1C78      225 ;
1C78 88      226         dey
1C79 10 F8      227         bpl <3
1C7B      228 ;
1C7B 18      229         clc
1C7C      230 ;
1C7C A5 CE      231         lda GENPTR
1C7E 69 20      232         adc #ENTRYLEN
1C80 85 CE      233         sta GENPTR
1C82      234 ;
1C82 90 02      235         bcc >4
1C84      236 ;
1C84 E6 CF      237         inc GENPTR+1
1C86      238 ;
1C86 EE D5 02 239 ^4      inc DCBFLEN
1C89 D0 DE      240         bne <2
1C8B      241 ;
1C8B CE D5 02 242         dec DCBFLEN
1C8E D0 0B      243         bne >6

```

```

; overflow, so stop
; always taken

```

```
1C90          244 ;
1C90          245 ;
1C90          246 ; Check next EPROM.
1C90          247 ;
1C90 EE 94 02 248 ^5      inc EPNMBR
1C93          249 ;
1C93 AD 99 02 250      lda EPEND
1C96 CD 94 02 251      cmp EPNMBR
1C99 B0 C9    252      bcs <1
1C9B          253 ;
1C9B A0 05    254 ^6      ld y #DCBFLEN-DCBCMD
1C9D          255 ;
1C9D AD D5 02 256      lda DCBFLEN
1CA0 91 EE    257      sta (CMDPTR),Y
1CA2          258 ;
1CA2 4C F5 1B 259      jmp BINDONE
1CA5          260 ;
1CA5          261 ;
1CA5          262      icl "ZIP.L"
```

```
LLOAD ZIP.L,A$4000
```



```

1CA5          1          ttl "EOS+ Source Code, ZIP.L"
1CA5          2          ;
1CA5          3          ;
1CA5          4          ; ZIP.L
1CA5          5          ;
1CA5          6          ;
1CA5          7          ; Unlock the ZipChip, test if present, get its status, set
1CA5          8          ; slots 1 and 6 to normal, and disable it.
1CA5          9          ;
1CA5 A9 80     10 DOZCOFF  lda #$80
1CA7 8D 9A 02  11          sta ZSTATUS
1CAA          12          ;
1CAA 20 7D 1D  13          jsr DOZCOPEN
1CAD          14          ;
1CAD AD 5C C0  15          lda ZCSLOTS
1CB0 49 FF     16          eor #NEGONE
1CB2 8D 5C C0  17          sta ZCSLOTS
1CB5          18          ;
1CB5 CD 5C C0  19          cmp ZCSLOTS
1CB8 D0 1A     20          bne SETANNUN
1CBA          21          ;
1CBA 49 FF     22          eor #NEGONE
1CBC 8D 5C C0  23          sta ZCSLOTS
1CBF          24          ;
1CBF CD 5C C0  25          cmp ZCSLOTS
1CC2 D0 10     26          bne SETANNUN
1CC4          27          ;
1CC4 8E 9A 02  28          stx ZSTATUS          ; save state of Bit 4
1CC7          29          ;
1CC7 A9 40     30          lda #%01000000      ; paddle fast, LC cache OFF
1CC9 8D 9B 02  31          sta ZCACHE
1CCC          32          ;
1CCC A9 C0     33          lda #%11000000      ; paddle normal, LC cache OFF
1CCE 8D 5F C0  34          sta ZCCACHE
1CD1          35          ;
1CD1 8D 5A C0  36          sta ZCCTRL          ; disables ZipChip
1CD4          37          ;
1CD4 AD 58 C0  38 SETANNUN lda ANN1OFF
1CD7 AD 5A C0  39          lda ANN2OFF
1CDA AD 5D C0  40          lda ANN3ON
1CDD AD 5F C0  41          lda ANN4ON
1CE0          42          ;
1CE0 60        43          rts
1CE1          44          ;
1CE1          45          ;
1CE1          46          ; Unlock the ZipChip if present and if it was on before,
1CE1          47          ; turn it back on.
1CE1          48          ;
1CE1 AD 9A 02  49 DOZCON  lda ZSTATUS
1CE4 30 13     50          bmi >2
1CE6          51          ;
1CE6 20 7D 1D  52          jsr DOZCOPEN
1CE9          53          ;
1CE9 AD 9A 02  54          lda ZSTATUS          ; BNE to disable ZipChip
1CEC D0 08     55          bne >1
1CEE          56          ;
1CEE AD 9B 02  57          lda ZCACHE          ; recall paddle fast, LC OFF
1CF1 8D 5F C0  58          sta ZCCACHE          ; save setting
1CF4          59          ;
1CF4 A9 A5     60          lda #ZCLOCK          ; lock ZipChip

```

```

1CF6      61 ;
1CF6 8D 5A C0 62 ^1      sta ZCCTRL
1CF9      63 ;
1CF9 4C D4 1C 64 ^2      jmp SETANNUN
1CFC      65 ;
1CFC      66 ;
1CFC      67 ; Reset the ZipChip in a power up state.
1CFC      68 ;
1CFC AD 9A 02 69 DOZCRSET lda ZSTATUS
1CFF 30 12    70      bmi >1
1D01      71 ;
1D01 20 7D 1D 72      jsr DOZCOPEN
1D04      73 ;
1D04 AD 88 25 74      lda ZCDEFLT      ; slots
1D07 8D 5C C0 75      sta ZCSLOTS
1D0A      76 ;
1D0A AD 89 25 77      lda ZCDEFLT+1    ; speed
1D0D 8D 5D C0 78      sta ZCSPEED
1D10      79 ;
1D10 8D 5A C0 80      sta ZCCTRL      ; lock ZipChip
1D13      81 ;
1D13 4C D4 1C 82 ^1      jmp SETANNUN
1D16      83 ;
1D16      84 ;
1D16      85 ; Read the ZipChip configuration; unable to read current
1D16      86 ; ZipChip speed. Assume it is set to maximum speed.
1D16      87 ;
1D16 20 7D 1D 88 DOZCREAD jsr DOZCOPEN
1D19      89 ;
1D19 A0 0B    90      ldy #ZCOPTNS-1
1D1B      91 ;
1D1B A9 00    92      lda #ZERO
1D1D      93 ;
1D1D 99 D0 02 94 ^1      sta ZCSETBL,Y
1D20      95 ;
1D20 88      96      dey
1D21 10 FA    97      bpl <1
1D23      98 ;
1D23 8A      99      txa      ; BEQ if ZipChip enabled
1D24 F0 03   100      beq >2
1D26      101 ;
1D26 EE D0 02 102      inc ZCSETBL      ; state
1D29      103 ;
1D29      104 ;
1D29      105 ; Extract default language card caching.
1D29      106 ;
1D29 AD 9B 02 107 ^2      lda ZCACHE
1D2C      108 ;
1D2C 0A      109      asl
1D2D 2E D2 02 110      rol ZCSETBL+2      ; language card cache
1D30      111 ;
1D30      112 ;
1D30      113 ; Extract default paddle delay.
1D30      114 ;
1D30 0A      115      asl
1D31 2E D3 02 116      rol ZCSETBL+3
1D34      117 ;
1D34 A2 07    118      ldx #7
1D36      119 ;
1D36 AD 5C C0 120      lda ZCSLOTS
1D39      121 ;

```

```

1D39 0A          122  ^3      asl
1D3A 3E D4 02    123      rol ZCSETBL+4,X      ; slots and speaker
1D3D             124      ;
1D3D CA          125      dex
1D3E 10 F9       126      bpl <3
1D40             127      ;
1D40 8D 5A C0    128      sta ZCCTRL
1D43             129      ;
1D43 60          130      rts
1D44             131      ;
1D44             132      ;
1D44             133      ; Save the ZipChip configuration.
1D44             134      ;
1D44 20 7D 1D    135  DOZCSAVE jsr DOZCOPEN
1D47             136      ;
1D47 AD D0 02    137      lda ZCSETBL      ; state
1D4A 29 01       138      and #1
1D4C F0 02       139      beq >1
1D4E             140      ;
1D4E A9 10       141      lda #ZCOFFVAL
1D50             142      ;
1D50 8D 9A 02    143  ^1      sta ZSTATUS
1D53             144      ;
1D53 AE D1 02    145      ldx ZCSETBL+1      ; speed
1D56             146      ;
1D56 BD 8A 25    147      lda ZCSPDTBL,X
1D59 8D 5D C0    148      sta ZCSPEED
1D5C             149      ;
1D5C A9 00       150      lda #ZERO
1D5E             151      ;
1D5E 4E D3 02    152      lsr ZCSETBL+3      ; paddle
1D61 6A          153      ror
1D62             154      ;
1D62 4E D2 02    155      lsr ZCSETBL+2      ; language card
1D65 6A          156      ror
1D66             157      ;
1D66 8D 9B 02    158      sta ZCACHE
1D69             159      ;
1D69 A2 07       160      ldx #7
1D6B             161      ;
1D6B A9 00       162      lda #ZERO
1D6D             163      ;
1D6D 5E D4 02    164  ^2      lsr ZCSETBL+4,X
1D70 2A          165      rol
1D71             166      ;
1D71 CA          167      dex
1D72 10 F9       168      bpl <2
1D74             169      ;
1D74 8D 5C C0    170      sta ZCSLOTS
1D77             171      ;
1D77 A9 00       172      lda #ZERO
1D79 8D 5A C0    173      sta ZCCTRL
1D7C             174      ;
1D7C 60          175      rts
1D7D             176      ;
1D7D             177      ;
1D7D A9 5A       178  DOZCOPEN lda #ZCUNLOCK
1D7F             179      ;
1D7F 8D 5A C0    180      sta ZCCTRL
1D82 8D 5A C0    181      sta ZCCTRL
1D85 8D 5A C0    182      sta ZCCTRL

```

```

1D88 8D 5A C0      183      sta ZCCTRL
1D8B              184      ;
1D8B AD 5B C0      185      lda ZCSTATS          ; get ZipChip status
1D8E 29 10          186      and #ZCSTAT          ; mask out enabled bit
1D90 AA            187      tax
1D91              188      ;
1D91 8D 5B C0      189      sta ZCSTATS          ; enable ZipChip
1D94              190      ;
1D94 A0 D8          191      ldy #DOZCOPEN-DOZCOFF
1D96              192      ;
1D96 B9 A4 1C      193      ^1 lda DOZCOFF-1,Y
1D99              194      ;
1D99 88            195      dey
1D9A D0 FA          196      bne <1
1D9C              197      ;
1D9C 60            198      rts
1D9D              199      ;
1D9D              200      ;
1D9D 20 C8 20      201      ZCONFIG jsr PRINT
1DA0 55            202      byt HOMECMD
1DA1 59            203      byt CNTRCMD
1DA2 DA E9 F0      204      asc "ZipChip Configuration"
1DA5 C3 E8 E9
1DA8 F0 A0 C3
1DAB EF EE E6
1DAE E9 E7 F5
1DB1 F2 E1 F4
1DB4 E9 EF EE
1DB7 00 63          205      hex 0063
1DB9 DA E9 F0      206      asc "ZipChip State"
1DBC C3 E8 E9
1DBF F0 A0 D3
1DC2 F4 E1 F4
1DC5 E5
1DC6 8D 8D          207      byt RETURN,RETURN
1DC8 DA E9 F0      208      asc "ZipChip Speed"
1DCB C3 E8 E9
1DCE F0 A0 D3
1DD1 F0 E5 E5
1DD4 E4
1DD5 8D 8D          209      byt RETURN,RETURN
1DD7 CC E1 EE      210      asc "Language Card Caching"
1DDA E7 F5 E1
1DDD E7 E5 A0
1DE0 C3 E1 F2
1DE3 E4 A0 C3
1DE6 E1 E3 E8
1DE9 E9 EE E7
1DEC 8D 8D          211      byt RETURN,RETURN
1DEE A0 D0 E1      212      asc " Paddle Speed"
1DF1 E4 E4 EC
1DF4 E5 A0 D3
1DF7 F0 E5 E5
1DFA E4
1DFB 8D            213      byt RETURN
1DFC D3 F0 E5      214      asc "Speaker Speed"
1DFF E1 EB E5
1E02 F2 A0 D3
1E05 F0 E5 E5
1E08 E4
1E09 8D            215      byt RETURN

```

```

1E0A 50          216      byt RTNCMD
1E0B          217      ;
1E0B A9 B1      218      lda #"1"
1E0D 8D 97 02   219      sta TEMPVAL
1E10          220      ;
1E10 20 C8 20   221      ^1    jsr PRINT
1E13 8D          222      byt RETURN
1E14 A0 D3 EC   223      asc " Slot "
1E17 EF F4 A0
1E1A 50          224      byt RTNCMD
1E1B          225      ;
1E1B AD 97 02   226      lda TEMPVAL
1E1E 20 3B 21   227      jsr DOCOUT
1E21          228      ;
1E21 20 C8 20   229      jsr PRINT
1E24 A0 D3 F0   230      asc " Speed"
1E27 E5 E5 E4
1E2A 50          231      byt RTNCMD
1E2B          232      ;
1E2B EE 97 02   233      inc TEMPVAL
1E2E          234      ;
1E2E AD 97 02   235      lda TEMPVAL
1E31 C9 B8      236      cmp #"8"
1E33 D0 DB      237      bne <1
1E35          238      ;
1E35 20 C8 20   239      jsr PRINT
1E38 75          240      hex 75
1E39 59          241      byt CNTRCMD
1E3A BC AD A0   242      asc "<- Select ->"
1E3D D3 E5 EC
1E40 E5 E3 F4
1E43 A0 AD BE
1E46 77          243      hex 77
1E47 59          244      byt CNTRCMD
1E48 A8 D0 A9   245      asc "(P)revious      (N)ext      (Q)uit"
1E4B F2 E5 F6
1E4E E9 EF F5
1E51 F3 A0 A0
1E54 A0 A0 A8
1E57 CE A9 E5
1E5A F8 F4 A0
1E5D A0 A0 A0
1E60 A8 D1 A9
1E63 F5 E9 F4
1E66 18 63      246      hex 1863
1E68 50          247      byt RTNCMD
1E69          248      ;
1E69 20 16 1D   249      jsr DOZCREAD
1E6C          250      ;
1E6C A2 0B      251      ldx #ZCOPTNS-1
1E6E 8E 9D 02   252      stx NUMSELC
1E71          253      ;
1E71 20 D7 1E   254      ^1    jsr ZCDISP
1E74          255      ;
1E74 CE 9D 02   256      dec NUMSELC
1E77 10 F8      257      bpl <1
1E79          258      ;
1E79 A2 00      259      ^2    ldx #ZERO
1E7B          260      ;
1E7B          261      ;
1E7B 8E 9D 02   262      ZCLOOP  stx NUMSELC

```

```

1E7E      263 ;
1E7E 20 D7 1E 264      jsr ZCDISP
1E81      265 ;
1E81 20 49 1F 266 ^3      jsr GETKEY
1E84      267 ;
1E84 AE 9D 02 268      ldx NUMSELC
1E87      269 ;
1E87 C9 88    270      cmp #LARROW
1E89 F0 04    271      beq >4
1E8B      272 ;
1E8B C9 8B    273      cmp #UARROW
1E8D D0 07    274      bne >5
1E8F      275 ;
1E8F CA      276 ^4      dex
1E90 10 E9    277      bpl ZCLOOP
1E92      278 ;
1E92 A2 0B    279      ldx #ZCOPTNS-1
1E94 D0 E5    280      bne ZCLOOP          ; always taken
1E96      281 ;
1E96 C9 95    282 ^5      cmp #RARROW
1E98 F0 08    283      beq >6
1E9A      284 ;
1E9A C9 8A    285      cmp #DARROW
1E9C F0 04    286      beq >6
1E9E      287 ;
1E9E C9 8D    288      cmp #RETURN
1EA0 D0 07    289      bne >7
1EA2      290 ;
1EA2 E8      291 ^6      inx
1EA3      292 ;
1EA3 E0 0C    293      cpx #ZCOPTNS
1EA5 D0 D4    294      bne ZCLOOP
1EA7      295 ;
1EA7 F0 D0    296      beq <2          ; always taken
1EA9      297 ;
1EA9 C9 D1    298 ^7      cmp #"Q"
1EAB D0 03    299      bne >8
1EAD      300 ;
1EAD 4C 44 1D 301      jmp DOZCSAVE
1EB0      302 ;
1EB0 C9 D0    303 ^8      cmp #"P"
1EB2 D0 05    304      bne >9
1EB4      305 ;
1EB4 FE D0 02 306      inc ZCSETBL,X
1EB7 D0 07    307      bne >1          ; always taken
1EB9      308 ;
1EB9 C9 CE    309 ^9      cmp #"N"
1EBB D0 C4    310      bne <3
1EBD      311 ;
1EBD DE D0 02 312      dec ZCSETBL,X
1EC0      313 ;
1EC0 E0 01    314 ^1      cpx #1
1EC2 D0 B7    315      bne ZCLOOP
1EC4      316 ;
1EC4 AD D1 02 317      lda ZCSETBL+1
1EC7 10 02    318      bpl >2
1EC9      319 ;
1EC9 A9 03    320      lda #ZCNSPEED-1
1ECB      321 ;
1ECB C9 04    322 ^2      cmp #ZCNSPEED
1ECD 90 02    323      bcc >3

```

```

1ECF          324 ;
1ECF A9 00    325      lda #ZERO
1ED1          326 ;
1ED1 8D D1 02 327 ^3      sta ZCSETBL+1
1ED4          328 ;
1ED4 4C 7B 1E 329      jmp ZCLOOP
1ED7          330 ;
1ED7          331 ;
1ED7 20 C8 20 332 ZCDISP    jsr PRINT
1EDA 18      333      hex 18
1EDB A0 A0    334      asc "  "
1EDD 50      335      byt RTNCMD
1EDE          336 ;
1EDE AE 9D 02 337      ldx NUMSELC
1EE1          338 ;
1EE1 BD 71 25 339      lda LINETBL,X
1EE4 85 25    340      sta CV
1EE6          341 ;
1EE6 20 20 21 342      jsr DOVTAB
1EE9          343 ;
1EE9 20 C8 20 344      jsr PRINT
1EEC 18      345      hex 18
1EED AD BE A0 346      asc "-> "
1EF0 50      347      byt RTNCMD
1EF1          348 ;
1EF1 AC 9D 02 349      ldy NUMSELC
1EF4          350 ;
1EF4 B9 D0 02 351      lda ZCSETBL,Y
1EF7          352 ;
1EF7 C0 00    353      cpy #ZERO
1EF9 D0 08    354      bne >1
1EFB          355 ;
1EFB 29 01    356      and #1
1EFD F0 25    357      beq >4
1EFF          358 ;
1EFF A0 03    359      ldy #ZCTEXT1-ZCTEXT
1F01 D0 21    360      bne >4 ; always taken
1F03          361 ;
1F03 C0 01    362 ^1      cpy #1
1F05 D0 07    363      bne >2
1F07          364 ;
1F07 0A      365      asl
1F08 0A      366      asl
1F09          367 ;
1F09 69 06    368      adc #ZCTEXT2-ZCTEXT
1F0B A8      369      tay
1F0C          370 ;
1F0C D0 16    371      bne >4 ; always taken
1F0E          372 ;
1F0E C0 02    373 ^2      cpy #2
1F10 D0 0A    374      bne >3
1F12          375 ;
1F12 A0 16    376      ldy #ZCTEXT3-ZCTEXT
1F14          377 ;
1F14 29 01    378      and #1
1F16 F0 0C    379      beq >4
1F18          380 ;
1F18 A0 1E    381      ldy #ZCTEXT4-ZCTEXT
1F1A D0 08    382      bne >4 ; always taken
1F1C          383 ;
1F1C A0 26    384 ^3      ldy #ZCTEXT5-ZCTEXT

```

1F1E		385	;	
1F1E	29 01	386		and #1
1F20	F0 02	387		beq >4
1F22		388	;	
1F22	A0 2C	389		ldy #ZCTEXT6-ZCTEXT
1F24		390	;	
1F24	B9 44 26	391	^4	lda ZCTEXT,Y
1F27	48	392		pha
1F28		393	;	
1F28	09 80	394		ora #\$80
1F2A	20 3B 21	395		jsr DOCOUT
1F2D		396	;	
1F2D	C8	397		iny
1F2E		398	;	
1F2E	68	399		pla
1F2F	30 F3	400		bmi <4
1F31		401	;	
1F31	60	402		rts
1F32		403	;	
1F32		404	;	
1F32		405		icl "SUBS.L"

LLOAD SUBS.L,A\$4000


```

1F32          1          ttl "EOS+ Source Code, SUBS.L"
1F32          2          ;
1F32          3          ;
1F32          4          ; SUBS.L
1F32          5          ;
1F32          6          ;
1F32 38        7          EOSBELL  sec
1F33          8          ;
1F33 A9 80     9          lda #$80
1F35          10         ;
1F35 A0 80    11         ^1      ldy #$80
1F37          12         ;
1F37 88       13         ^2      dey
1F38 D0 FD    14         bne <2
1F3A          15         ;
1F3A 2C 30 C0 16         bit SPKR
1F3D          17         ;
1F3D E9 01    18         sbc #1
1F3F D0 F4    19         bne <1
1F41          20         ;
1F41 60       21         rts
1F42          22         ;
1F42          23         ;
1F42 A9 60    24         RDKEY   lda #$60
1F44 20 3B 21 25         jsr DOCOUT
1F47          26         ;
1F47 C6 24    27         dec CH
1F49          28         ;
1F49 AD 00 C0 29         GETKEY  lda KEY
1F4C          30         ;
1F4C EA       31         nop
1F4D          32         ;
1F4D 10 FA    33         bpl GETKEY
1F4F          34         ;
1F4F 2C 10 C0 35         bit CLRKEY
1F52          36         ;
1F52 C9 FF    37         cmp #NEGONE
1F54 D0 02    38         bne >1
1F56          39         ;
1F56 A9 88    40         lda #LARROW
1F58          41         ;
1F58 C9 E0    42         ^1      cmp #LWRCASE
1F5A 90 02    43         bcc >2
1F5C          44         ;
1F5C 29 DF    45         and #LWRMASK
1F5E          46         ;
1F5E 60       47         ^2      rts
1F5F          48         ;
1F5F          49         ;
1F5F C9 B0    50         GETHEX  cmp #"0"
1F61 90 19    51         bcc >3
1F63          52         ;
1F63 C9 BA    53         cmp #"9"+1
1F65 90 08    54         bcc >1
1F67          55         ;
1F67 C9 C1    56         cmp #"A"
1F69 90 11    57         bcc >3
1F6B          58         ;
1F6B C9 C7    59         cmp #"F"+1
1F6D B0 0D    60         bcs >3

```

```

1F6F          61 ;
1F6F 48       62 ^1 pha
1F70          63 ;
1F70 C9 BA    64 cmp #"9"+1
1F72 90 02    65 bcc >2
1F74          66 ;
1F74 E9 07    67 sbc #7
1F76          68 ;
1F76 29 0F    69 ^2 and #VALUMASK
1F78 AA       70 tax
1F79          71 ;
1F79 68       72 pla
1F7A          73 ;
1F7A 18       74 clc
1F7B          75 ;
1F7B 60       76 rts
1F7C          77 ;
1F7C 38       78 ^3 sec
1F7D          79 ;
1F7D 60       80 rts
1F7E          81 ;
1F7E          82 ;
1F7E          83 GETNUM:
1F7E 20 42 1F 84 ^1 jsr RDKEY
1F81          85 ;
1F81 C9 8D    86 cmp #RETURN
1F83 F0 23    87 beq >2
1F85          88 ;
1F85 C9 88    89 cmp #LARROW
1F87 F0 21    90 beq >3
1F89          91 ;
1F89 20 5F 1F 92 jsr GETHEX
1F8C B0 F0    93 bcs <1
1F8E          94 ;
1F8E 20 3B 21 95 jsr DOCOUT
1F91          96 ;
1F91 06 2E    97 asl DSTPTR
1F93 26 2F    98 rol DSTPTR+1
1F95 06 2E    99 asl DSTPTR
1F97 26 2F   100 rol DSTPTR+1
1F99 06 2E   101 asl DSTPTR
1F9B 26 2F   102 rol DSTPTR+1
1F9D 06 2E   103 asl DSTPTR
1F9F 26 2F   104 rol DSTPTR+1
1FA1          105 ;
1FA1 8A       106 txa
1FA2 05 2E    107 ora DSTPTR
1FA4 85 2E    108 sta DSTPTR
1FA6          109 ;
1FA6 A9 01    110 lda #1
1FA8          111 ;
1FA8 18       112 ^2 clc
1FA9          113 ;
1FA9 60       114 rts
1FAA          115 ;
1FAA 20 36 21 116 ^3 jsr DOSPACE
1FAD          117 ;
1FAD C6 24    118 dec CH
1FAF C6 24    119 dec CH
1FB1          120 ;
1FB1 38       121 sec

```

```

1FB2          122 ;
1FB2 60        123      rts
1FB3          124 ;
1FB3          125 ;
1FB3          126 GETVAL:
1FB3 A2 00     127 ^1    ldx #ZERO
1FB5          128 ;
1FB5 20 42 1F  129 ^2    jsr RDKEY
1FB8          130 ;
1FB8 C9 88     131      cmp #LARROW
1FBA D0 0C     132      bne >4
1FBC          133 ;
1FBC 20 36 21  134 ^3    jsr DOSPACE
1FBF          135 ;
1FBF C6 24     136      dec CH
1FC1          137 ;
1FC1 CA        138      dex
1FC2 30 EF     139      bmi <1
1FC4          140 ;
1FC4 C6 24     141      dec CH
1FC6 10 ED     142      bpl <2
1FC8          143 ;
1FC8 C9 8D     144 ^4    cmp #RETURN
1FCA F0 1C     145      beq >5
1FCC          146 ;
1FCC C9 B0     147      cmp #"0"
1FCE 90 E5     148      bcc <2
1FD0          149 ;
1FD0 C9 BA     150      cmp #"9"+1
1FD2 B0 E1     151      bcs <2
1FD4          152 ;
1FD4 20 3B 21  153      jsr DOCOUT
1FD7          154 ;
1FD7 29 0F     155      and #VALUMASK
1FD9 9D 9C 02  156      sta NUMIN,X
1FDC          157 ;
1FDC E8        158      inx
1FDD          159 ;
1FDD E0 03     160      cpx #3
1FDF D0 D4     161      bne <2
1FE1          162 ;
1FE1 20 42 1F  163      jsr RDKEY
1FE4          164 ;
1FE4 C9 88     165      cmp #LARROW
1FE6 F0 D4     166      beq <3
1FE8          167 ;
1FE8 CA        168 ^5    dex
1FE9 30 14     169      bmi >8
1FEB          170 ;
1FEB A0 00     171      ldy #ZERO
1FED          172 ;
1FED 18        173      clc
1FEE          174 ;
1FEE BD 9C 02  175      lda NUMIN,X
1FF1          176 ;
1FF1 CA        177 ^6    dex
1FF2 30 0B     178      bmi >8
1FF4          179 ;
1FF4 C8        180      iny
1FF5          181 ;
1FF5 DE 9C 02  182 ^7    dec NUMIN,X

```

```

1FF8 30 F7      183      bmi <6
1FFA           184      ;
1FFA 79 7D 25   185      adc DECTBLL,Y
1FFD 90 F6      186      bcc <7
1FFF           187      ;
1FFF 60         188      ^8      rts
2000           189      ;
2000           190      ;
2000           191      ; Fall into PRTSDV.
2000           192      ;
2000           193      EDITSDV:
2000 20 C8 20   194      ^1      jsr PRINT
2003 00 76      195      hex 0076
2005 C5 EE F4   196      asc "Enter Slot:  "
2008 E5 F2 A0
200B D3 EC EF
200E F4 BA A0
2011 A0
2012 57         197      byt EOLCMD
2013 50         198      byt RTNCMD
2014           199      ;
2014 20 B3 1F   200      jsr GETVAL
2017 B0 0C      201      bcs >2
2019           202      ;
2019 A8         203      tay
201A F0 E4      204      beq <1
201C           205      ;
201C C9 08      206      cmp #7+1
201E B0 E0      207      bcs <1
2020           208      ;
2020 85 EB      209      sta MSLOT
2022           210      ;
2022 20 68 20   211      jsr PRTSDV
2025           212      ;
2025 20 C8 20   213      ^2      jsr PRINT
2028 00 76      214      hex 0076
202A C5 EE F4   215      asc "Enter Drive:  "
202D E5 F2 A0
2030 C4 F2 E9
2033 F6 E5 BA
2036 A0 A0
2038 57         216      byt EOLCMD
2039 50         217      byt RTNCMD
203A           218      ;
203A 20 B3 1F   219      jsr GETVAL
203D B0 0C      220      bcs >3
203F           221      ;
203F A8         222      tay
2040 F0 E3      223      beq <2
2042           224      ;
2042 C9 52      225      cmp #81+1
2044 B0 DF      226      bcs <2
2046           227      ;
2046 85 EC      228      sta DRIVE
2048           229      ;
2048 20 68 20   230      jsr PRTSDV
204B           231      ;
204B 20 C8 20   232      ^3      jsr PRINT
204E 00 76      233      hex 0076
2050 C5 EE F4   234      asc "Enter Volume:  "
2053 E5 F2 A0

```

```

2056 D6 EF EC
2059 F5 ED E5
205C BA A0 A0
205F 57          235          byt EOLCMD
2060 50          236          byt RTNCMD
2061          237          ;
2061 20 B3 1F    238          jsr GETVAL
2064 B0 39      239          bcs RTN01
2066          240          ;
2066 85 ED      241          sta VOLUME
2068          242          ;
2068          243          ;
2068          244          ; Fall into PRTDEC.
2068          245          ;
2068 20 C8 20    246 PRTSDV   jsr PRINT
206B 00 71      247          hex 0071          ; position cursor on line 17
206D 50          248          byt RTNCMD
206E          249          ;
206E A2 01      250          ldx #1          ; 1 digit
2070 A0 04      251          ldy #4          ; CH
2072 A5 EB      252          lda MSLOT
2074          253          ;
2074 20 86 20    254          jsr PRTDEC
2077          255          ;
2077 A2 02      256          ldx #2          ; 2 digits
2079 A0 08      257          ldy #8          ; CH
207B A5 EC      258          lda DRIVE
207D          259          ;
207D 20 86 20    260          jsr PRTDEC
2080          261          ;
2080 A2 03      262          ldx #3          ; 3 digits
2082 A0 0D      263          ldy #13         ; CH
2084 A5 ED      264          lda VOLUME
2086          265          ;
2086          266          ;
2086 84 24      267 PRTDEC   sty CH
2088          268          ;
2088 A0 B0      269 ^1      ldy #"0"
208A          270          ;
208A 38          271          sec
208B          272          ;
208B FD 7C 25    273 ^2      sbc DECTBLL-1,X
208E 90 03      274          bcc >3
2090          275          ;
2090 C8          276          iny
2091 D0 F8      277          bne <2
2093          278          ;
2093 7D 7C 25    279 ^3      adc DECTBLL-1,X
2096 48          280          pha
2097          281          ;
2097 98          282          tya
2098 20 3B 21    283          jsr DOCOUT
209B          284          ;
209B 68          285          pla
209C          286          ;
209C CA          287          dex
209D D0 E9      288          bne <1
209F          289          ;
209F 60          290 RTN01   rts
20A0          291          ;
20A0          292          ;

```

```

20A0          293 ; Find the slot number of the next EPROM card from SLOTMAP.
20A0          294 ; If the slot number exceeds slot 7, start over at slot 1.
20A0          295 ; The value in SLOTMAP must be greater than 1 to be valid.
20A0          296 ; Return with the slot page value of 0xCs in the A-reg.
20A0          297 ;
20A0 AE 91 02 298 NEXTMAP ldx EPSLOT
20A3          299 ;
20A3 E8       300 ^1      inx
20A4          301 ;
20A4 E0 08    302         cpx #8
20A6 D0 02    303         bne >2
20A8          304 ;
20A8 A2 01    305         ldx #1
20AA          306 ;
20AA AD 92 02 307 ^2      lda SLOTMAP
20AD          308 ;
20AD 3D 80 25 309         and MAPMASKS,X
20B0 F0 F1    310         beq <1
20B2          311 ;
20B2 8A       312         txa
20B3 09 C0    313         ora /PAGEC0
20B5          314 ;
20B5 A0 D0    315         ldy #EPMAPEOS
20B7          316 ;
20B7 60       317         rts
20B8          318 ;
20B8          319 ;
20B8 18       320 CLRUSER clc
20B9          321 ;
20B9 B0 00    322         bcs *+2
20BB          323         dfs !-1
20BA          324 ;
20BA          325 ;
20BA 38       326 SETUSER sec
20BB          327 ;
20BB 84 07    328         sty YREG
20BD 85 08    329         sta AREG
20BF          330 ;
20BF AC F6 BF 331         ldy MNGUSER
20C2 AD F7 BF 332         lda MNGUSER+1
20C5          333 ;
20C5 4C 49 21 334         jmp JSRMEM
20C8          335 ;
20C8          336 ;
20C8 68       337 PRINT   pla
20C9 85 FC    338         sta PRNTPTR
20CB 68       339         pla
20CC 85 FD    340         sta PRNTPTR+1
20CE          341 ;
20CE E6 FC    342 ^1      inc PRNTPTR
20D0 D0 02    343         bne >2
20D2          344 ;
20D2 E6 FD    345         inc PRNTPTR+1
20D4          346 ;
20D4 A0 00    347 ^2      ldy #ZERO
20D6          348 ;
20D6 B1 FC    349         lda (PRNTPTR),Y
20D8 10 06    350         bpl >3
20DA          351 ;
20DA 20 3B 21 352         jsr DOCOUT
20DD          353 ;

```

```

20DD 4C CE 20      354      jmp <1
20E0              355      ;
20E0 C9 50        356      ^3      cmp #MAXCH
20E2 B0 04        357      bcs >4
20E4              358      ;
20E4 85 24        359      sta CH
20E6              360      ;
20E6 90 E6        361      bcc <1          ; always taken
20E8              362      ;
20E8 20 EE 20     363      ^4      jsr PRINT01
20EB              364      ;
20EB 4C CE 20     365      jmp <1
20EE              366      ;
20EE C9 60        367      PRINT01  cmp #MINCV
20F0 90 06        368      bcc >5
20F2              369      ;
20F2 29 1F        370      and #CVMASK
20F4 85 25        371      sta CV
20F6              372      ;
20F6 A9 56        373      lda #TABVCMD
20F8              374      ;
20F8 29 0F        375      ^5      and #PRNTMASK
20FA AA          376      tax
20FB D0 0D        377      bne >6
20FD              378      ;
20FD BA          379      tsx
20FE              380      ;
20FE A5 FC        381      lda PRNTPTR
2100 9D 01 01     382      sta STACK+1,X
2103 A5 FD        383      lda PRNTPTR+1
2105 9D 02 01     384      sta STACK+2,X
2108              385      ;
2108 18          386      clc
2109              387      ;
2109 60          388      rts
210A              389      ;
210A E0 09        390      ^6      cpx #CNTRCMD&PRNTMASK
210C D0 31        391      bne DOJSRMEM
210E              392      ;
210E A9 9F        393      lda #SPACE-1
2110              394      ;
2110 C8          395      ^7      iny
2111              396      ;
2111 D1 FC        397      cmp (PRNTPTR),Y
2113 90 FB        398      bcc <7
2115              399      ;
2115 98          400      tya
2116              401      ;
2116 49 FF        402      eor #NEGONE
2118 65 21        403      adc WNDWIDTH
211A              404      ;
211A 4A          405      lsr
211B              406      ;
211B 65 20        407      adc WNDLFT
211D 85 24        408      sta CH
211F              409      ;
211F 60          410      rts
2120              411      ;
2120              412      ;
2120 A2 06        413      DOVTAB  ldx #6
2122 D0 1B        414      bne DOJSRMEM          ; always taken

```

```

2124          415 ;
2124          416 ;
2124 A2 09      417 DOSHOOK   ldx #9
2126 D0 17      418          bne DOJSRMEM          ; always taken
2128          419 ;
2128          420 ;
2128 86 06      421 DOPRNTAX  stx XREG
212A          422 ;
212A A2 0A      423          ldx #10
212C D0 11      424          bne DOJSRMEM          ; always taken
212E          425 ;
212E          426 ;
212E A2 0B      427 DOPRBYTE  ldx #11
2130 D0 0D      428          bne DOJSRMEM          ; always taken
2132          429 ;
2132          430 ;
2132 A2 0C      431 DOPRHEX   ldx #12
2134 D0 09      432          bne DOJSRMEM          ; always taken
2136          433 ;
2136          434 ;
2136 A9 A0      435 DOSPACE   lda #SPACE
2138          436 ;
2138 2C 00 00    437          bit *-*
213B          438          dfs !-2
2139          439 ;
2139 A9 8D      440 DOCROUT   lda #RETURN
213B          441 ;
213B 86 06      442 DOCOUT    stx XREG
213D          443 ;
213D A2 00      444          ldx #ZERO
213F          445 ;
213F          446 ;
213F          447 ; Save Y-reg and A-reg; use X-reg to load routine address.
213F          448 ; Fall into JSRMEM.
213F          449 ;
213F 84 07      450 DOJSRMEM  sty YREG
2141 85 08      451          sta AREG
2143          452 ;
2143 BC 57 25    453          ldy ADDRBTBLL,X
2146 BD 64 25    454          lda ADDRBTBLH,X
2149          455 ;
2149          456 ;
2149          457 ; Y-reg and A-reg values need to be saved at YREG and AREG
2149          458 ; before loading Y-reg and A-reg with address of routine
2149          459 ; to execute outside of the EPROM card. The X-reg value
2149          460 ; is safe and does not need to be saved.
2149          461 ;
2149 8C A6 02     462 JSRMEM    sty MEMJMP
214C 8D A7 02     463          sta MEMJMP+1
214F          464 ;
214F 4C 4C 01     465          jmp EPJSR
2152          466 ;
2152          467 ;
2152          468 ; Load DOS 4.5L into Main memory.
2152          469 ;
2152 A0 04         470 LOADOSL   ldy #DOSLPRMS-CATALOG
2154 20 85 22     471          jsr COPYPRM0
2157          472 ;
2157 AD 82 C0     473          lda ROM2WP
215A          474 ;
215A A0 00       475          ldy #ZERO

```



```

215C A9 21      476      lda /PAGEC0-PAGE9F
215E           477      ;
215E 84 2C      478      sty LENPTR
2160 85 2D      479      sta LENPTR+1
2162           480      ;
2162 A0 00      481      ld y #PAGE9F
2164 A9 9F      482      lda /PAGE9F
2166           483      ;
2166 84 2E      484      sty DSTPTR
2168 85 2F      485      sta DSTPTR+1
216A           486      ;
216A 4C 1D 01   487      jmp EPMOVE
216D           488      ;
216D           489      ;
216D           490      ; Load DOS 4.5H into the Language card and Main memory.
216D           491      ;
216D A0 15      492      LOADOSH ld y #DOSHPRMS-CATALOG
216F 20 85 22   493      jsr COPYPRM0
2172           494      ;
2172           495      ;
2172           496      ; Load Language Card Bank 2.
2172           497      ;
2172 AD 81 C0    498      lda ROM2WE
2175 AD 81 C0    499      lda ROM2WE
2178           500      ;
2178 A0 00      501      ld y #ZERO
217A A9 1A      502      lda /PAGEEA-PAGED0
217C           503      ;
217C 84 2C      504      sty LENPTR
217E 85 2D      505      sta LENPTR+1
2180           506      ;
2180 A0 00      507      ld y #PAGED0
2182 A9 D0      508      lda /PAGED0
2184           509      ;
2184 20 AB 21    510      jsr LOADOSH2
2187           511      ;
2187           512      ;
2187           513      ; Load Language Card Bank 1.
2187           514      ;
2187 AD 89 C0    515      lda ROM1WE
218A AD 89 C0    516      lda ROM1WE
218D           517      ;
218D A0 00      518      ld y #ZERO
218F A9 0E      519      lda /PAGEDE-PAGED0
2191           520      ;
2191 84 2C      521      sty LENPTR
2193 85 2D      522      sta LENPTR+1
2195           523      ;
2195 A0 00      524      ld y #PAGED0
2197 A9 D0      525      lda /PAGED0
2199           526      ;
2199 20 AB 21    527      jsr LOADOSH2
219C           528      ;
219C           529      ;
219C           530      ; Write protect Language card and load remaining memory.
219C           531      ;
219C AD 82 C0    532      lda ROM2WP
219F           533      ;
219F A0 00      534      ld y #ZERO
21A1 A9 02      535      lda /PAGEC0-PAGEBE
21A3           536      ;

```

```

21A3 84 2C      537          sty LENPTR
21A5 85 2D      538          sta LENPTR+1
21A7           539          ;
21A7 A0 00      540          ldy #PAGEBE
21A9 A9 BE      541          lda /PAGEBE
21AB           542          ;
21AB 84 2E      543 LOADOSH2 sty DSTPTR
21AD 85 2F      544          sta DSTPTR+1
21AF           545          ;
21AF 4C 1D 01   546          jmp EPMOVE
21B2           547          ;
21B2           548          ;
21B2           549          ; Load F8 ROM into the Language card.
21B2           550          ;
21B2 A2 00      551 COPYROM ldx #ZERO
21B4 8E 94 02   552          stx EPNMBR
21B7           553          ;
21B7 BD BC 08   554 ^1          lda ROMPARMS+1,X
21BA 95 2A      555          sta SRCPTR,X
21BC           556          ;
21BC E8         557          inx
21BD           558          ;
21BD E0 06      559          cpx #PARMSIZE
21BF D0 F6      560          bne <1
21C1           561          ;
21C1 AD 81 C0   562          lda ROM2WE
21C4 AD 81 C0   563          lda ROM2WE
21C7           564          ;
21C7 A2 00      565          ldx #ZERO
21C9           566          ;
21C9           567          .if HWCARD
21C9 A9 80      568          lda #EPOFFVAL
21CB           569          .el
21CB           570          lda #QLOFFVAL
21CB           571          .fi
21CB           572          ;
21CB 20 20 01   573          jsr EPMOVE2
21CE           574          ;
21CE AD 82 C0   575          lda ROM2WP
21D1           576          ;
21D1 60         577          rts
21D2           578          ;
21D2           579          ;
21D2           580          ; Initializes the pointers to begin reading an EPROM's
21D2           581          ; catalog. The four catalog sync bytes are verified
21D2           582          ; first.
21D2           583          ;
21D2 A0 CA      584 INITCAT ldy #CATPARMS-CATALOG
21D4 20 8A 22   585          jsr COPYPARM
21D7           586          ;
21D7 20 B2 22   587          jsr READBLK
21DA           588          ;
21DA A0 03      589          ldy #SYNC.L-1
21DC           590          ;
21DC B9 AC 02   591 ^1          lda SYNCBUFR,Y
21DF D9 76 26   592          cmp SYNCBYTS,Y
21E2 D0 05      593          bne >2
21E4           594          ;
21E4 88         595          dey
21E5 10 F5      596          bpl <1
21E7           597          ;

```

```

21E7 18          598          clc
21E8            599          ;
21E8 60          600          rts
21E9            601          ;
21E9 38          602          ^2      sec
21EA            603          ;
21EA 60          604          rts
21EB            605          ;
21EB            606          ;
21EB            607          ; Gets the catalog entry currently pointed to.  Points to
21EB            608          ; the next entry on exit.  Assumes INITCAT was previously
21EB            609          ; called.  FILEENTRY+1 points to FILETYPE, thus ENTRYLEN is
21EB            610          ; reduced by one byte.
21EB            611          ;
21EB A5 2A       612  GETENTRY  lda  SRCPTR
21ED 48          613          pha
21EE A5 2B       614          lda  SRCPTR+1
21F0 48          615          pha
21F1            616          ;
21F1 A0 1F       617          ldy  #ENTRYLEN-1
21F3 A9 00       618          lda  /ENTRYLEN-1
21F5            619          ;
21F5 84 2C       620          sty  LENPTR
21F7 85 2D       621          sta  LENPTR+1
21F9            622          ;
21F9 A0 B1       623          ldy  #FILEENTRY+1
21FB A9 02       624          lda  /FILEENTRY+1
21FD            625          ;
21FD 84 2E       626          sty  DSTPTR
21FF 85 2F       627          sta  DSTPTR+1
2201            628          ;
2201 20 1D 01    629          jsr  EPMOVE
2204            630          ;
2204 68          631          pla
2205 85 2B       632          sta  SRCPTR+1
2207 68          633          pla
2208 85 2A       634          sta  SRCPTR
220A            635          ;
220A AD 94 02    636          lda  EPNMBR
220D 8D B0 02    637          sta  FILEPNUM
2210            638          ;
2210 A0 00       639          ldy  #ZERO
2212 8C 9E 02    640          sty  FLENGTH
2215            641          ;
2215 B9 B8 02    642          ^1      lda  FILENAME,Y
2218 48          643          pha
2219            644          ;
2219 09 80       645          ora  #MSBSET
221B 99 B8 02    646          sta  FILENAME,Y
221E            647          ;
221E EE 9E 02    648          inc  FLENGTH
2221            649          ;
2221 68          650          pla
2222 30 07       651          bmi  >2
2224            652          ;
2224 C8          653          iny
2225            654          ;
2225 C0 18       655          cpy  #NAME SIZE
2227 D0 EC       656          bne  <1
2229            657          ;
2229 F0 0C       658          beq  >4

```

```

222B          659 ;
222B A9 A0    660 ^2      lda #SPACE
222D          661 ;
222D C8       662 ^3      iny
222E          663 ;
222E C0 18    664          cpy #NAME SIZE
2230 F0 05    665          beq >4
2232          666 ;
2232 99 B8 02  667          sta FILENAME,Y
2235          668 ;
2235 D0 F6     669          bne <3          ; always taken
2237          670 ;
2237 18        671 ^4      clc
2238          672 ;
2238 A9 07      673          lda #PARMSIZE+1
223A 6D 9E 02  674          adc FLENGTH
223D          675 ;
223D 65 2A     676          adc SRCPTR
223F 85 2A     677          sta SRCPTR
2241 90 02     678          bcc >5
2243          679 ;
2243 E6 2B     680          inc SRCPTR+1
2245          681 ;
2245 60         682 ^5      rts
2246          683 ;
2246          684 ;
2246          685 ; Begin with EPSTRT and locate its Catalog.
2246          686 ;
2246 86 CE      687 FINDFILE stx GENPTR
2248 85 CF      688          sta GENPTR+1
224A          689 ;
224A AD 98 02  690          lda EPSTRT
224D 8D 94 02  691          sta EPNMBR
2250          692 ;
2250 20 D2 21  693 ^1      jsr INITCAT
2253 B0 23     694          bcs >4
2255          695 ;
2255          696 ;
2255          697 ; Get a Catalog entry.  If FILETYPE is zero, at ENDCAT.
2255          698 ;
2255 20 EB 21  699 ^2      jsr GETENTRY
2258          700 ;
2258 AD B1 02  701          lda FILETYPE
225B F0 1B     702          beq >4
225D          703 ;
225D          704 ;
225D          705 ; Compare filename lengths.
225D          706 ;
225D AD 9E 02  707          lda FLENGTH
2260 CD A3 02  708          cmp FILELEN
2263 D0 F0     709          bne <2
2265          710 ;
2265          711 ;
2265          712 ; Compare filenames.
2265          713 ;
2265 A0 00      714          ldy #ZERO
2267          715 ;
2267 B1 CE      716 ^3      lda (GENPTR),Y
2269 09 80     717          ora #MSBSET
226B          718 ;
226B D9 B8 02  719          cmp FILENAME,Y

```

```

226E D0 E5      720      bne <2
2270           721      ;
2270 C8         722      iny
2271           723      ;
2271 CC A3 02   724      cpy FILELEN
2274 D0 F1     725      bne <3
2276           726      ;
2276           727      ;
2276           728      ; Found a matching filename.
2276           729      ;
2276 18         730      clc
2277           731      ;
2277 60         732      rts
2278           733      ;
2278           734      ;
2278           735      ; Check next EPROM.
2278           736      ;
2278 EE 94 02   737      ^4      inc EPNMBR
227B           738      ;
227B AD 99 02   739      lda EPEND
227E CD 94 02   740      cmp EPNMBR
2281 B0 CD      741      bcs <1
2283           742      ;
2283           743      ;
2283           744      ; Filename not found.
2283           745      ;
2283 38         746      sec
2284           747      ;
2284 60         748      rts
2285           749      ;
2285           750      ;
2285           751      ; Copy the file's parameter block pointed to by Y-reg.
2285           752      ; Fall into SELCBANK.
2285           753      ;
2285 A9 00      754      COPYPRM0 lda #ZERO
2287 8D 94 02   755      sta EPNMBR
228A           756      ;
228A A2 00     757      COPYPARM ldx #ZERO
228C           758      ;
228C B9 00 08   759      ^1      lda CATALOG,Y
228F 9D B1 02   760      sta FILETYPE,X
2292           761      ;
2292 C8         762      iny
2293 E8         763      inx
2294           764      ;
2294 E0 07      765      cpx #PARMSIZE+1
2296 D0 F4     766      bne <1
2298           767      ;
2298           768      ;
2298           769      ; Extract the EPROM bank from the EPROM offset address.
2298           770      ;
2298 AC B2 02   771      SELCBANK ldy SRCVAL
229B AD B3 02   772      lda SRCVAL+1
229E           773      ;
229E 4A        774      lsr
229F 4A        775      lsr
22A0 4A        776      lsr
22A1 4A        777      lsr
22A2 4A        778      lsr
22A3           779      ;
22A3 8D 95 02   780      sta EPBANK

```

```

22A6          781 ;
22A6 AD B3 02 782      lda SRCVAL+1
22A9 29 1F    783      and #BANKMASK
22AB          784 ;
22AB          785      .if DEBUG
22AB 09 08    786      ora /PAGE08
22AD          787      .el
22AD          788      ora /PAGEE0
22AD          789      .fi
22AD          790 ;
22AD 84 2A    791      sty SRCPTR
22AF 85 2B    792      sta SRCPTR+1
22B1          793 ;
22B1 60       794      rts
22B2          795 ;
22B2          796 ;
22B2          797 ; Read a block of data from an EPROM.
22B2          798 ;
22B2 20 98 22 799 READBLK jsr SELCBANK
22B5          800 ;
22B5 AC B4 02 801      ldy LENVAL
22B8 AD B5 02 802      lda LENVAL+1
22BB          803 ;
22BB 84 2C    804      sty LENPTR
22BD 85 2D    805      sta LENPTR+1
22BF          806 ;
22BF AC B6 02 807      ldy DSTVAL
22C2 AD B7 02 808      lda DSTVAL+1
22C5          809 ;
22C5 84 2E    810      sty DSTPTR
22C7 85 2F    811      sta DSTPTR+1
22C9          812 ;
22C9 4C 1D 01 813      jmp EPMOVE
22CC          814 ;
22CC          815 ;
22CC          816 ; Read two bytes of address data from memory.
22CC          817 ;
22CC 84 2A    818 READADR sty SRCPTR
22CE 85 2B    819      sta SRCPTR+1
22D0          820 ;
22D0 A0 02    821      ldy #2
22D2 A9 00    822      lda #ZERO
22D4          823 ;
22D4 84 2C    824      sty LENPTR
22D6 85 2D    825      sta LENPTR+1
22D8          826 ;
22D8 A0 AA    827      ldy #ADDRBUFR
22DA A9 02    828      lda /ADDRBUFR
22DC          829 ;
22DC 84 2E    830      sty DSTPTR
22DE 85 2F    831      sta DSTPTR+1
22E0          832 ;
22E0 A2 00    833      ldx #ZERO
22E2          834 ;
22E2          835      .if HWCARD
22E2 A9 80    836      lda #EPOFFVAL
22E4          837      .el
22E4          838      lda #QLOFFVAL
22E4          839      .fi
22E4          840 ;
22E4 4C 20 01 841      jmp EPMOVE2

```

```
22E7          842  ;  
22E7          843  ;  
22E7          844      icl "PAGE1.L"
```

```
LLOAD PAGE1.L,A$4000
```

```
22E7          1          ttl "EOS+ Source Code, PAGE1.L"
22E7          2          ;
22E7          3          ;
22E7          4          ; PAGE1.L
22E7          5          ;
22E7          6          ;
22E7          7          ; Use the slot value in X-reg to prepare the slot*16 value
22E7          8          ; for the Device Select index for this EPROM card. Slot 3
22E7          9          ; ROM is already enabled.
22E7         10          ;
22E7 8A        11  BUILDMAP txa
22E8         12          ;
22E8 0A        13          asl
22E9 0A        14          asl
22EA 0A        15          asl
22EB 0A        16          asl
22EC         17          ;
22EC 8D A4 02  18          sta SLOT16
22EF AA        19          tax
22F0         20          ;
22F0         21          ;
22F0         22          ; Move the EPROM mapping code to the stack.
22F0         23          ;
22F0 A0 49     24          ldy #EPMAPLEN-1
22F2         25          ;
22F2 B9 18 23  26          ^1   lda MAPCODE,Y
22F5 99 10 01  27          sta MAPPAGE,Y
22F8         28          ;
22F8 88        29          dey
22F9 10 F7     30          bpl <1
22FB         31          ;
22FB         32          ;
22FB         33          ; Initialize the page-zero pointers used in slot mapping,
22FB         34          ; then make the map of the slots.
22FB         35          ;
22FB C8        36          iny
22FC 8C 92 02  37          sty SLOTMAP
22FF         38          ;
22FF A9 E1     39          lda /PAGEE1
2301         40          ;
2301 84 2E     41          sty DSTPTR
2303 85 2F     42          sta DSTPTR+1
2305         43          ;
2305 A9 C1     44          lda /PAGEC1
2307         45          ;
2307 84 2A     46          sty SRCPTR
2309 85 2B     47          sta SRCPTR+1
230B         48          ;
230B A0 B8     49          ldy #EPOFF          ; slot interface routine entry
230D         50          ;
230D 8C A8 02  51          sty SLOTJMP
2310 8D A9 02  52          sta SLOTJMP+1
2313         53          ;
2313         54          .if HWCARD
2313 A9 80     55          lda #EPOFFVAL
2315         56          .el
2315         57          lda #QLOFFVAL
2315         58          .fi
2315         59          ;
2315 4C 10 01  60          jmp MAPPAGE
```



```

2318      61 ;
2318      62 ;
2318      63 ; Mapping code for all EPROM cards. The slots that contain
2318      64 ; an EPROM card are recorded in SLOTMAP with its slot bit
2318      65 ; set to 1.
2318      66 ;
2318      67 MAPCODE:
2318      68 ;
2318      69      phs STKCODE
0110      70 ;
0110      71 MAPPAGE:
0110      72 ;
0110      73 ;
0110      74 ; Turn this EPROM card OFF.
0110      75 ;
0110 9D 80 C0      76      sta EPSELC,X
0113      77 ;
0113 A2 01      78      ldx #1
0115      79 ;
0115      80 ;
0115      81 ; If an EPROM card resides in this slot, turn that card ON.
0115      82 ;
0115 8E A5 02      83 ^1      stx SLOT
0118 8A      84      txa
0119      85 ;
0119 0A      86      asl
011A 0A      87      asl
011B 0A      88      asl
011C 0A      89      asl
011D      90 ;
011D AA      91      tax
011E      92 ;
011E A9 00      93      lda #EPONVAL
0120 9D 80 C0      94      sta EPSELC,X
0123      95 ;
0123      96 ;
0123      97 ; Test for an EPROM card.
0123      98 ;
0123 A0 F8      99      ldy #EPBINTXT
0125      100 ;
0125 B9 44 25     101 ^2      lda EPTEXT-EPBINTXT&NEGONE,Y
0128      102 ;
0128 D1 2A      103      cmp (SRCPTR),Y
012A D0 0C      104      bne >3
012C      105 ;
012C D1 2E      106      cmp (DSTPTR),Y
012E D0 08      107      bne >3
0130      108 ;
0130 C8      109      iny
0131 D0 F2      110      bne <2
0133      111 ;
0133      112 ;
0133      113 ; An EPROM card has been successfully found. Turn the
0133      114 ; EPROM card OFF and mark this slot as found.
0133      115 ;
0133 20 57 01     116      jsr EPOFFJMP
0136      117 ;
0136 38      118      sec
0137      119 ;
0137 90 00      120      bcc *+2
0139      121      dfs !-1

```

```

0138      122 ;
0138 18      123 ^3      clc
0139      124 ;
0139 6E 92 02 125      ror SLOTMAP
013C      126 ;
013C      127 ;
013C      128 ; Go to the next slot to test.
013C      129 ;
013C 2C FF CF 130      bit CLRROM
013F      131 ;
013F E6 2B      132      inc SRCPTR+1
0141 E6 2F      133      inc DSTPTR+1
0143      134 ;
0143 EE A9 02 135      inc SLOTJMP+1
0146      136 ;
0146 AE A5 02 137      ldx SLOT
0149      138 ;
0149 E8      139      inx
014A      140 ;
014A E0 08      141      cpx #8
014C D0 C7      142      bne <1
014E      143 ;
014E AE A4 02 144      ldx SLOT16
0151      145 ;
0151 A9 00      146      lda #EPONVAL
0153 9D 80 C0 147      sta EPSELC,X
0156      148 ;
0156 60      149      rts
0157      150 ;
0157      151 ;
0157      152 ; Use EPROM routine to turn EPROM card that is under test
0157      153 ; to OFF.
0157      154 ;
0157 6C A8 02 155 EPOFFJMP jmp (SLOTJMP)
015A      156 ;
015A      157 ;
004A      158 EPMAPLEN equ *-MAPPAGE
015A      159 ;
015A      160 ;
015A      161      phs MAPCODE+EPMAPLEN
2362      162 ;
2362      163 ;
2362      164 ; Use the slot value in X-reg to prepare the slot*16 value.
2362      165 ;
2362 8A      166 MOVEEPBM txa
2363      167 ;
2363 0A      168      asl
2364 0A      169      asl
2365 0A      170      asl
2366 0A      171      asl
2367      172 ;
2367 09 80      173      ora #EPSELC
2369 AA      174      tax
236A      175 ;
236A      176 ;
236A      177 ; Move the EPROM card bank management code to the stack.
236A      178 ;
236A A0 79      179      ldy #EPBMLEN-1
236C      180 ;
236C B9 85 23 181 ^1      lda EPBMCODE,Y
236F 99 10 01 182      sta EPBMPAGE,Y

```

```

2372      183 ;
2372 88      184      dey
2373 10 F7    185      bpl <1
2375      186 ;
2375      187 ;
2375      188 ; Modify the stack code for direct EPROM card access.
2375      189 ;
2375 8E 23 01 190      stx EPBMMOD1+1
2378 8E 4F 01 191      stx EPBMMOD2+1
237B 8E 5E 01 192      stx EPBMMOD3+1
237E      193 ;
237E 8E 69 01 194      stx EXECMOD1+1
2381 8E 74 01 195      stx EXECMOD2+1
2384      196 ;
2384 60      197      rts
2385      198 ;
2385      199 ;
2385      200 ; EPROM card bank management code.
2385      201 ;
2385      202 EPBMCODE:
2385      203 ;
2385      204      phs STKCODE
0110      205 ;
0110      206 EPBMPAGE:
0110      207 ;
0110      208 ;
0110      209 ; Configure the X-reg and A-reg based on EPNMBR and EPBANK.
0110      210 ;
0110      211 EPCONFIG:
0110      212      .if HWCARD
0110 AE 95 02 213      ldx EPBANK
0113 AD 94 02 214      lda EPNMBR
0116 60      215      rts
0117      216      dfs 6,NEGONE
011D      217      .el
011D      218      lda EPBANK
011D      219      lsr
011D      220      tax
011D      221      lda EPNMBR
011D      222      bcc >1
011D      223      ora #EPUSR
011D      224 ^1      rts
011D      225      .fi
011D      226 ;
011D      227 ;
011D      228 ; Move data from EPROM to memory, or memory to memory.
011D      229 ;
011D 20 10 01 230 EPMOVE    jsr EPCONFIG
0120      231 ;
0120 A0 00      232 EPMOVE2   ldy #ZERO
0122      233 ;
0122 9D 80 C0    234 EPBMMOD1  sta EPSELC,X
0125      235 ;
0125 A5 2C      236 ^1      lda LENPTR
0127 D0 06      237      bne >2
0129      238 ;
0129 A5 2D      239      lda LENPTR+1
012B F0 2D      240      beq EPRETURN
012D      241 ;
012D C6 2D      242      dec LENPTR+1
012F      243 ;

```

```

012F C6 2C      244 ^2      dec LENPTR
0131           245 ;
0131 B1 2A      246      lda (SRCPTR),Y
0133 91 2E      247      sta (DSTPTR),Y
0135           248 ;
0135 E6 2E      249      inc DSTPTR
0137 D0 02      250      bne >3
0139           251 ;
0139 E6 2F      252      inc DSTPTR+1
013B           253 ;
013B E6 2A      254 ^3      inc SRCPTR
013D D0 E6      255      bne <1
013F           256 ;
013F E6 2B      257      inc SRCPTR+1
0141 D0 E2      258      bne <1
0143           259 ;
0143 A9 E0      260      lda /PAGEE0
0145 85 2B      261      sta SRCPTR+1
0147           262 ;
0147 EE 95 02   263      inc EPBANK
014A D0 D1      264      bne EPMOVE          ; always taken
014C           265 ;
014C           266 ;
014C           267 ; Exit EOS+, execute routine, and resume EOS+.
014C           268 ;
014C           269      .if HWCARD
014C A9 80      270 EPJSR   lda #EPOFFVAL
014E           271      .el
014E           272 EPJSR   lda #QLOFFVAL
014E           273      .fi
014E           274 ;
014E 8D 80 C0   275 EPBMMOD2 sta EPSELC
0151           276 ;
0151 A6 06      277      ldx XREG
0153 A4 07      278      ldy YREG
0155 A5 08      279      lda AREG
0157           280 ;
0157 20 62 01   281      jsr DOMEMJMP
015A           282 ;
015A           283 ;
015A           284 ; Return to EPROM 0, bank 0.
015A           285 ;
015A 48         286 EPRETURN pha
015B           287 ;
015B A9 00      288      lda #EPONVAL
015D 8D 80 C0   289 EPBMMOD3 sta EPSELC
0160           290 ;
0160 68         291      pla
0161           292 ;
0161 60         293      rts
0162           294 ;
0162           295 ;
0162           296 ; Execute routine via MEMJMP.
0162           297 ;
0162 6C A6 02   298 DOMEMJMP jmp (MEMJMP)
0165           299 ;
0165           300 ;
0165           301 ; Exec management code.
0165           302 ;
0165           303 ; Read a single character from an EPROM until a NULL
0165           304 ; character is reached. Copy EPNMBR and EPBANK to

```

```

0165      305 ; page-zero in order to increase speed.
0165      306 ;
0165 20 10 01 307 EPEXEC jsr EPCONFIG
0168      308 ;
0168 9D 80 C0 309 EXECMOD1 sta EPSELC,X
016B      310 ;
016B A0 00 311 ldy #ZERO
016D      312 ;
016D B1 FA 313 lda (EXECPTR),Y
016F F0 E9 314 beq EPRETURN
0171      315 ;
0171      316 .if HWCARD
0171 A2 80 317 ldx #EPOFFVAL
0173      318 .el
0173      319 ldx #QLOFFVAL
0173      320 .fi
0173      321 ;
0173 8E 80 C0 322 EXECMOD2 stx EPSELC
0176      323 ;
0176 20 ED FD 324 jsr COUT
0179      325 ;
0179 E6 FA 326 inc EXECPTR
017B D0 E8 327 bne EPEXEC
017D      328 ;
017D E6 FB 329 inc EXECPTR+1
017F D0 E4 330 bne EPEXEC
0181      331 ;
0181 A9 E0 332 lda /PAGEE0
0183 85 FB 333 sta EXECPTR+1
0185      334 ;
0185 EE 95 02 335 inc EPBANK
0188 D0 DB 336 bne EPEXEC ; always taken
018A      337 ;
018A      338 ;
007A      339 EPBMLEN equ *-EPBMPAGE
018A      340 ;
018A      341 ;
018A      342 phs EPBMCODE+EPBMLEN
23FF      343 ;
23FF      344 ;
23FF      345 ; Internal DCB structures.
23FF      346 ;
23FF      347 ;
23FF      348 ; Load Lisa80 DCB
23FF      349 ;
23FF      350 LLDCB:
23FF      351 ;
23FF 02 352 LLCMD byt RUNCMD ; run
2400 F0 353 LLEPROM byt SRCHALL ; search all EPROMs
2401 00 00 354 LLALTADR adr *-* ; no alternate address
2403 FF 355 LLSTAT byt NEGONE ; return status
2404 0A 356 LLNAMLEN byt LLNLEN ; length of filename
2405 D8 02 357 LLNAMADR adr LLNADR ; address of filename
2407 CC CF C1 358 LLNAME asc "LOADLISA80"
240A C4 CC C9
240D D3 C1 B8
2410 B0
2411      359 ;
000A      360 LLNLEN equ *-LLNAME
0012      361 LLDLEN equ *-LLDCB
02D8      362 LLNADR equ DCBBUFR+LLNAME-LLDCB

```

```

2411          363 ;
2411          364 ;
2411          365 ; RamDisk Config DCB
2411          366 ;
2411          367 RDDCB:
2411          368 ;
2411 02        369 RDCMD      byt  RUNCMD          ; run
2412 F0        370 RDEPROM    byt  SRCHALL          ; search all EPROMs
2413 00 00     371 RDALTADR   adr  *-*              ; no alternate address
2415 FF        372 RDSTAT     byt  NEGONE             ; return status
2416 0E        373 RDNAMLEN   byt  RDNLEN           ; length of filename
2417 D8 02     374 RDNAMADR   adr  RDNADR          ; address of filename
2419 D2 E1 ED  375 RDNNAME    asc  "RamDisk Config"
241C C4 E9 F3
241F EB A0 C3
2422 EF EE E6
2425 E9 E7
2427          376 ;
000E          377 RDNLEN      equ  *-RDNNAME
0016          378 RDDLEN      equ  *-RDDCB
02D8          379 RDNADR      equ  DCBBUFR+RDNNAME-RDDCB
2427          380 ;
2427          381 ;
2427          382 ; FID DCB
2427          383 ;
2427          384 FDDCB:
2427          385 ;
2427 02        386 FDCMD      byt  RUNCMD          ; run
2428 F0        387 FDEPROM    byt  SRCHALL          ; search all EPROMs
2429 00 00     388 FDALTADR   adr  *-*              ; no alternate address
242B FF        389 FDSTAT     byt  NEGONE             ; return status
242C 03        390 FDNAMLEN   byt  FDNLEN           ; length of filename
242D D8 02     391 FDNAMADR   adr  FDNADR          ; address of filename
242F C6 C9 C4  392 FDNAME     asc  "FID"
2432          393 ;
0003          394 FDNLEN      equ  *-FDNAME
000B          395 FDDLLEN     equ  *-FDDCB
02D8          396 FDNADR      equ  DCBBUFR+FDNAME-FDDCB
2432          397 ;
2432          398 ;
2432          399 ; ADT2 DCB
2432          400 ;
2432          401 ADDCB:
2432          402 ;
2432 02        403 ADCMD      byt  RUNCMD          ; run
2433 F0        404 ADEPROM    byt  SRCHALL          ; search all EPROMs
2434 00 00     405 ADALTADR   adr  *-*              ; no alternate address
2436 FF        406 ADSTAT     byt  NEGONE             ; return status
2437 04        407 ADNAMLEN   byt  ADNLEN           ; length of filename
2438 D8 02     408 ADNAMADR   adr  ADNADR          ; address of filename
243A C1 C4 D4  409 ADNAME     asc  "ADT2"
243D B2
243E          410 ;
0004          411 ADNLEN      equ  *-ADNAME
000C          412 ADDLEN      equ  *-ADDCB
02D8          413 ADNADR      equ  DCBBUFR+ADNAME-ADDCB
243E          414 ;
243E          415 ;
243E          416 ; Set Clock DCB
243E          417 ;
243E          418 SCDCB:

```

```

243E          419 ;
243E 02        420 SCCMD      byt  RUNCMD          ; run
243F F0        421 SCEPROM   byt  SRCHALL         ; search all EPROMs
2440 00 00     422 SCALTADR  adr  *-*           ; no alternate address
2442 FF        423 SCSTAT    byt  NEGONE          ; return status
2443 09        424 SCNAMLEN  byt  SCNLEN          ; length of filename
2444 D8 02     425 SCNAMADR  adr  SCNADR          ; address of filename
2446 D3 E5 F4  426 SCNAME    asc  "Set Clock"
2449 A0 C3 EC
244C EF E3 EB
244F          427 ;
0009          428 SCNLEN     equ  *-SCNAME
0011          429 SCDLEN     equ  *-SCDCB
02D8          430 SCNADR     equ  DCBBUFR+SCNAME-SCDCB
244F          431 ;
244F          432 ;
244F          433 ; Applesoft Formatter DCB
244F          434 ;
244F          435 AFDCB:
244F          436 ;
244F 02        437 AFCMD      byt  RUNCMD          ; run
2450 F0        438 AFEPROM   byt  SRCHALL         ; search all EPROMs
2451 00 00     439 AFALTADR  adr  *-*           ; no alternate address
2453 FF        440 AFSTAT    byt  NEGONE          ; return status
2454 13        441 AFNAMLEN  byt  AFNLEN          ; length of filename
2455 D8 02     442 AFNAMADR  adr  AFNADR          ; address of filename
2457 C1 F0 F0  443 AFNAME    asc  "Applesoft Formatter"
245A EC E5 F3
245D EF E6 F4
2460 A0 C6 EF
2463 F2 ED E1
2466 F4 F4 E5
2469 F2
246A          444 ;
0013          445 AFNLEN     equ  *-AFNAME
001B          446 AFDLEN     equ  *-AFDCB
02D8          447 AFNADR     equ  DCBBUFR+AFNAME-AFDCB
246A          448 ;
246A          449 ;
246A          450 ; Load SOURCEROR DCB
246A          451 ;
246A          452 LSDCB:
246A          453 ;
246A 02        454 LSCMD      byt  RUNCMD          ; run
246B F0        455 LSEPROM   byt  SRCHALL         ; search all EPROMs
246C 00 00     456 LSALTADR  adr  *-*           ; no alternate address
246E FF        457 LSSTAT    byt  NEGONE          ; return status
246F 0E        458 LSNAMLEN  byt  LSNLEN          ; length of filename
2470 D8 02     459 LSNAMADR  adr  LSNADR          ; address of filename
2472 CC EF E1  460 LSNAME    asc  "Load Sourceror"
2475 E4 A0 D3
2478 EF F5 F2
247B E3 E5 F2
247E EF F2
2480          461 ;
000E          462 LSNLEN     equ  *-LSNAME
0016          463 LSDLEN     equ  *-LSDCB
02D8          464 LSNADR     equ  DCBBUFR+LSNAME-LSDCB
2480          465 ;
2480          466 ;
2480          467 ; EPROM Burner DCB

```

```
2480          468 ;
2480          469 EBD CB:
2480          470 ;
2480 02        471 EBCMD      byt  RUNCMD          ; run
2481 F0        472 EBEPROM   byt  SRCHALL          ; search all EPROMs
2482 00 00     473 EBALTADR  adr  *-*              ; no alternate address
2484 FF        474 EBSTAT    byt  NEGONE           ; return status
2485 0C        475 EBNAMLEN  byt  EBNLEN           ; length of filename
2486 D8 02     476 EBNAMADR  adr  EBNADR          ; address of filename
2488 C5 D0 D2  477 EBNAME    asc  "EPROM Burner"
248B CF CD A0
248E C2 F5 F2
2491 EE E5 F2
2494          478 ;
000C          479 EBNLEN     equ  *-EBNAME
0014          480 EBDLEN     equ  *-EBDCB
02D8          481 EBNADR     equ  DCBBUFR+EBNAME-EBDCB
2494          482 ;
2494          483 ;
2494          484 ; VTOC Manager DCB
2494          485 ;
2494          486 VTDCB:
2494          487 ;
2494 02        488 VTCMD      byt  RUNCMD          ; run
2495 F0        489 VTEPROM   byt  SRCHALL          ; search all EPROMs
2496 00 00     490 VTALTADR  adr  *-*              ; no alternate address
2498 FF        491 VTSTAT    byt  NEGONE           ; return status
2499 0C        492 VTNAMLEN  byt  VTNLEN           ; length of filename
249A D8 02     493 VTNAMADR  adr  VTNADR          ; address of filename
249C D6 D4 CF  494 VTNAME    asc  "VTOC Manager"
249F C3 A0 CD
24A2 E1 EE E1
24A5 E7 E5 F2
24A8          495 ;
000C          496 VTNLEN     equ  *-VTNAME
0014          497 VTDLEN     equ  *-VTDCB
02D8          498 VTNADR     equ  DCBBUFR+VTNAME-VTDCB
24A8          499 ;
24A8          500 ;
24A8          501 ; Volume Manager DCB
24A8          502 ;
24A8          503 VODCB:
24A8          504 ;
24A8 02        505 VOCMD      byt  RUNCMD          ; run
24A9 F0        506 VOEPROM   byt  SRCHALL          ; search all EPROMs
24AA 00 00     507 VOALTADR  adr  *-*              ; no alternate address
24AC FF        508 VOSTAT    byt  NEGONE           ; return status
24AD 0E        509 VONAMLEN  byt  VONLEN           ; length of filename
24AE D8 02     510 VONAMADR  adr  VONADR          ; address of filename
24B0 D6 EF EC  511 VONAME    asc  "Volume Manager"
24B3 F5 ED E5
24B6 A0 CD E1
24B9 EE E1 E7
24BC E5 F2
24BE          512 ;
000E          513 VONLEN     equ  *-VONAME
0016          514 VODLEN     equ  *-VODCB
02D8          515 VONADR     equ  DCBBUFR+VONAME-VODCB
24BE          516 ;
24BE          517 ;
24BE          518 ; Volume Duplicate DCB
```



```

24BE          519 ;
24BE          520 VDDCB:
24BE          521 ;
24BE 02        522 VDCMD      byt  RUNCMD          ; run
24BF F0        523 VDEPROM    byt  SRCHALL          ; search all EPROMs
24C0 00 00     524 VDALTADR   adr  *-*          ; no alternate address
24C2 FF        525 VDSTAT     byt  NEGONE          ; return status
24C3 10        526 VDNAMLEN   byt  VDNLEN          ; length of filename
24C4 D8 02     527 VDNAMADR   adr  VDNADR          ; address of filename
24C6 D6 EF EC  528 VDNAME     asc  "Volume Duplicate"
24C9 F5 ED E5
24CC A0 C4 F5
24CF F0 EC E9
24D2 E3 E1 F4
24D5 E5
24D6          529 ;
0010          530 VDNLEN     equ  *-VDNAME
0018          531 VDDLLEN    equ  *-VDDCB
02D8          532 VDNADR     equ  DCBBUFR+VDNAME-VDDCB
24D6          533 ;
24D6          534 ;
24D6          535 ; Disk Window DCB
24D6          536 ;
24D6          537 DWDCB:
24D6          538 ;
24D6 02        539 DWCMD      byt  RUNCMD          ; run
24D7 F0        540 DWEPROM    byt  SRCHALL          ; search all EPROMs
24D8 00 00     541 DWALTADR   adr  *-*          ; no alternate address
24DA FF        542 DWSTAT     byt  NEGONE          ; return status
24DB 0B        543 DWNAMLEN   byt  DWNLEN          ; length of filename
24DC D8 02     544 DWNAMADR   adr  DWNADR          ; address of filename
24DE C4 E9 F3  545 DWNAME     asc  "Disk Window"
24E1 EB A0 D7
24E4 E9 EE E4
24E7 EF F7
24E9          546 ;
000B          547 DWNLEN     equ  *-DWNAME
0013          548 DWDLEN     equ  *-DWDCB
02D8          549 DWNADR     equ  DCBBUFR+DWNAME-DWDCB
24E9          550 ;
24E9          551 ;
24E9          552 ; Binary File Install DCB
24E9          553 ;
24E9          554 BFDCB:
24E9          555 ;
24E9 02        556 BFCMD      byt  RUNCMD          ; run
24EA F0        557 BFEPROM    byt  SRCHALL          ; search all EPROMs
24EB 00 00     558 BFALTADR   adr  *-*          ; no alternate address
24ED FF        559 BFSTAT     byt  NEGONE          ; return status
24EE 13        560 BFNAMLEN   byt  BFNLEN          ; length of filename
24EF D8 02     561 BFNAMADR   adr  BFNADR          ; address of filename
24F1 C2 E9 EE  562 BFNAME     asc  "Binary File Install"
24F4 E1 F2 F9
24F7 A0 C6 E9
24FA EC E5 A0
24FD C9 EE F3
2500 F4 E1 EC
2503 EC
2504          563 ;
0013          564 BFNLEN     equ  *-BFNAME
001B          565 BFDLEN     equ  *-BFDCB

```

```
02D8          566 BFNADR    equ DCBBUFR+BFNAME-BFDCB
2504          567 ;
2504          568 ;
2504          569 ; BigMac DCB
2504          570 ;
2504          571 BMDCB:
2504          572 ;
2504 02         573 BMCMD     byt  RUNCMD          ; run
2505 F0        574 BMEPROM   byt  SRCHALL         ; search all EPROMs
2506 00 00     575 BMALTADR  adr  *-*             ; no alternate address
2508 FF        576 BMSTAT    byt  NEGONE          ; return status
2509 0B        577 BMNAMLEN  byt  BMNLEN          ; length of filename
250A D8 02    578 BMNAMADR  adr  BMNADR          ; address of filename
250C CC EF E1 579 BMNAME    asc  "Load BigMac"
250F E4 A0 C2
2512 E9 E7 CD
2515 E1 E3
2517          580 ;
000B          581 BMNLEN     equ  *-BMNAME
0013          582 BMDLEN     equ  *-BMDCB
02D8          583 BMNADR     equ  DCBBUFR+BMNAME-BMDCB
2517          584 ;
2517          585 ;
2517          586 ; Scan Disk DCB
2517          587 ;
2517          588 SDDCB:
2517          589 ;
2517 02        590 SDCMD      byt  RUNCMD          ; run
2518 F0        591 SDEPROM   byt  SRCHALL         ; search all EPROMs
2519 00 00     592 SDALTADR  adr  *-*             ; no alternate address
251B FF        593 SDSTAT    byt  NEGONE          ; return status
251C 04        594 SDNAMLEN  byt  SDNLEN          ; length of filename
251D D8 02    595 SDNAMADR  adr  SDNADR          ; address of filename
251F D3 E3 E1 596 SDNAME     asc  "Scan"
2522 EE
2523          597 ;
0004          598 SDNLEN     equ  *-SDNAME
000C          599 SDDLLEN    equ  *-SDDCB
02D8          600 SDNADR     equ  DCBBUFR+SDNAME-SDDCB
2523          601 ;
2523          602 ;
2523          603          icl  "TABLES.L"
```

LLOAD TABLES.L,A\$4000

```
2523          1          ttl "EOS+ Source Code, TABLES.L"
2523          2          ;
2523          3          ;
2523          4          ; TABLES.L
2523          5          ;
2523          6          ;
2523          7          CMDTBL:
2523 F1 13          8          adr AHNDLR-1
2525 F8 13          9          adr BHNDLR-1
2527 0E 14         10         adr CHNDLR-1
2529 18 14         11         adr DHNDLR-1
252B 22 14         12         adr EHNDLR-1
252D 2D 14         13         adr FHNDLR-1
252F 30 14         14         adr GHNDLR-1
2531 3F 14         15         adr HHNDLR-1
2533 48 14         16         adr IHNDLR-1
2535 51 14         17         adr JHNDLR-1
2537 5A 14         18         adr KHNDLR-1
2539 63 14         19         adr LHNDLR-1
253B 6C 14         20         adr MHNDLR-1
253D 75 14         21         adr NHNDLR-1
253F 7B 14         22         adr OHNDLR-1
2541 7E 14         23         adr PHNDLR-1
2543 AA 14         24         adr QHNDLR-1
2545 B3 14         25         adr RHNDLR-1
2547 BC 14         26         adr SHNDLR-1
2549 C5 14         27         adr THNDLR-1
254B CE 14         28         adr UHNDLR-1
254D D7 14         29         adr VHNDLR-1
254F E0 14         30         adr WHNDLR-1
2551 E9 14         31         adr XHNDLR-1
2553 F2 14         32         adr YHNDLR-1
2555 FB 14         33         adr ZHNDLR-1
2557          34          ;
2557          35          ;
2557          36          ADDRTBLL:
2557 ED          37          byt COUT          ; X = 0
2558 84          38          byt SETNORM        ; X = 1
2559 2F          39          byt INIT          ; X = 2
255A 93          40          byt SETVID        ; X = 3
255B 89          41          byt SETKBD        ; X = 4
255C 58          42          byt HOME          ; X = 5
255D 22          43          byt VTAB          ; X = 6
255E 9C          44          byt CLREOL        ; X = 7
255F 42          45          byt CLREOP        ; X = 8
2560 EA          46          byt HOOKDOS       ; X = 9
2561 41          47          byt PRNTAX        ; X = 10
2562 DA          48          byt PRBYTE        ; X = 11
2563 E3          49          byt PRHEX         ; X = 12
2564          50          ;
2564          51          ADDRTBLH:
2564 FD          52          hby COUT
2565 FE          53          hby SETNORM
2566 FB          54          hby INIT
2567 FE          55          hby SETVID
2568 FE          56          hby SETKBD
2569 FC          57          hby HOME
256A FC          58          hby VTAB
256B FC          59          hby CLREOL
256C FC          60          hby CLREOP
```

```

256D 03          61          hby HOOKDOS
256E F9          62          hby PRNTAX
256F FD          63          hby PRBYTE
2570 FD          64          hby PRHEX
2571             65          ;
2571             66          ;
2571             67          LINETBL:
2571 03 05 07     68          hex 030507090A0C
2574 09 0A 0C
2577 0D 0E 0F     69          hex 0D0E0F101112
257A 10 11 12
257D             70          ;
257D             71          ;
257D 01 0A 64     72          DECTBLL byt 1,10,100
2580             73          ;
2580             74          ;
2580             75          MAPMASKS:
2580 01 02 04     76          hex 01020408
2583 08
2584 10 20 40     77          hex 10204080
2587 80
2588             78          ;
2588             79          ;
2588             80          ZCDEFLT:
2588             81          ;
2588 E6           82          byt %11100110          ; slots
2589             83          ;
2589 00           84          hex 00          ; speed (from ZCSPDTBL)
258A             85          ;
258A             86          ;
258A 00 08 40     87          ZCSPDTBL hex 000840C8
258D C8
258E             88          ;
258E             89          ;
258E D4 C1 C2     90          PARMTYPE asc "TABBBRSP"
2591 C2 C2 D2
2594 D3 D0
2596             91          ;
2596             92          ;
2596             93          LOADTBL:
2596 20 19         94          adr LOADTEXT-1
2598 C4 19         95          adr EXECAS-1
259A 03 1A         96          adr EXECBIN0-1
259C F4 19         97          adr EXECBIN1-1
259E FD 19         98          adr EXECBIN2-1
25A0             99          ;
25A0             100         ;
25A0             101         RUNTBL:
25A0 3F 15         102         adr EXECTEXT-1
25A2 C4 19         103         adr EXECAS-1
25A4 03 1A         104         adr EXECBIN0-1
25A6 F4 19         105         adr EXECBIN1-1
25A8 FD 19         106         adr EXECBIN2-1
25AA             107         ;
25AA             108         ;
25AA             109         TYPETBL:
25AA 00           110         byt TYPETEXT-TYPTTEXTS
25AB 05           111         byt TYPEAS-TYPTTEXTS
25AC 0F           112         byt TYPEBIN0-TYPTTEXTS
25AD 23           113         byt TYPEBIN1-TYPTTEXTS
25AE 35           114         byt TYPEBIN2-TYPTTEXTS

```

```

25AF 47          115          byt TYPERESD-TYPTEXTS
25B0 50          116          byt TYPESYS-TYPTEXTS
25B1 57          117          byt TYPEPRI-TYPTEXTS
25B2           118          ;
25B2           119          ;
25B2           120          TYPTEXTS:
25B2           121          ;
25B2 D4 E5 F8    122          TYPETEXT asc "Text"
25B5 F4
25B6 00          123          hex 00
25B7           124          ;
25B7 C1 F0 F0    125          TYPEAS  asc "Applesoft"
25BA EC E5 F3
25BD EF E6 F4
25C0 00          126          hex 00
25C1           127          ;
25C1 C2 E9 EE    128          TYPEBIN0 asc "Binary, main memory"
25C4 E1 F2 F9
25C7 AC A0 ED
25CA E1 E9 EE
25CD A0 ED E5
25D0 ED EF F2
25D3 F9
25D4 00          129          hex 00
25D5           130          ;
25D5 C2 E9 EE    131          TYPEBIN1 asc "Binary, LC Bank 1"
25D8 E1 F2 F9
25DB AC A0 CC
25DE C3 A0 C2
25E1 E1 EE EB
25E4 A0 B1
25E6 00          132          hex 00
25E7           133          ;
25E7 C2 E9 EE    134          TYPEBIN2 asc "Binary, LC Bank 2"
25EA E1 F2 F9
25ED AC A0 CC
25F0 C3 A0 C2
25F3 E1 EE EB
25F6 A0 B2
25F8 00          135          hex 00
25F9           136          ;
25F9 D2 E5 F3    137          TYPERESD asc "Reserved"
25FC E5 F2 F6
25FF E5 E4
2601 00          138          hex 00
2602           139          ;
2602 D3 F9 F3    140          TYPESYS  asc "System"
2605 F4 E5 ED
2608 00          141          hex 00
2609           142          ;
2609 D0 F2 E9    143          TYPEPRI  asc "Primary"
260C ED E1 F2
260F F9
2610 00          144          hex 00
2611           145          ;
2611           146          ;
2611           147          ; Addresses of text strings that are issued from the EOS
2611           148          ; Main Selection Menu in order to support the DOS CATALOG
2611           149          ; and the RUN HELLO commands.
2611           150          ;
2611           151          TEXTTBLL:

```

```

2611 1B          152          byt CATTEXT          ; X = 0
2612 25          153          byt HELOTEXT         ; X = 1
2613 31          154          byt SLTTEXT          ; X = 2
2614 34          155          byt DRVTEXT          ; X = 3
2615 38          156          byt VOLTEXT          ; X = 4
2616            157          ;
2616            158          TEXTTBLH:
2616 26          159          hby CATTEXT
2617 26          160          hby HELOTEXT
2618 26          161          hby SLTTEXT
2619 26          162          hby DRVTEXT
261A 26          163          hby VOLTEXT
261B            164          ;
261B            165          ;
261B 8D 84       166          CATTEXT  byt RETURN,CTRLD
261D C3 C1 D4    167          asc "CATALOG"
2620 C1 CC CF
2623 C7
2624 00          168          byt ZERO
2625            169          ;
2625 8D 84       170          HELOTEXT byt RETURN,CTRLD
2627 D2 D5 CE    171          asc "RUN HELLO"
262A A0 C8 C5
262D CC CC CF
2630 00          172          byt ZERO
2631            173          ;
2631 AC D3       174          SLTTEXT  asc ",S"
2633 00          175          byt ZERO
2634            176          ;
2634 AC C4 A4    177          DRVTEXT  asc ",D$"
2637 00          178          byt ZERO
2638            179          ;
2638 AC D6 A4    180          VOLTEXT  asc ",V$"
263B 00          181          byt ZERO
263C            182          ;
263C            183          ;
263C            184          ; Text used for discovering EPROM cards in slots.
263C            185          ;
263C C5 D0 C2    186          EPTEXT   asc "EPBINEOS"          ; EPBIN text
263F C9 CE C5
2642 CF D3
2644            187          ;
2644            188          ;
2644            189          ; Text strings used in the ZipChip display routine.
2644            190          ;
2644 CF EE 20     191          ZCTEXT   dci "On "
2647 CF E6 66     192          ZCTEXT1  dci "Off"
264A B8 AE B0     193          ZCTEXT2  dci "8.00"
264D 30
264E B6 AE B0     194          dci "6.00"
2651 30
2652 B4 AE B0     195          dci "4.00"
2655 30
2656 B2 AE B0     196          dci "2.00"
2659 30
265A C5 EE E1     197          ZCTEXT3  dci "Enabled "
265D E2 EC E5
2660 E4 20
2662 C4 E9 F3     198          ZCTEXT4  dci "Disabled"
2665 E1 E2 EC
2668 E5 64

```

```
266A C6 E1 F3    199  ZCTEXT5  dci "Fast  "
266D F4 A0 20
2670 CE EF F2    200  ZCTEXT6  dci "Normal"
2673 ED E1 6C
2676                201  ;
2676                202  ;
2676 C4 B8 90    203  SYNCBYTS byt SYNCBYT0,SYNCBYT1,SYNCBYT2,SYNCBYT3
2679 ED
267A                204  ;
267A                205  ;
267A                206  ; EOS code end.
267A                207  ;
267A                208  ;
00FA                209  EOSEND    equ EOSVCTRS&NEGONE
267A                210  ;
267A                211  ;
267A                212          .if DEBUG
267A                213          dfs EOSEND-*&NEGONE
26FA                214          .el
26FA                215          dfs EOSVCTRS-*,NEGONE
26FA                216          .fi
26FA                217  ;
26FA                218  ;
26FA 00 10        219  NMIRTN    adr EOS
26FC 00 10        220  RESET     adr EOS
26FE 00 10        221  IRQRTN    adr EOS
2700                222  ;
2700                223  ;
```

BSAVE SEG01,D1,A\$0800,B,L\$1F00

```
2700                224          usr SEG01,D1
2700                225  ;
2700                226  ;
2700                227          icl "BANKS.L,D2"
```

LLOAD BANKS.L,D2,A\$4000

```
2700          1          ttl "EOS+ Source Code, BANKS.L"
2700          2          ;
2700          3          ;
2700          4          ; BANKS.L
2700          5          ;
2700          6          ;
CD D1
```

```
2700          7          dcm "CD D1"
2700          8          ;
2700          9          ;
2700         10          ; First half of EPROM.
2700         11          ;
2700         12          ;
2000         13          org PAGE20
2000         14          obj PAGE08
2000         15          usr
2000         16          ;
2000         17          ;
2000         18          DOSL.O:
2000         19          ;
```

BLOAD DOS4.5L,A\$0800

```
2000         20          usr .DOS4.5L
2000         21          ;
2000         22          dfs DOSL.L
4100         23          ;
4100         24          ;
```

BSAVE SEG02,A\$0800,B,L\$2100

```
4100         25          usr SEG02
4100         26          ;
4100         27          ;
4100         28          obj PAGE08
4100         29          usr
4100         30          ;
4100         31          ;
4100         32          DOSH.O:
4100         33          ;
```

BLOAD DOS4.5H,A\$0800

```
4100         34          usr .DOS4.5H
4100         35          ;
4100         36          dfs DOSH.L
6B00         37          ;
6B00         38          ;
```

BSAVE SEG03,A\$0800,B,L\$2A00

```
6B00         39          usr SEG03
6B00         40          ;
6B00         41          ;
6B00         42          obj PAGE08
6B00         43          usr
6B00         44          ;
6B00         45          ;
```



```
6B00          46  LISA81.O:
6B00          47  ;
```

```
BLOAD LISA80.1,A$0800
```

```
6B00          48          usr .LISA80.1
6B00          49  ;
6B00          50          dfs PAGE80-*
8000          51  ;
8000          52  ;
```

```
BSAVE SEG04,A$0800,B,L$1500
```

```
8000          53          usr SEG04
8000          54  ;
8000          55  ;
1300          56  LEFTOVER equ LISA81.O+LISA81.L-PAGE80
8000          57  ;
8000          58          usr
8000          59  ;
8000          60  ;
8000          61          dfs LEFTOVER
9300          62  ;
9300          63  ;
```

```
BSAVE SEG05,A$1D00,B,L$1300
```

```
9300          64          usr SEG05
9300          65  ;
9300          66  ;
9300          67  ; Second half of EPROM.
9300          68  ;
9300          69  ;
8000          70          org PAGE80
8000          71          obj PAGE08
8000          72          usr
8000          73  ;
8000          74  ;
```

```
BLOAD SEG05,A$0800
```

```
8000          75          usr .SEG05
8000          76  ;
8000          77          dfs LEFTOVER
9300          78  ;
9300          79  ;
9300          80  LISA82.O:
9300          81  ;
```

```
BLOAD LISA80.2,A$1B00
```

```
9300          82          usr .LISA80.2
9300          83  ;
9300          84  ;
9300          85          dfs LISA82.L
A300          86  ;
A300          87  ;
```

```
BSAVE SEG05,A$0800,B,L$2300
```

```
A300          88          usr SEG05
```

```
A300      89  ;
A300      90  ;
A300      91      obj PAGE08
A300      92      usr
A300      93  ;
A300      94  ;
A300      95  LISA83.O:
A300      96  ;
```

BLOAD LISA80.3,A\$0800

```
A300      97      usr .LISA80.3
A300      98  ;
A300      99      dfs LISA83.L
A940     100  ;
A940     101  ;
A940     102  SETUP8.O:
A940     103  ;
```

BLOAD SETUP80,A\$0E40

```
A940     104      usr .SETUP80
A940     105  ;
A940     106      dfs SETUP8.L
C254     107  ;
C254     108  ;
```

BSAVE SEG06,A\$0800,B,L\$1F54

```
C254     109      usr SEG06
C254     110  ;
C254     111  ;
C254     112      obj PAGE08
C254     113      usr
C254     114  ;
C254     115  ;
C254     116  LLISA8.O:
C254     117  ;
```

BLOAD LOADLISA80,A\$0800

```
C254     118      usr .LOADLISA80
C254     119  ;
C254     120      dfs LLISA8.L
C414     121  ;
C414     122  ;
C414     123  RAMDSK.O:
C414     124  ;
```

BLOAD RAMDISK,A\$09C0

```
C414     125      usr .RAMDISK
C414     126  ;
C414     127      dfs RAMDSK.L
DF18     128  ;
DF18     129  ;
```

BSAVE SEG07,A\$0800,B,L\$1CC4

```
DF18     130      usr SEG07
DF18     131  ;
```

```
DF18      132  ;
DF18      133      obj PAGE08
DF18      134      usr
DF18      135  ;
DF18      136  ;
DF18      137  FID.O:
DF18      138  ;
```

BLOAD FID,A\$0800

```
DF18      139      usr .FID
DF18      140  ;
DF18      141      dfs FID.L
F22E      142  ;
F22E      143  ;
F22E      144  CLK.O:
F22E      145  ;
```

BLOAD SETCLOCK,A\$1B16

```
F22E      146      usr .SETCLOCK
F22E      147  ;
F22E      148      dfs CLK.L
F878      149  ;
F878      150  ;
F878      151  LIST.O:
F878      152  ;
```

BLOAD ASLIST,A\$2160

```
F878      153      usr .ASLIST
F878      154  ;
F878      155      dfs LIST.L
FF51      156  ;
FF51      157  ;
00AF      158  UNUSED equ ZERO-*
FF51      159  ;
FF51      160      dfs UNUSED,NEGONE
0000      161  ;
0000      162  ;
```

BSAVE SEG08,A\$0800,B,L\$20E8

```
0000      163      usr SEG08
0000      164  ;
0000      165  ;
CD D2
```

```
0000      166      dcm "CD D2"
0000      167  ;
0000      168  ;
0000      169      stt "EOS Symbol Table"
0000      170  ;
0000      171  ;
0000      172      end 111
```

*** End of Assembly

Symbol List starts at 0x7800, ends at 0x94E8, used 0x1CE8, remaining 0x1EF8

Symbols unsorted:

XREG	0006	YREG	0007	AREG	0008	XSAV	0016	YSAV	0017
ASAV	0018	WNDLFT	0020	WNDWDTH	0021	WNDTOP	0022	WNCBTM	0023
CH	0024	CV	0025	SRCPTR	002A	LENPTR	002C	DSTPTR	002E
PROMPT	0033	CSWL	0036	KSWL	0038	ASPGMST	0067	ASVARS	0069
CURLIN	0075	DSCTMP	009D	ASPEND	00AF	CHRGOT	00B7	CHRADR	00B8
GENPTR	00CE	ASONERR	00D8	MSLOT	00EB	DRIVE	00EC	VOLUME	00ED
CMDPTR	00EE	EXECPTR	00FA	PRNTPTR	00FC	FINDERR	080C	FINDEP	0900
BINJMP	0959	BINADR	095C	EOSLDCB	095E	BINCMD	095E	BINEPN	095F
BINFALT	0960	BINSTAT	0962	BINFLEN	0963	BINFADR	0964	FILNAM	0966
FILEND	0974	QLCARD	0000	EPCARD	0001	ZERO	0000	NEGONE	00FF
QLMASK	0007	EPMASK	000F	PRNTMASK	000F	VALUMASK	000F	MENUMASK	001F
CVMASK	001F	BANKMASK	001F	MSBCLR	007F	MSBSET	0080	CTRLC	0083
CTRLD	0084	LARROW	0088	DARROW	008A	UARROW	008B	RETURN	008D
CTRLS	0093	RARROW	0095	ESCAPE	009B	SPACE	00A0	DOLLAR	00A4
COMMA	00AC	LWRMASK	00DF	LWRCASE	00E0	SLOT3	0003	INDENT	0005
PARMSIZE	0006	DCBSIZE	0008	NAMESIZE	0018	ENTRYLEN	0020	ASPNUM4	0004
ASPNUM5	0005	ASPNUM6	0006	MAXASNUM	000C	INTERNAL	0001	EXTERNAL	0002
LOADCMD	0001	RUNCMD	0002	CATCMD	0003	QLSRCH	0070	EPSRCH	00F0
ERR00	0000	ERR01	0001	ERR02	0002	ERR03	0003	ERR04	0004
ERR05	0005	TESTCNT	0020	MAXCH	0050	MINCV	0060	RTNCMD	0050
NORMCMD	0051	INITCMD	0052	VIDCMD	0053	KBDCMD	0054	HEMCMC	0055
TABVCMD	0056	EOLCMD	0057	EOPCMD	0058	CNTRCMD	0059	EPONVAL	0000
EPUSR	0008	QLOFFVAL	0010	EPOFFVAL	0080	ZCONVAL	0000	ZCOPTNS	000C
ZCOFFVAL	0010	ZCSTAT	0010	ZCNSPEED	0004	ZCUNLOCK	005A	ZCLOCK	00A5
ENDCAT	0000	TEXTFILE	0001	APLSOFT	0002	BINARY0	0004	BINARY1	0008
BINARY2	0010	RESERVED	0020	SYSTEM	0040	PRIMARY	0080	SYNCHYT0	00C4
SYNCHYT1	00B8	SYNCHYT2	0090	SYNCHYT3	00ED	RUNMODE	00FF	STACK	0100
PAGESIZE	0100	STKCODE	0110	INPUT	0200	PRISLOT	0290	EPSLOT	0291
SLOTMAP	0292	APPLTYPE	0293	EPNMBR	0294	EPBANK	0295	RTNTYPE	0296
TEMPVAL	0297	EPSTRT	0298	EPEND	0299	ZSTATUS	029A	ZCACHE	029B
NUMIN	029C	NUMSEL	029D	FLENGTH	029E	RUNFLAG	029F	ASPRNUM	02A0
ASSTATUS	02A1	EPSEARCH	02A2	FILELEN	02A3	SLOT16	02A4	SLOT	02A5
MEMJMP	02A6	SLOTJMP	02A8	ADDRBUFR	02AA	SYNCHUFR	02AC	FILENTRY	02B0
FILEPNUM	02B0	FILETYPE	02B1	SRCVAL	02B2	LENVAL	02B4	DSTVAL	02B6
FILENAME	02B8	ZCSETBL	02D0	NUMSCRN	02D0	FIRSTIME	02D1	FILECNT	02D2
NUMNTRY	02D3	LSTOPNTY	02D4	NTRYSTRT	02D5	NTRYEND	02D6	FILTYPE	02D7
INDEX	02D8	ASPRADRS	02D0	ASPCMD	02D0	ASPSTAT	02D2	ASPSRCH	02D4
ASPFILE	02D6	ASPNUM	02D6	ASPADR	02D8	ASPFILES	02D8	ASPPARMS	02DA
DCBBUFR	02D0	DCBCMD	02D0	DCBEPN	02D1	DCBFALT	02D2	DCBSTAT	02D4
DCBFLEN	02D5	DCBFADR	02D6	DOSWARM	03D0	DOSCOLD	03D3	HOOKDOS	03EA
XMODE	04FB	PWRUP0	0478	PWRUP1	0578	PWRUP2	0678	PWRUP3	0778
LINE00	0400	LINE01	0480	LINE02	0500	LINE03	0580	LINE04	0600
LINE05	0680	LINE06	0700	LINE07	0780	LINE08	0428	LINE09	04A8
LINE10	0528	LINE11	05A8	LINE12	0628	LINE13	06A8	LINE14	0728
LINE15	07A8	LINE16	0450	LINE17	04D0	LINE18	0550	LINE19	05D0
LINE20	0650	LINE21	06D0	LINE22	0750	LINE23	07D0	STARTAS	0801
BANKSIZE	2000	MNGUSER	BFF6	INITDOS	BFF8	PAGE08	0800	PAGE09	0900
PAGE20	2000	PAGE60	6000	PAGE80	8000	PAGE9F	9F00	PAGEBE	BE00
PAGEC0	C000	PAGEC1	C100	PAGEC7	C700	PAGED0	D000	PAGEDE	DE00
PAGEE0	E000	PAGEE1	E100	PAGEE7	E700	PAGEE8	E800	PAGEEA	EA00
KEY	C000	STR80OFF	C000	RAMRDOFF	C002	RAMWROFF	C004	CXROMOFF	C006
CXROMON	C007	AUXZPOFF	C008	C3ROMOFF	C00A	C3ROMON	C00B	VID80OFF	C00C
ALTCHOFF	C00E	RDCXROM	C015	CLRKEY	C010	SPKR	C030	HOOKSLT	C010
UHOOKSLT	C018	TEXTON	C051	PAGE1ON	C054	HIRESOFF	C056	ANN1OFF	C058
ANN2OFF	C05A	ANN3ON	C05D	ANN4ON	C05F	ZCCTRL	C05A	ZCSTATS	C05B
ZCSLOTS	C05C	ZCSPEED	C05D	ZCDELAY	C05E	ZCCACHE	C05F	LCSEL	C080

EPSELC	C080	RAM2WP	C080	ROM2WE	C081	ROM2WP	C082	RAM2WE	C083
RAM1WP	C088	ROM1WE	C089	ROM1WP	C08A	RAM1WE	C08B	CLRROM	CFFF
RUNAS	D566	CHKCOM	DEBE	PTRGET	DFE3	LISASTRT	E000	STRINI	E3D5
PRNTAX	F941	RSETADR1	FA62	INIT	FB2F	VTAB	FC22	CLREOP	FC42
HOME	FC58	CLREOL	FC9C	CROUT	FD8E	PRBYTE	FDDA	PRHEX	FDE3
COUT	FDED	SETNORM	FE84	SETKBD	FE89	INPORT	FE8B	SETVID	FE93
OUTPORT	FE95	RSETADR2	FF59	MONITOR	FF65	EOSVCTRS	FFFA	SYNC.L	0004
ROM.L	3000	ROM.D	D000	DOSL.L	2100	DOSL.D	9F00	DOSH.L	2A00
DOSH.D	BE00	LLISA8.L	01C0	LLISA8.D	0900	LISA81.L	2800	LISA81.D	D000
LISA82.L	1000	LISA82.D	D000	LISA83.L	0640	LISA83.D	B7C0	SETUP8.L	1914
SETUP8.D	0900	RAMDSK.L	1B04	RAMDSK.D	4000	FID.L	1316	FID.D	0900
CLK.L	064A	CLK.D	0900	LIST.L	06D9	LIST.D	8800	DEBUG	0001
HWCARD	0001	SRCHALL	00F0	CATALOG	0800	DOSLPRMS	0804	DOSHPRMS	0815
LISAPRMS	0826	LISA1PRM	0837	LISA2PRM	0846	LISA3PRM	0855	SETUPRMS	0864
RMDSKPRM	0872	FIDPARMS	0887	CLKPARMS	0891	LISTPRMS	08A1	ROMPARMS	08BB
CATPARMS	08CA	HWSLOT	0007	HWSLOT16	0070	HWSLOT16	00C7	LABEL	0000
EPASEOS	0C00	ASEXIT	0C50	BINEXIT	0C80	RTNEXIT	0CA8	EPOFF	0CB8
EPUSER1	0CC0	EPUSER2	0CC8	EPMAPEOS	0CD0	EPBINEOS	0CE0	EPEOS	0CF0
EPBINTXT	0CF8	EOS	1000	MAPEOS	10B0	USERRTN1	1118	MAIN	1135
MAIN2	134C	MAIN3	134F	SELC	1375	SELCERR	13DD	AHNDLR	13F2
BHNDLR	13F9	BHNDLR2	13FE	CHNDLR	140F	USERRTN2	1416	DHNDLR	1419
EHNDLR	1423	FHNDLR	142E	GHNDLR	1431	HHNDLR	1440	IHNDLR	1449
JHNDLR	1452	KHNDLR	145B	LHNDLR	1464	MHNDLR	146D	NHNDLR	1476
OHNDLR	147C	PHNDLR	147F	QHNDLR	14AB	RHNDLR	14B4	SHNDLR	14BD
THNDLR	14C6	UHNDLR	14CF	VHNDLR	14D8	WHNDLR	14E1	XHNDLR	14EA
YHNDLR	14F3	ZHNDLR	14FC	DOEOSDCB	150A	EXITAS	1521	EXITBIN	1524
EXITRTN	1527	EXECTEXT	1540	DOEXEC	1551	DOEXEC2	155F	EOSCAT	1569
EOSCAT2	156C	CATHDR	1619	CATFTR	165D	CHKCAT	16B4	SHOWCAT	16DE
SELCFIL	173A	SELCFIL2	175F	CKCATCMD	17D0	GETFILE2	180A	GETFILE	1811
SHOWFILE	1824	RUNLOAD	18EF	RUNFILE	190F	LOADFILE	1918	LOADTEXT	1921
EXECAS	19C5	EXECBIN1	19F5	EXECBIN2	19FE	EXECBIN0	1A04	ASEOS	1A16
DOASFILE	1A8B	DOASCAT	1AB3	SAVFILE	1AD7	RTNCLC	1B64	GETEPRNG	1B66
GETRANGE	1B6D	SAVPARM2	1B94	SAVPARM	1B96	GETASVAL	1B9E	SETASPTR	1BA9
BINEOS	1BB6	BINEOS2	1BD2	BINDONE	1BF5	BINLOAD	1C0E	BINRUN	1C0F
BINCAT	1C45	DOZCOFF	1CA5	SETANNUN	1CD4	DOZCON	1CE1	DOZCRSET	1CFC
DOZCREAD	1D16	DOZCSAVE	1D44	DOZCOPEN	1D7D	ZCCONFIG	1D9D	ZCLOOP	1E7B
ZCDISP	1ED7	EOSBELL	1F32	RDKEY	1F42	GETKEY	1F49	GETHEX	1F5F
GETNUM	1F7E	GETVAL	1FB3	EDITSDV	2000	PRTSDV	2068	PRTDEC	2086
RTN01	209F	NEXTMAP	20A0	CLRUSER	20B8	SETUSER	20BA	PRINT	20C8
PRINT01	20EE	DOVTAB	2120	DOSHOCK	2124	DOPRNTAX	2128	DOPRBYTE	212E
DOPRHEX	2132	DOSPACE	2136	DOCROUT	2139	DOCOUT	213B	DOJSRMEM	213F
JSRMEM	2149	LOADOSL	2152	LOADOSH	216D	LOADOSH2	21AB	COPYROM	21B2
INITCAT	21D2	GETENTRY	21EB	FINDFILE	2246	COPYPRM0	2285	COPYPARM	228A
SELCBANK	2298	READBLK	22B2	READADR	22CC	BUILDMAP	22E7	MAPCODE	2318
MAPPAGE	0110	EPOFFJMP	0157	EPMAPLEN	004A	MOVEEPPBM	2362	EPBMCODE	2385
EPBMPAGE	0110	EPCONFIG	0110	EPMOVE	011D	EPMOVE2	0120	EPBMMOD1	0122
EPJSR	014C	EPBMMOD2	014E	EPRETURN	015A	EPBMMOD3	015D	DOMEMJMP	0162
EPEXEC	0165	EXECMOD1	0168	EXECMOD2	0173	EPBMLEN	007A	LLDCB	23FF
LLCMD	23FF	LLEPROM	2400	LLALTADR	2401	LLSTAT	2403	LLNAMLEN	2404
LLNAMADR	2405	LLNAME	2407	LLNLEN	000A	LLDLEN	0012	LLNADR	02D8
RDDCB	2411	RDCMD	2411	RDEPROM	2412	RDALTADR	2413	RDSTAT	2415
RDNAMLEN	2416	RDNAMADR	2417	RDNAME	2419	RDNLEN	000E	RDDLEN	0016
RDNADR	02D8	FDDCB	2427	FDCMD	2427	FDEPROM	2428	FDALTADR	2429
FDSTAT	242B	FDNAMLEN	242C	FDNAMADR	242D	FDNAME	242F	FDNLEN	0003
FDDLLEN	000B	FDNADR	02D8	ADDCB	2432	ADCMD	2432	ADEPROM	2433
ADALTADR	2434	ADSTAT	2436	ADNAMLEN	2437	ADNAMADR	2438	ADNAME	243A
ADNLEN	0004	ADDLEN	000C	ADNADR	02D8	SCDCB	243E	SCCMD	243E
SCEPROM	243F	SCALTADR	2440	SCSTAT	2442	SCNAMLEN	2443	SCNAMADR	2444
SCNAME	2446	SCNLEN	0009	SCDLEN	0011	SCNADR	02D8	AFDCB	244F
AFCMD	244F	AFEPROM	2450	AFALTADR	2451	AFSTAT	2453	AFNAMLEN	2454
AFNAMADR	2455	AFNAME	2457	AFNLEN	0013	AFDLEN	001B	AFNADR	02D8

LSDCB	246A	LSCMD	246A	LSEEPROM	246B	LSALTADR	246C	LSSTAT	246E
LSNAMLEN	246F	LSNAMADR	2470	LSNAME	2472	LSNLEN	000E	LSDLEN	0016
LSNADR	02D8	EBDCB	2480	EBCMD	2480	EBEPROM	2481	EBALTADR	2482
EBSTAT	2484	EBNAMLEN	2485	EBNAMADR	2486	EBNAME	2488	EBNLEN	000C
EBDLEN	0014	EBNADR	02D8	VTDCB	2494	VTCMD	2494	VTEPROM	2495
VTALTADR	2496	VTSTAT	2498	VTNAMLEN	2499	VTNAMADR	249A	VTNAME	249C
VTNLEN	000C	VTDLLEN	0014	VTNADR	02D8	VODCB	24A8	VOCMD	24A8
VOEPROM	24A9	VOALTADR	24AA	VOSTAT	24AC	VONAMLEN	24AD	VONAMADR	24AE
VONAME	24B0	VONLEN	000E	VODLEN	0016	VONADR	02D8	VDDCB	24BE
VDCMD	24BE	VDEPROM	24BF	VDALTADR	24C0	VDSTAT	24C2	VDNAMLEN	24C3
VDNAMADR	24C4	VDNAME	24C6	VDNLEN	0010	VDDLLEN	0018	VDNADR	02D8
DWDCB	24D6	DWCMD	24D6	DWEPROM	24D7	DWALTADR	24D8	DWSTAT	24DA
DWNAMLEN	24DB	DWNAMADR	24DC	DWNAME	24DE	DWNLEN	000B	DWDLEN	0013
DWNADR	02D8	BFDCB	24E9	BFCMD	24E9	BFEPROM	24EA	BFALTADR	24EB
BFSTAT	24ED	BFNAMLEN	24EE	BFNAMADR	24EF	BFNAME	24F1	BFNLEN	0013
bfdlen	001B	BFNADR	02D8	BMDCB	2504	BMCMD	2504	BMEPROM	2505
BMALTADR	2506	BMSTAT	2508	BMNAMLEN	2509	BMNAMADR	250A	BMNAME	250C
BMNLEN	000B	BMDLEN	0013	BMNADR	02D8	SDDCB	2517	SDCMD	2517
SDEPROM	2518	SDALTADR	2519	SDSTAT	251B	SDNAMLEN	251C	SDNAMADR	251D
SDNAME	251F	SDNLEN	0004	SDDLLEN	000C	SDNADR	02D8	CMDTBL	2523
ADDRTBLL	2557	ADDRTBLLH	2564	LINETBL	2571	DECTBLL	257D	MAPMASKS	2580
ZCDEFLT	2588	ZCSPDTBL	258A	PARMTYPE	258E	LOADTBL	2596	RUNTBL	25A0
TYPETBL	25AA	TYPTEXTS	25B2	TYPETEXT	25B2	TYPEAS	25B7	TYPEBIN0	25C1
TYPEBIN1	25D5	TYPEBIN2	25E7	TYPERSD	25F9	TYPESYS	2602	TYPEPRI	2609
TEXTTBL	2611	TEXTTBLH	2616	CATTEXT	261B	HELOTEXT	2625	SLTTEXT	2631
DRVTEXT	2634	VOLTEXT	2638	EPTEXT	263C	ZCTEXT	2644	ZCTEXT1	2647
ZCTEXT2	264A	ZCTEXT3	265A	ZCTEXT4	2662	ZCTEXT5	266A	ZCTEXT6	2670
SYNCBYS	2676	EOSEND	00FA	NMIRTN	26FA	RESET	26FC	IRQRTN	26FE
DOSL.O	2000	DOSH.O	4100	LISA81.O	6B00	LEFTOVER	1300	LISA82.O	9300
LISA83.O	A300	SETUP8.O	A940	LLISA8.O	C254	RAMDSK.O	C414	FID.O	DF18
CLK.O	F22E	LIST.O	F878	UNUSED	00AF				

Symbols alphabetically sorted:

ADALTADR	2434	ADCMD	2432	ADDCB	2432	ADDLEN	000C	ADDRBUFR	02AA
ADDRTBLLH	2564	ADDRTBLL	2557	ADEPROM	2433	ADNADR	02D8	ADNAMADR	2438
ADNAME	243A	ADNAMLEN	2437	ADNLEN	0004	ADSTAT	2436	AFALTADR	2451
AFCMD	244F	AFDCB	244F	AFDLEN	001B	AFEPROM	2450	AFNADR	02D8
AFNAMADR	2455	AFNAME	2457	AFNAMLEN	2454	AFNLEN	0013	AFSTAT	2453
AHNDLR	13F2	ALTCHOFF	C00E	ANN1OFF	C058	ANN2OFF	C05A	ANN3ON	C05D
ANN4ON	C05F	APLSOFT	0002	APPLTYPE	0293	AREG	0008	ASAV	0018
ASEOS	1A16	ASEXIT	0C50	ASONERR	00D8	ASPADR	02D8	ASPCMD	02D0
ASPEND	00AF	ASPFIL	02D6	ASPFILS	02D8	ASPGMST	0067	ASPNUM	02D6
ASPNUM4	0004	ASPNUM5	0005	ASPNUM6	0006	ASPPARMS	02DA	ASPRADRS	02D0
ASPRNUM	02A0	ASPSRCH	02D4	ASPSTAT	02D2	ASSTATUS	02A1	ASVARS	0069
AUXZPOFF	C008	BANKMASK	001F	BANKSIZE	2000	BFALTADR	24EB	BFCMD	24E9
BFDCB	24E9	bfdlen	001B	BFEPROM	24EA	BFNADR	02D8	BFNAMADR	24EF
BFNAME	24F1	BFNAMLEN	24EE	BFNLEN	0013	BFSTAT	24ED	BHNDLR	13F9
BHNDLR2	13FE	BINADR	095C	BINARY0	0004	BINARY1	0008	BINARY2	0010
BINCAT	1C45	BINCMD	095E	BINDONE	1BF5	BINEOS	1BB6	BINEOS2	1BD2
BINEPN	095F	BINEXIT	0C80	BINFADR	0964	BINFALT	0960	BINFLEN	0963
BINJMP	0959	BINLOAD	1C0E	BINRUN	1C0F	BINSTAT	0962	BMALTADR	2506
BMCMD	2504	BMDCB	2504	BMDLEN	0013	BMEPROM	2505	BMNADR	02D8
BMNAMADR	250A	BMNAME	250C	BMNAMLEN	2509	BMNLEN	000B	BMSTAT	2508
BUILDMAP	22E7	C3ROMOFF	C00A	C3ROMON	C00B	CATALOG	0800	CATCMD	0003
CATFTR	165D	CATHDR	1619	CATPARMS	08CA	CATTEXT	261B	CH	0024
CHKCAT	16B4	CHKCOM	DEBE	CHNDLR	140F	CHRADR	00B8	CHRGOT	00B7
CKCATCMD	17D0	CLK.D	0900	CLK.L	064A	CLK.O	F22E	CLKPARMS	0891
CLREOL	FC9C	CLREOP	FC42	CLRKEY	C010	CLRROM	CFFF	CLRUSER	20B8
CMDPTR	00EE	CMDTBL	2523	CNTRCMD	0059	COMMA	00AC	COPYPARM	228A

COPYPRM0	2285	COPYROM	21B2	COUT	FD8E	CROUT	FD8E	CSWL	0036
CTRLC	0083	CTRLD	0084	CTRLS	0093	CURLIN	0075	CV	0025
CVMASK	001F	CXROMOFF	C006	CXROMON	C007	DARROW	008A	DCBBUFR	02D0
DCBCMD	02D0	DCBEPN	02D1	DCBFADR	02D6	DCBFALT	02D2	DCBFLEN	02D5
DCBSIZE	0008	DCBSTAT	02D4	DEBUG	0001	DECTBLL	257D	DHNDLR	1419
DOASCAT	1AB3	DOASFILE	1A8B	DOCOUT	213B	DOCROUT	2139	DOEOSDCB	150A
DOEXEC	1551	DOEXEC2	155F	DOJSRMEM	213F	DOLLAR	00A4	DOMEMJMP	0162
DOPRBYTE	212E	DOPRHEX	2132	DOPRNTAX	2128	DOSCOLD	03D3	DOSH.D	BE00
DOSH.L	2A00	DOSH.O	4100	DOSHOK	2124	DOSHPRMS	0815	DOSL.D	9F00
DOSL.L	2100	DOSL.O	2000	DOSLPRMS	0804	DOSPACE	2136	DOSWARM	03D0
DOVTAB	2120	DOZCOFF	1CA5	DOZCON	1CE1	DOZCOPEN	1D7D	DOZCREAD	1D16
DOZCRSET	1CFC	DOZCSAVE	1D44	DRIVE	00EC	DRVTEXT	2634	DSCTMP	009D
DSTPTR	002E	DSTVAL	02B6	DWALTADR	24D8	DWCMD	24D6	DWDCB	24D6
DWDLEN	0013	DWEPROM	24D7	DWNADR	02D8	DWNAMADR	24DC	DWNAME	24DE
DWNAMLEN	24DB	DWNLEN	000B	DWSTAT	24DA	EBALTADR	2482	EBCMD	2480
EBDCB	2480	EBDLEN	0014	EBEPROM	2481	EBNADR	02D8	EBNAMADR	2486
EBNAME	2488	EBNAMLEN	2485	EBNLEN	000C	EBSTAT	2484	EDITSDV	2000
EHNDLR	1423	ENDCAT	0000	ENTRYLEN	0020	EOLCMD	0057	EOPCMD	0058
EOS	1000	EOSBELL	1F32	EOSCAT	1569	EOSCAT2	156C	EOSEND	00FA
EOSLDCB	095E	EOSVCTRS	FFFA	EPASEOS	0C00	EPBANK	0295	EPBINEOS	0CE0
EPBINTXT	0CF8	EPBMCODE	2385	EPBMLEN	007A	EPBMMOD1	0122	EPBMMOD2	014E
EPBMMOD3	015D	EPBMPAGE	0110	EPCARD	0001	EPCONFIG	0110	EPEND	0299
EPEOS	0CF0	EPEXEC	0165	EPJSR	014C	EPMAPEOS	0CD0	EPMAPLEN	004A
EPMASK	000F	EPMOVE	011D	EPMOVE2	0120	EPNMBR	0294	EPOFF	0CB8
EPOFFJMP	0157	EPOFFVAL	0080	EPONVAL	0000	EPRETURN	015A	EPSEARCH	02A2
EPSELC	C080	EPSLOT	0291	EPSRCH	00F0	EPSTRT	0298	EPTEXT	263C
EPUSER1	0CC0	EPUSER2	0CC8	EPUSR	0008	ERR00	0000	ERR01	0001
ERR02	0002	ERR03	0003	ERR04	0004	ERR05	0005	ESCAPE	009B
EXECAS	19C5	EXECBIN0	1A04	EXECBIN1	19F5	EXECBIN2	19FE	EXECMOD1	0168
EXECMOD2	0173	EXECPTR	00FA	EXECTEXT	1540	EXITAS	1521	EXITBIN	1524
EXITRTN	1527	EXTERNAL	0002	FDALTADR	2429	FDCMD	2427	FDDCB	2427
FDDLEN	000B	FDEPROM	2428	FDNADR	02D8	FDNAMADR	242D	FDNAME	242F
FDNAMLEN	242C	FDNLEN	0003	FDSTAT	242B	FHNDLR	142E	FID.D	0900
FID.L	1316	FID.O	DF18	FIDPARMS	0887	FILECNT	02D2	FILELEN	02A3
FILENAME	02B8	FILEND	0974	FILENTRY	02B0	FILEPNUM	02B0	FILETYPE	02B1
FILNAM	0966	FILTYPE	02D7	FINDEP	0900	FINDERR	080C	FINDFILE	2246
FIRSTIME	02D1	FLENGTH	029E	GENPTR	00CE	GETASVAL	1B9E	GETENTRY	21EB
GETEPRNG	1B66	GETFILE	1811	GETFILE2	180A	GETHEX	1F5F	GETKEY	1F49
GETNUM	1F7E	GETRANGE	1B6D	GETVAL	1FB3	GHNDLR	1431	HELOTEXT	2625
HHNDLR	1440	HIRESOFF	C056	HOME	FC58	HOMECDM	0055	HOOKDOS	03EA
HOOKSLT	C010	HWCARD	0001	HWSLOT	0007	HWSLOT16	0070	HWSLOT CX	00C7
IHNDLR	1449	INDENT	0005	INDEX	02D8	INIT	FB2F	INITCAT	21D2
INITCMD	0052	INITDOS	BFF8	INPORT	FE8B	INPUT	0200	INTERNAL	0001
IRQRTN	26FE	JHNDLR	1452	JSRMEM	2149	KBDCMD	0054	KEY	C000
KHNDLR	145B	KSWL	0038	LABEL	0000	LARROW	0088	LCSELC	C080
LEFTOVER	1300	LENPTR	002C	LENVAL	02B4	LHNDLR	1464	LINE00	0400
LINE01	0480	LINE02	0500	LINE03	0580	LINE04	0600	LINE05	0680
LINE06	0700	LINE07	0780	LINE08	0428	LINE09	04A8	LINE10	0528
LINE11	05A8	LINE12	0628	LINE13	06A8	LINE14	0728	LINE15	07A8
LINE16	0450	LINE17	04D0	LINE18	0550	LINE19	05D0	LINE20	0650
LINE21	06D0	LINE22	0750	LINE23	07D0	LINETBL	2571	LISA1PRM	0837
LISA2PRM	0846	LISA3PRM	0855	LISA81.D	D000	LISA81.L	2800	LISA81.O	6B00
LISA82.D	D000	LISA82.L	1000	LISA82.O	9300	LISA83.D	B7C0	LISA83.L	0640
LISA83.O	A300	LISAPRMS	0826	LISA8TRT	E000	LIST.D	8800	LIST.L	06D9
LIST.O	F878	LISTPRMS	08A1	LLALTADR	2401	LLCMD	23FF	LLDCB	23FF
LLDLEN	0012	LLEPROM	2400	LLISA8.D	0900	LLISA8.L	01C0	LLISA8.O	C254
LLNADR	02D8	LLNAMADR	2405	LLNAME	2407	LLNAMLEN	2404	LLNLEN	000A
LLSTAT	2403	LOADCMD	0001	LOADFILE	1918	LOADOSH	216D	LOADOSH2	21AB
LOADOSL	2152	LOADTBL	2596	LOADTEXT	1921	LSALTADR	246C	LSCMD	246A
LSDCB	246A	LSDLEN	0016	LSEEPROM	246B	LSNADR	02D8	LSNAMADR	2470
LSNAME	2472	LSNAMLEN	246F	LSNLEN	000E	LSSTAT	246E	LSTOPNTY	02D4

LWRCASE	00E0	LWRMASK	00DF	MAIN	1135	MAIN2	134C	MAIN3	134F
MAPCODE	2318	MAPEOS	10B0	MAPMASKS	2580	MAPPAGE	0110	MAXASNUM	000C
MAXCH	0050	MEMJMP	02A6	MENUMASK	001F	MHNDLR	146D	MINCV	0060
MNGUSER	BFF6	MONITOR	FF65	MOVEEPBM	2362	MSBCLR	007F	MSBSET	0080
MSLOT	00EB	NAMESIZE	0018	NEGONE	00FF	NEXTMAP	20A0	NHNDLR	1476
NMIRTN	26FA	NORMCMD	0051	NTRYEND	02D6	NTRYSTRT	02D5	NUMIN	029C
NUMNTRY	02D3	NUMSCRN	02D0	NUMSELC	029D	OHNDLR	147C	OUTPORT	FE95
PAGE08	0800	PAGE09	0900	PAGE10N	C054	PAGE20	2000	PAGE60	6000
PAGE80	8000	PAGE9F	9F00	PAGEBE	BE00	PAGEC0	C000	PAGEC1	C100
PAGEC7	C700	PAGED0	D000	PAGEDE	DE00	PAGEE0	E000	PAGEE1	E100
PAGEE7	E700	PAGEE8	E800	PAGEEA	EA00	PAGESIZE	0100	PARMSIZE	0006
PARMTYPE	258E	PHNDLR	147F	PRBYTE	FDDA	PRHEX	FDE3	PRIMARY	0080
PRINT	20C8	PRINT01	20EE	PRISLOT	0290	PRNTAX	F941	PRNTMASK	000F
PRNTPTR	00FC	PROMPT	0033	PRTDEC	2086	PRTSDV	2068	PTRGET	DFE3
PWRUP0	0478	PWRUP1	0578	PWRUP2	0678	PWRUP3	0778	QHNDLR	14AB
QLCARD	0000	QLMASK	0007	QLOFFVAL	0010	QLSRCH	0070	RAM1WE	C08B
RAM1WP	C088	RAM2WE	C083	RAM2WP	C080	RAMDSK.D	4000	RAMDSK.L	1B04
RAMDSK.O	C414	RAMRDOFF	C002	RAMWROFF	C004	RARROW	0095	RDALTADR	2413
RDCMD	2411	RDCXROM	C015	RDDCB	2411	RDDLLEN	0016	RDEPROM	2412
RDKEY	1F42	RDNADR	02D8	RDNAMADR	2417	RDNAME	2419	RDNAMLEN	2416
RDNLLEN	000E	RDSTAT	2415	READADR	22CC	READBLK	22B2	RESERVED	0020
RESET	26FC	RETURN	008D	RHNDLR	14B4	RMDSKPRM	0872	ROM.D	D000
ROM.L	3000	ROM1WE	C089	ROM1WP	C08A	ROM2WE	C081	ROM2WP	C082
ROMPARMS	08BB	RSETADR1	FA62	RSETADR2	FF59	RTN01	209F	RTNCLC	1B64
RTNCMD	0050	RTNEXIT	0CA8	RTNTYPE	0296	RUNAS	D566	RUNCMD	0002
RUNFILE	190F	RUNFLAG	029F	RUNLOAD	18EF	RUNMODE	00FF	RUNTBL	25A0
SAVFILE	1AD7	SAVPARM	1B96	SAVPARM2	1B94	SCALTADR	2440	SCCMD	243E
SCDCB	243E	SCDLEN	0011	SCEPROM	243F	SCNADR	02D8	SCNAMADR	2444
SCNAME	2446	SCNAMLEN	2443	SCNLLEN	0009	SCSTAT	2442	SDALTADR	2519
SDCMD	2517	SDDCB	2517	SDDLLEN	000C	SDEPROM	2518	SDNADR	02D8
SDNAMADR	251D	SDNAME	251F	SDNAMLEN	251C	SDNLLEN	0004	SDSTAT	251B
SELC	1375	SELCBANK	2298	SELCERR	13DD	SELCFIL2	175F	SELCFIL1	173A
SETANNUN	1CD4	SETASPTR	1BA9	SETKBD	FE89	SETNORM	FE84	SETUP8.D	0900
SETUP8.L	1914	SETUP8.O	A940	SETUPRMS	0864	SETUSER	20BA	SETVID	FE93
SHNDLR	14BD	SHOWCAT	16DE	SHOWFILE	1824	SLOT	02A5	SLOT16	02A4
SLOT3	0003	SLOTJMP	02A8	SLOTMAP	0292	SLTTEXT	2631	SPACE	00A0
SPKR	C030	SRCHALL	00F0	SRCPTR	002A	SRCVAL	02B2	STACK	0100
STARTAS	0801	STKCODE	0110	STR80OFF	C000	STRINI	E3D5	SYNC.L	0004
SYNCBUFR	02AC	SYNCBYT0	00C4	SYNCBYT1	00B8	SYNCBYT2	0090	SYNCBYT3	00ED
SYNCBYTS	2676	SYSTEM	0040	TABVCMD	0056	TEMPVAL	0297	TESTCNT	0020
TEXTFILE	0001	TEXTON	C051	TEXTTBLH	2616	TEXTTBLI	2611	THNDLR	14C6
TYPEAS	25B7	TYPEBIN0	25C1	TYPEBIN1	25D5	TYPEBIN2	25E7	TYPEPRI	2609
TYPERSD	25F9	TYPESYS	2602	TYPETBL	25AA	TYPETEXT	25B2	TYPTEXTS	25B2
UARROW	008B	UHNDLR	14CF	UHOOKSLT	C018	UNUSED	00AF	USERRTN1	1118
USERRTN2	1416	VALUMASK	000F	VDALTADR	24C0	VDCMD	24BE	VDDCB	24BE
VDDLLEN	0018	VDEPROM	24BF	VDNADR	02D8	VDNAMADR	24C4	VDNAME	24C6
VDNAMLEN	24C3	VDNLLEN	0010	VDSTAT	24C2	VHNDLR	14D8	VID80OFF	C00C
VIDCMD	0053	VOALTADR	24AA	VOCMD	24A8	VODCB	24A8	VODLEN	0016
VOEPROM	24A9	VOLTEXT	2638	VOLUME	00ED	VONADR	02D8	VONAMADR	24AE
VONAME	24B0	VONAMLEN	24AD	VONLEN	000E	VOSTAT	24AC	VTAB	FC22
VTALTADR	2496	VT CMD	2494	VTDCB	2494	VTDLLEN	0014	VTEPROM	2495
VTNADR	02D8	VTNAMADR	249A	VTNAME	249C	VTNAMLEN	2499	VTNLLEN	000C
VTSTAT	2498	WHNDLR	14E1	WNDBTM	0023	WNDLFT	0020	WNDTOP	0022
WNDWDTH	0021	XHNDLR	14EA	XMODE	04FB	XREG	0006	XSAV	0016
YHNDLR	14F3	YREG	0007	YSAV	0017	ZCACHE	029B	ZCCACHE	C05F
ZCCONFIG	1D9D	ZCCTRL	C05A	ZCDEFLT	2588	ZCDELAY	C05E	ZCDISP	1ED7
ZCLOCK	00A5	ZCLOOP	1E7B	ZCNSPEED	0004	ZCOFFVAL	0010	ZCONVAL	0000
ZCOPTNS	000C	ZCSETBL	02D0	ZCSLOTS	C05C	ZCSPDTBL	258A	ZCSPEED	C05D
ZCSTAT	0010	ZCSTATS	C05B	ZCTEXT	2644	ZCTEXT1	2647	ZCTEXT2	264A
ZCTEXT3	265A	ZCTEXT4	2662	ZCTEXT5	266A	ZCTEXT6	2670	ZCUNLOCK	005A
ZERO	0000	ZHNDLR	14FC	ZSTATUS	029A				

Symbols numerically sorted:

ZERO	0000	ZCONVAL	0000	QLCARD	0000	LABEL	0000	ERR00	0000
EPONVAL	0000	ENDCAT	0000	TEXTFILE	0001	LOADCMD	0001	INTERNAL	0001
HWCARD	0001	ERR01	0001	EPCARD	0001	DEBUG	0001	RUNCMD	0002
EXTERNAL	0002	ERR02	0002	APLSOFT	0002	SLOT3	0003	FDNLEN	0003
ERR03	0003	CATCMD	0003	ZCNSPEED	0004	SYNC.L	0004	SDNLEN	0004
ERR04	0004	BINARY0	0004	ASPNUM4	0004	ADNLEN	0004	INDENT	0005
ERR05	0005	ASPNUM5	0005	XREG	0006	PARMSIZE	0006	ASPNUM6	0006
YREG	0007	QLMASK	0007	HWSLOT	0007	EPUSR	0008	DCBSIZE	0008
BINARY1	0008	AREG	0008	SCNLEN	0009	LLNLEN	000A	FDDLEN	000B
DWNLEN	000B	BMNLEN	000B	ZCOPTNS	000C	VTNLEN	000C	SDDLEN	000C
MAXASNUM	000C	EBNLEN	000C	ADDLEN	000C	VONLEN	000E	RDNLEN	000E
LSNLEN	000E	VALUMASK	000F	PRNTMASK	000F	EPMASK	000F	ZCSTAT	0010
ZCOFFVAL	0010	VDNLEN	0010	QLOFFVAL	0010	BINARY2	0010	SCDLEN	0011
LLDLEN	0012	DWDLEN	0013	BMDLEN	0013	BFNLEN	0013	AFNLEN	0013
VTDLN	0014	EBDLN	0014	XSAV	0016	VODLEN	0016	RDDLEN	0016
LSDLEN	0016	YSAV	0017	VDDLLEN	0018	NAMESIZE	0018	ASAV	0018
BFDLEN	001B	AFDLN	001B	MENUMASK	001F	CVMASK	001F	BANKMASK	001F
WNDLFT	0020	TESTCNT	0020	RESERVED	0020	ENTRYLEN	0020	WNDWDTH	0021
WNDTOP	0022	WNBDM	0023	CH	0024	CV	0025	SRCPTR	002A
LENPTR	002C	DSTPTR	002E	PROMPT	0033	CSWL	0036	KSWL	0038
SYSTEM	0040	EPMAPLEN	004A	RTNCMD	0050	MAXCH	0050	NORMCMD	0051
INITCMD	0052	VIDCMD	0053	KBDCMD	0054	HEMCEMD	0055	TABVCMD	0056
EOLCMD	0057	EOPCMD	0058	CNTRCMD	0059	ZCUNLOCK	005A	MINCV	0060
ASPGMST	0067	ASVARS	0069	QLSRCH	0070	HWSLOT16	0070	CURLIN	0075
EPBMLEN	007A	MSBCLR	007F	PRIMARY	0080	MSBSET	0080	EPOFFVAL	0080
CTRLC	0083	CTRLD	0084	LARROW	0088	DARROW	008A	UARROW	008B
RETURN	008D	SYNBYT2	0090	CTRLS	0093	RARROW	0095	ESCAPE	009B
DSCTMP	009D	SPACE	00A0	DOLLAR	00A4	ZCLOCK	00A5	COMMA	00AC
UNUSED	00AF	ASPEND	00AF	CHRGOT	00B7	SYNBYT1	00B8	CHRADR	00B8
SYNBYT0	00C4	HWSLOTX	00C7	GENPTR	00CE	ASONERR	00D8	LWRMASK	00DF
LWRCASE	00E0	MSLOT	00EB	DRIVE	00EC	VOLUME	00ED	SYNBYT3	00ED
CMDPTR	00EE	SRCHALL	00F0	EPSRCH	00F0	EXECPTR	00FA	EOSEND	00FA
PRNTPTR	00FC	RUNMODE	00FF	NEGONE	00FF	STACK	0100	PAGESIZE	0100
STKCODE	0110	MAPPAGE	0110	EPCONFIG	0110	EPBMPAGE	0110	EPMOVE	011D
EPMOVE2	0120	EPBMOD1	0122	EPJSR	014C	EPBMOD2	014E	EPOFFJMP	0157
EPRETURN	015A	EPBMOD3	015D	DOMEMJMP	0162	EPEXEC	0165	EXECMOD1	0168
EXECMOD2	0173	LLISA8.L	01C0	INPUT	0200	PRISLOT	0290	EPSLOT	0291
SLOTMAP	0292	APPLTYPE	0293	EPNMBR	0294	EPBANK	0295	RTNTYPE	0296
TEMPVAL	0297	EPSTRT	0298	EPEND	0299	ZSTATUS	029A	ZCACHE	029B
NUMIN	029C	NUMSEL	029D	FLENGTH	029E	RUNFLAG	029F	ASPRNUM	02A0
ASSTATUS	02A1	EPSEARCH	02A2	FILELEN	02A3	SLOT16	02A4	SLOT	02A5
MEMJMP	02A6	SLOTJMP	02A8	ADDRBUFR	02AA	SYNBUFR	02AC	FILEPNUM	02B0
FILENTRY	02B0	FILETYPE	02B1	SRCVAL	02B2	LENVAL	02B4	DSTVAL	02B6
FILENAME	02B8	ZCSETBL	02D0	NUMSCRN	02D0	DCBCMD	02D0	DCBBUFR	02D0
ASPRADRS	02D0	ASPCMD	02D0	FIRSTIME	02D1	DCBEPN	02D1	FILECNT	02D2
DCBFALT	02D2	ASPSTAT	02D2	NUMNTRY	02D3	LSTOPNTY	02D4	DCBSTAT	02D4
ASPSRCH	02D4	NTRYSTRT	02D5	DCBFLEN	02D5	NTRYEND	02D6	DCBFADR	02D6
ASPNUM	02D6	ASPFILE	02D6	FILTYPE	02D7	VTNADR	02D8	VONADR	02D8
VDNADR	02D8	SDNADR	02D8	SCNADR	02D8	RDNADR	02D8	LSNADR	02D8
LLNADR	02D8	INDEX	02D8	FDNADR	02D8	EBNADR	02D8	DWNADR	02D8
BMNADR	02D8	BFNADR	02D8	ASPFILS	02D8	ASPADR	02D8	AFNADR	02D8
ADNADR	02D8	ASPPARMS	02DA	DOSWARM	03D0	DOSCOLD	03D3	HOOKDOS	03EA
LINE00	0400	LINE08	0428	LINE16	0450	PWRUP0	0478	LINE01	0480
LINE09	04A8	LINE17	04D0	XMODE	04FB	LINE02	0500	LINE10	0528
LINE18	0550	PWRUP1	0578	LINE03	0580	LINE11	05A8	LINE19	05D0
LINE04	0600	LINE12	0628	LISA83.L	0640	CLK.L	064A	LINE20	0650
PWRUP2	0678	LINE05	0680	LINE13	06A8	LINE21	06D0	LIST.L	06D9

LINE06	0700	LINE14	0728	LINE22	0750	PWRUP3	0778	LINE07	0780
LINE15	07A8	LINE23	07D0	PAGE08	0800	CATALOG	0800	STARTAS	0801
DOSLPRMS	0804	FINDERR	080C	DOSHPRMS	0815	LISAPRMS	0826	LISA1PRM	0837
LISA2PRM	0846	LISA3PRM	0855	SETUPRMS	0864	RMDSKPRM	0872	FIDPARMS	0887
CLKPARMS	0891	LISTPRMS	08A1	ROMPARMS	08BB	CATPARMS	08CA	SETUP8.D	0900
PAGE09	0900	LLISA8.D	0900	FINDEP	0900	FID.D	0900	CLK.D	0900
BINJMP	0959	BINADR	095C	EOSLDCB	095E	BINCMD	095E	BINEPN	095F
BINFALT	0960	BINSTAT	0962	BINFLEN	0963	BINFADR	0964	FILNAM	0966
FILEND	0974	EPASEOS	0C00	ASEXIT	0C50	BINEXIT	0C80	RTNEXIT	0CA8
EPOFF	0CB8	EPUSER1	0CC0	EPUSER2	0CC8	EPMAPEOS	0CD0	EPBINEOS	0CE0
EPEOS	0CF0	EPBINTXT	0CF8	LISA82.L	1000	EOS	1000	MAPEOS	10B0
USERRTN1	1118	MAIN	1135	LEFTOVER	1300	FID.L	1316	MAIN2	134C
MAIN3	134F	SELC	1375	SELCERR	13DD	AHNDLR	13F2	BHNDLR	13F9
BHNDLR2	13FE	CHNDLR	140F	USERRTN2	1416	DHNDLR	1419	EHNDLR	1423
FHNDLR	142E	GHNDLR	1431	HHNDLR	1440	IHNDLR	1449	JHNDLR	1452
KHNDLR	145B	LHNDLR	1464	MHNDLR	146D	NHNDLR	1476	OHNDLR	147C
PHNDLR	147F	QHNDLR	14AB	RHNDLR	14B4	SHNDLR	14BD	THNDLR	14C6
UHNDLR	14CF	VHNDLR	14D8	WHNDLR	14E1	XHNDLR	14EA	YHNDLR	14F3
ZHNDLR	14FC	DOEOSDCB	150A	EXITAS	1521	EXITBIN	1524	EXITRTN	1527
EXETEXT	1540	DOEXEC	1551	DOEXEC2	155F	EOSCAT	1569	EOSCAT2	156C
CATHDR	1619	CATFTR	165D	CHKCAT	16B4	SHOWCAT	16DE	SELCFILE	173A
SELCFIL2	175F	CKCATCMD	17D0	GETFILE2	180A	GETFILE	1811	SHOWFILE	1824
RUNLOAD	18EF	RUNFILE	190F	SETUP8.L	1914	LOADFILE	1918	LOADTEXT	1921
EXECAS	19C5	EXECBIN1	19F5	EXECBIN2	19FE	EXECBIN0	1A04	ASEOS	1A16
DOASFILE	1A8B	DOASCAT	1AB3	SAVFILE	1AD7	RAMDSK.L	1B04	RTNCLC	1B64
GETEPRNG	1B66	GETRANGE	1B6D	SAVPARM2	1B94	SAVPARM	1B96	GETASVAL	1B9E
SETASPTR	1BA9	BINEOS	1BB6	BINEOS2	1BD2	BINDONE	1BF5	BINLOAD	1C0E
BINRUN	1C0F	BINCAT	1C45	DOZCOFF	1CA5	SETANNUN	1CD4	DOZCON	1CE1
DOZCRSET	1CFC	DOZCREAD	1D16	DOZCSAVE	1D44	DOZCOPEN	1D7D	ZCCONFIG	1D9D
ZCLOOP	1E7B	ZCDISP	1ED7	EOSBELL	1F32	RDKEY	1F42	GETKEY	1F49
GETHEX	1F5F	GETNUM	1F7E	GETVAL	1FB3	PAGE20	2000	EDITSDV	2000
DOSL.O	2000	BANKSIZE	2000	PRTSDV	2068	PRTDEC	2086	RTN01	209F
NEXTMAP	20A0	CLRUSER	20B8	SETUSER	20BA	PRINT	20C8	PRINT01	20EE
DOSL.L	2100	DOVTAB	2120	DOSHOO	2124	DOPRNTAX	2128	DOPRBYTE	212E
DOPRHEX	2132	DOSPACE	2136	DOCROUT	2139	DOCOUT	213B	DOJSRMEM	213F
JSRMEM	2149	LOADOSL	2152	LOADOSH	216D	LOADOSH2	21AB	COPYROM	21B2
INITCAT	21D2	GETENTRY	21EB	FINDFILE	2246	COPYPRM0	2285	COPYPARM	228A
SELCBANK	2298	READBLK	22B2	READADR	22CC	BUILDMAP	22E7	MAPCODE	2318
MOVEEPBM	2362	EPBMCODE	2385	LLDCB	23FF	LLCMD	23FF	LLEPROM	2400
LLALTADR	2401	LLSTAT	2403	LLNAMLEN	2404	LLNAMADR	2405	LLNAME	2407
RDDCB	2411	RDCMD	2411	RDEPROM	2412	RDALTADR	2413	RDSTAT	2415
RDNAMLEN	2416	RDNAMADR	2417	RDNAME	2419	FDDCB	2427	FDCMD	2427
FDEPROM	2428	FDALTADR	2429	FDSTAT	242B	FDNAMLEN	242C	FDNAMADR	242D
FDNAME	242F	ADDCB	2432	ADCMD	2432	ADEPROM	2433	ADALTADR	2434
ADSTAT	2436	ADNAMLEN	2437	ADNAMADR	2438	ADNAME	243A	SCDCB	243E
SCCMD	243E	SCEPROM	243F	SCALTADR	2440	SCSTAT	2442	SCNAMLEN	2443
SCNAMADR	2444	SCNAME	2446	AFDCB	244F	AFCMD	244F	AFEPROM	2450
AFALTADR	2451	AFSTAT	2453	AFNAMLEN	2454	AFNAMADR	2455	AFNAME	2457
LSDCB	246A	LSCMD	246A	LSEPROM	246B	LSALTADR	246C	LSSTAT	246E
LSNAMLEN	246F	LSNAMADR	2470	LSNAME	2472	EBDCB	2480	EBCMD	2480
EBEPROM	2481	EBALTADR	2482	EBSTAT	2484	EBNAMLEN	2485	EBNAMADR	2486
EBNAME	2488	VTDCB	2494	VTCMD	2494	VTEPROM	2495	VTALTADR	2496
VTSTAT	2498	VTNAMLEN	2499	VTNAMADR	249A	VTNAME	249C	VODCB	24A8
VOCMD	24A8	VOEPROM	24A9	VOALTADR	24AA	VOSTAT	24AC	VONAMLEN	24AD
VONAMADR	24AE	VONAME	24B0	VDDCB	24BE	VDCMD	24BE	VDEPROM	24BF
VDALTADR	24C0	VDSTAT	24C2	VDNAMLEN	24C3	VDNAMADR	24C4	VDNAME	24C6
DWDCB	24D6	DWCMD	24D6	DWEPROM	24D7	DWALTADR	24D8	DWSTAT	24DA
DWNAMLEN	24DB	DWNAMADR	24DC	DWNAME	24DE	BFDCB	24E9	BFCMD	24E9
BFEPROM	24EA	BFALTADR	24EB	BFSTAT	24ED	BFNAMLEN	24EE	BFNAMADR	24EF
BFNAME	24F1	BMDCB	2504	BMCMD	2504	BMEPROM	2505	BMALTADR	2506
BMSTAT	2508	BMNAMLEN	2509	BMNAMADR	250A	BMNAME	250C	SDDCB	2517

SDCMD	2517	SDEPROM	2518	SDALTADR	2519	SDSTAT	251B	SDNAMLEN	251C
SDNAMADR	251D	SDNAME	251F	CMDTBL	2523	ADDRTBLL	2557	ADDRTBLH	2564
LINETBL	2571	DECTBLL	257D	MAPMASKS	2580	ZCDEFLT	2588	ZCSPDTBL	258A
PARMTYPE	258E	LOADTBL	2596	RUNTBL	25A0	TYPETBL	25AA	TYPTEXTS	25B2
TYPETEXT	25B2	TYPEAS	25B7	TYPEBIN0	25C1	TYPEBIN1	25D5	TYPEBIN2	25E7
TYPERESD	25F9	TYPESYS	2602	TYPEPRI	2609	TEXTTBLL	2611	TEXTTBLH	2616
CATTEXT	261B	HELOTEXT	2625	SLTTEXT	2631	DRVTEXT	2634	VOLTEXT	2638
EPTEXT	263C	ZCTEXT	2644	ZCTEXT1	2647	ZCTEXT2	264A	ZCTEXT3	265A
ZCTEXT4	2662	ZCTEXT5	266A	ZCTEXT6	2670	SYNCBYTS	2676	NMIRTN	26FA
RESET	26FC	IRQRTN	26FE	LISA81.L	2800	DOSH.L	2A00	ROM.L	3000
RAMDSK.D	4000	DOSH.O	4100	PAGE60	6000	LISA81.O	6B00	PAGE80	8000
LIST.D	8800	LISA82.O	9300	PAGE9F	9F00	DOSL.D	9F00	LISA83.O	A300
SETUP8.O	A940	LISA83.D	B7C0	PAGEBE	BE00	DOSH.D	BE00	MNGUSER	BFF6
INITDOS	BFF8	STR80OFF	C000	PAGEC0	C000	KEY	C000	RAMRDOFF	C002
RAMWROFF	C004	CXROMOFF	C006	CXROMON	C007	AUXZPOFF	C008	C3ROMOFF	C00A
C3ROMON	C00B	VID80OFF	C00C	ALTCHOFF	C00E	HOOKSLT	C010	CLRKEY	C010
RDCXROM	C015	UHOOKSLT	C018	SPKR	C030	TEXTON	C051	PAGE1ON	C054
HIRESOFF	C056	ANN1OFF	C058	ZCCTRL	C05A	ANN2OFF	C05A	ZCSTATS	C05B
ZCSLOTS	C05C	ZCSPEED	C05D	ANN3ON	C05D	ZCDELAY	C05E	ZCCACHE	C05F
ANN4ON	C05F	RAM2WP	C080	LCSELC	C080	EPSELC	C080	ROM2WE	C081
ROM2WP	C082	RAM2WE	C083	RAM1WP	C088	ROM1WE	C089	ROM1WP	C08A
RAM1WE	C08B	PAGEC1	C100	LLISA8.O	C254	RAMDSK.O	C414	PAGEC7	C700
CLRROM	CFFF	ROM.D	D000	PAGED0	D000	LISA82.D	D000	LISA81.D	D000
RUNAS	D566	PAGEDE	DE00	CHKCOM	DEBE	FID.O	DF18	PTRGET	DFE3
PAGEE0	E000	LISASTRT	E000	PAGEE1	E100	STRINI	E3D5	PAGEE7	E700
PAGEE8	E800	PAGEEA	EA00	CLK.O	F22E	LIST.O	F878	PRNTAX	F941
RSETADR1	FA62	INIT	FB2F	VTAB	FC22	CLREOP	FC42	HOME	FC58
CLREOL	FC9C	CROUT	FD8E	PRBYTE	FDDA	PRHEX	FDE3	COUT	FDED
SETNORM	FE84	SETKBD	FE89	INPORT	FE8B	SETVID	FE93	OUTPORT	FE95
RSETADR2	FF59	MONITOR	FF65	EOSVCTRS	FFFA				