

!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 3E FD    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 C4       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	SYNCTYPE0	equ	\$C4
00B8	156	SYNCTYPE1	equ	\$B8
0090	157	SYNCTYPE2	equ	\$90
00ED	158	SYNCTYPE3	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          ;
0000          7  DEBUG      equ 0
0000          8  HWCARD     equ QLCARD
0974          9          ;
0974         10          ;
0974         11          .if DEBUG
0974         12          org PAGE08
0974         13          .el
E000         14          org PAGEE0
E000         15          .fi
E000         16          ;
E000         17          obj PAGE08
E000         18          usr
E000         19          ;
E000         20          ;
E000         21          .if HWCARD
0070         22  SRCHALL   equ EPSRCH
E000         23          .el
0070         24  SRCHALL   equ QLSRCH
E000         25          .fi
E000         26          ;
E000         27          ;
E000         28  CATALOG:
E000         29          ;
E000 C4         30          byt SYNCBYT0
E001 B8         31          byt SYNCBYT1
E002 90         32          byt SYNCBYT2
E003 ED         33          byt SYNCBYT3
E004         34          ;
E004         35          ;
E004 44         36  DOSLPRMS byt SYSTEM|BINARY0
E005 00 20      37          adr DOSL.O
E007 00 21      38          adr DOSL.L
E009 00 9F      39          adr DOSL.D
E00B 44 4F 53   40          dci `DOS4.5.05L`
E00E 34 2E 35
E011 2E 30 35
E014 CC
E015          41          ;
E015          42          ;
E015 5C         43  DOSHPRMS byt SYSTEM|BINARY0|BINARY1|BINARY2
E016 00 41      44          adr DOSH.O
E018 00 2A      45          adr DOSH.L
E01A 00 BE      46          adr DOSH.D
E01C 44 4F 53   47          dci `DOS4.5.06H`
E01F 34 2E 35
E022 2E 30 36
E025 C8
E026          48          ;
E026          49          ;
E026 84         50  LISAPRMS byt PRIMARY|BINARY0
E027 54 C2      51          adr LLISA8.O
E029 C0 01      52          adr LLISA8.L
E02B 00 09      53          adr LLISA8.D
E02D 4C 4F 41   54          dci `LOADLISA80`
```

```

E030 44 4C 49
E033 53 41 38
E036 B0
E037          55 ;
E037          56 ;
E037 48      57 LISA1PRM byt SYSTEM|BINARY1
E038 00 6B   58          adr LISA81.O
E03A 00 28   59          adr LISA81.L
E03C 00 D0   60          adr LISA81.D
E03E 4C 49 53 61          dci `LISA80.1`
E041 41 38 30
E044 2E B1
E046          62 ;
E046          63 ;
E046 50      64 LISA2PRM byt SYSTEM|BINARY2
E047 00 93   65          adr LISA82.O
E049 00 10   66          adr LISA82.L
E04B 00 D0   67          adr LISA82.D
E04D 4C 49 53 68          dci `LISA80.2`
E050 41 38 30
E053 2E B2
E055          69 ;
E055          70 ;
E055 44      71 LISA3PRM byt SYSTEM|BINARY0
E056 00 A3   72          adr LISA83.O
E058 40 06   73          adr LISA83.L
E05A C0 B7   74          adr LISA83.D
E05C 4C 49 53 75          dci `LISA80.3`
E05F 41 38 30
E062 2E B3
E064          76 ;
E064          77 ;
E064 04      78 SETUPRMS byt BINARY0
E065 40 A9   79          adr SETUP8.O
E067 14 19   80          adr SETUP8.L
E069 00 09   81          adr SETUP8.D
E06B 53 45 54 82          dci `SETUP80`
E06E 55 50 38
E071 B0
E072          83 ;
E072          84 ;
E072 04      85 RMDSKPRM byt BINARY0
E073 14 C4   86          adr RAMDSK.O
E075 04 1B   87          adr RAMDSK.L
E077 00 40   88          adr RAMDSK.D
E079 52 61 6D 89          dci `RamDisk Config`
E07C 44 69 73
E07F 6B 20 43
E082 6F 6E 66
E085 69 E7
E087          90 ;
E087          91 ;
E087 04      92 FIDPARMS byt BINARY0
E088 18 DF   93          adr FID.O
E08A 16 13   94          adr FID.L
E08C 00 09   95          adr FID.D
E08E 46 49 C4 96          dci `FID`
E091          97 ;
E091          98 ;
E091 04      99 CLKPARMS byt BINARY0
E092 2E F2  100          adr CLK.O

```

```

E094 4A 06      101      adr CLK.L
E096 00 09      102      adr CLK.D
E098 53 65 74   103      dci 'Set Clock'
E09B 20 43 6C
E09E 6F 63 EB
E0A1           104      ;
E0A1           105      ;
E0A1 04         106      LISTPRMS byt BINARY0
E0A2 78 F8      107      adr LIST.O
E0A4 D9 06      108      adr LIST.L
E0A6 00 88      109      adr LIST.D
E0A8 41 70 70   110      dci 'Applesoft Formatter'
E0AB 6C 65 73
E0AE 6F 66 74
E0B1 20 46 6F
E0B4 72 6D 61
E0B7 74 74 65
E0BA F2
E0BB           111      ;
E0BB           112      ;
E0BB 20         113      ROMPARMS byt RESERVED
E0BC 00 D0      114      adr ROM.D
E0BE 00 30      115      adr ROM.L
E0C0 00 D0      116      adr ROM.D
E0C2 52 4F 4D   117      dci 'ROM Copy'
E0C5 20 43 6F
E0C8 70 F9
E0CA           118      ;
E0CA           119      ;
E0CA 20         120      CATPARMS byt RESERVED
E0CB 00 00      121      adr CATALOG&BANKMASK
E0CD 04 00      122      adr SYNC.L
E0CF AC 02      123      adr SYNCBUFR
E0D1 43 61 74   124      dci 'Catalog Sync'
E0D4 61 6C 6F
E0D7 67 20 53
E0DA 79 6E E3
E0DD           125      ;
E0DD           126      ;
E0DD 00         127      byt ENDCAT
E0DE           128      ;
E0DE           129      ;
E0DE           130      dfs PAGESIZE-*)&NEGONE,NEGONE
E100           131      ;
E100           132      ;
E100           133      .if DEBUG
E100           134      .el
E100           135      phs PAGEC1
C100           136      .fi
C100           137      ;
C100           138      ;
C100           139      icl "SLOT1.L"

```

```

LLOAD SLOT1.L,A$4000

```

```

C100          1          ttl "EOS+ Source Code, SLOT1.L"
C100          2          ;
C100          3          ;
C100          4          ; SLOT1.L
C100          5          ;
C100          6          ;
0001          7          HWSLOT    let 1
0010          8          HWSLOT16 let $10
00C1          9          HWSLOT16 let $C1
C100         10          ;
0000         11          LABEL    let 0
C100         12          ;
C100         13          ;
C100         14          ; This is the generic code that is assembled specifically
C100         15          ; for each of the seven slots in which an EPROM card may
C100         16          ; reside.
C100         17          ;
C100         18          ; Interface to process an ASEOS command.
C100         19          ;
C100         20          .if LABEL
C100         21          EPASEOS   sta ASAV
C100         22          .el
C100 85 18     23          sta ASAV
C102         24          .fi
C102         25          ;
C102 18       26          clc
C103 90 03    27          bcc >0          ; always taken
C105         28          ;
C105         29          ;
C105         30          ; Insert TESTROM verification code here. When the CXRESET
C105         31          ; routine is entered, it calls TSTROMCD to test for a ROM
C105         32          ; card. TSTROMCD calls TESTROM that looks for 0x38xx18 at
C105         33          ; 0xC305. If found, C3ROMOFF is enabled.
C105         34          ;
C105 38       35          sec
C106 90 00    36          bcc *+2
C108         37          dfs !-1
C107 18       38          clc
C108         39          ;
C108         40          ;
C108 86 16    41          ^0          stx XSAV
C10A 84 17    42          sty YSAV
C10C         43          ;
C10C A5 76    44          lda CURLIN+1
C10E C9 FF    45          cmp #RUNMODE
C110 D0 03    46          bne >2
C112         47          ;
C112 A5 18    48          ^1          lda ASAV          ; recall A-reg
C114         49          ;
C114 60       50          rts
C115         51          ;
C115 A2 00    52          ^2          ldx #ZERO
C117         53          ;
C117 8E A0 02 54          ^3          stx ASPRNUM
C11A         55          ;
C11A 20 B7 00 56          jsr CHRGOT
C11D F0 21    57          beq >4
C11F         58          ;
C11F 20 BE DE 59          jsr CHKCOM
C122         60          ;

```

```

C122 20 B7 00      61      jsr CHRGOT
C125 F0 19         62      beq >4
C127              63      ;
C127 C9 2C         64      cmp #COMMA&MSBCLR
C129 F0 15         65      beq >4
C12B              66      ;
C12B 20 E3 DF      67      jsr PTRGET
C12E              68      ;
C12E AE A0 02      69      ldx ASPRNUM
C131 E0 0C         70      cpx #MAXASNUM      ; too many parameters
C133 F0 DD         71      beq <1
C135              72      ;
C135 9D D0 02      73      sta ASPRADRS,X
C138              74      ;
C138 98           75      tya
C139 9D D1 02      76      sta ASPRADRS+1,X
C13C              77      ;
C13C E8           78      inx
C13D E8           79      inx
C13E              80      ;
C13E D0 D7         81      bne <3      ; always taken
C140              82      ;
C140 4E A0 02      83      ^4      lsr ASPRNUM
C143 F0 CD         84      beq <1      ; no parameters
C145              85      ;
C145 A9 00         86      lda #EPONVAL
C147 8D 90 C0      87      sta EPSELC+HWSLOT16
C14A              88      ;
C14A              89      ;
C14A              90      ; Initialize X-reg with this slot number and enter ASEOS.
C14A              91      ;
C14A A2 01         92      ldx #HWSLOT
C14C              93      ;
C14C 4C 10 F2      94      jmp ASEOS
C14F              95      ;
C14F              96      ;
C14F              97      dfs $50-*)&NEGONE,NEGONE
C150              98      ;
C150              99      ;
C150             100      ; Exit for ASEOS and to RUN Applesoft files. If a ZipChip
C150             101      ; is present, flush cache, enable ZipChip, and return to
C150             102      ; the external caller or address in MEMJMP.
C150             103      ;
C150             104      .if LABEL
C150             105      ASEXIT   lda ZSTATUS
C150             106      .el
C150 AD 9A 02      107      lda ZSTATUS
C153             108      .fi
C153             109      ;
C153 30 14         110      bmi >2
C155             111      ;
C155 A0 00         112      ldy #ZERO
C157 A9 60         113      lda /PAGE60
C159             114      ;
C159 84 CE         115      sty GENPTR
C15B 85 CF         116      sta GENPTR+1
C15D             117      ;
C15D B1 CE         118      ^1      lda (GENPTR),Y
C15F             119      ;
C15F C8           120      iny
C160 D0 FB         121      bne <1

```

```

C162          122 ;
C162 E6 CF    123      inc GENPTR+1
C164 10 F7    124      bpl <1
C166          125 ;
C166 20 DB F4 126      jsr DOZCON
C169          127 ;
C169          128      .if HWCARD
C169          129 ^2      lda #EPOFFVAL
C169          130      .el
C169 A9 10    131 ^2      lda #QLOFFVAL
C16B          132      .fi
C16B          133 ;
C16B 8D 90 C0 134      sta EPSELC+HWSLOT16
C16E          135 ;
C16E A5 18    136      lda ASAV
C170 A6 16    137      ldx XSAV
C172 A4 17    138      ldy YSAV
C174          139 ;
C174 6C A6 02 140      jmp (MEMJMP)
C177          141 ;
C177          142 ;
C177          143      dfs $80-*)&NEGONE,NEGONE
C180          144 ;
C180          145 ;
C180          146 ; Exit for BINEOS and to RUN binary files. If a ZipChip is
C180          147 ; present, flush cache, enable ZipChip, push DOSWARM onto
C180          148 ; the stack, load the address for EPBINEOS, and return to
C180          149 ; the external caller or address in MEMJMP.
C180          150 ;
C180          151 ; Load the Y-reg/A-reg with the address of EPBINEOS in
C180          152 ; order for the current file in memory to process more
C180          153 ; DCBs if other EPROM files need to be loaded into memory.
C180          154 ;
C180          155      .if LABEL
C180          156 BINEXIT  lda ZSTATUS
C180          157      .el
C180 AD 9A 02 158      lda ZSTATUS
C183          159      .fi
C183          160 ;
C183 30 14    161      bmi >2
C185          162 ;
C185 A0 00    163      ldy #ZERO
C187 A9 60    164      lda /PAGE60
C189          165 ;
C189 84 CE    166      sty GENPTR
C18B 85 CF    167      sta GENPTR+1
C18D          168 ;
C18D B1 CE    169 ^1      lda (GENPTR),Y
C18F          170 ;
C18F C8       171      iny
C190 D0 FB    172      bne <1
C192          173 ;
C192 E6 CF    174      inc GENPTR+1
C194 10 F7    175      bpl <1
C196          176 ;
C196 20 DB F4 177      jsr DOZCON
C199          178 ;
C199 A2 FF    179 ^2      ldx #NEGONE
C19B 9A       180      txs
C19C          181 ;
C19C A9 03    182      lda /DOSWARM-1

```

```

C19E 48          183          pha
C19F            184          ;
C19F A9 CF      185          lda #DOSWARM-1
C1A1 48          186          pha
C1A2            187          ;
C1A2 A0 E0      188          ld y #EPBINEOS
C1A4 A9 C1      189          lda #HWSLOT16
C1A6            190          ;
C1A6            191          ;
C1A6 EA         192          nop
C1A7 EA         193          nop
C1A8            194          ;
C1A8            195          ;
C1A8            196          ; Exit from EOS with the assumption that a return to EOS
C1A8            197          ; will be made by means of EPMAPEOS. The X-reg must be
C1A8            198          ; used in order to turn this EPROM card OFF.
C1A8            199          ;
C1A8            200          .if LABEL
C1A8            201          RTNEXIT:
C1A8            202          .fi
C1A8            203          ;
C1A8            204          .if HWCARD
C1A8            205          ldx #EPOFFVAL
C1A8            206          .el
C1A8 A2 10      207          ldx #QLOFFVAL
C1AA            208          .fi
C1AA            209          ;
C1AA 8E 90 C0   210          stx EPSELC+HWSLOT16
C1AD            211          ;
C1AD A2 10      212          ldx #HWSLOT16
C1AF            213          ;
C1AF 6C A6 02   214          jmp (MEMJMP)
C1B2            215          ;
C1B2            216          ;
C1B2            217          dfs $B8-*&NEGONE,NEGONE
C1B8            218          ;
C1B8            219          ;
C1B8            220          ; Special entrance in order to turn this EPROM card OFF.
C1B8            221          ;
C1B8            222          .if LABEL
C1B8            223          EPOFF:
C1B8            224          .fi
C1B8            225          ;
C1B8            226          .if HWCARD
C1B8            227          lda #EPOFFVAL
C1B8            228          .el
C1B8 A9 10      229          lda #QLOFFVAL
C1BA            230          .fi
C1BA            231          ;
C1BA 8D 90 C0   232          sta EPSELC+HWSLOT16
C1BD            233          ;
C1BD 60         234          rts
C1BE            235          ;
C1BE            236          ;
C1BE            237          dfs $C0-*&NEGONE,NEGONE
C1C0            238          ;
C1C0            239          ;
C1C0            240          ; Return from DOS CMDUSER 1 command.
C1C0            241          ;
C1C0            242          .if LABEL
C1C0            243          EPUSER1 lda #EPONVAL

```

```

C1C0          244          .el
C1C0 A9 00    245          lda #EPONVAL
C1C2          246          .fi
C1C2          247          ;
C1C2 8D 90 C0 248          sta EPSELC+HWSLOT16
C1C5          249          ;
C1C5 4C 12 E9 250          jmp USERRTN1
C1C8          251          ;
C1C8          252          ;
C1C8          253          ; Return from DOS CMDUSER 2 command.
C1C8          254          ;
C1C8          255          .if LABEL
C1C8          256 EPUSER2  lda #EPONVAL
C1C8          257          .el
C1C8 A9 00    258          lda #EPONVAL
C1CA          259          .fi
C1CA          260          ;
C1CA 8D 90 C0 261          sta EPSELC+HWSLOT16
C1CD          262          ;
C1CD 4C 10 EC 263          jmp USERRTN2
C1D0          264          ;
C1D0          265          ;
C1D0          266          ; Entrance into the EOS+ card at the EPMAPEOS location by
C1D0          267          ; using SLOTMAP from the EOS mapping function.
C1D0          268          ;
C1D0          269          .if LABEL
C1D0          270 EPMAPEOS  lda #EPONVAL
C1D0          271          .el
C1D0 A9 00    272          lda #EPONVAL
C1D2          273          .fi
C1D2          274          ;
C1D2 8D 90 C0 275          sta EPSELC+HWSLOT16
C1D5          276          ;
C1D5          277          ;
C1D5          278          ; Initialize X-reg with this slot number and enter MAPEOS.
C1D5          279          ;
C1D5 A2 01    280          ldx #HWSLOT
C1D7          281          ;
C1D7 4C AD E8 282          jmp MAPEOS
C1DA          283          ;
C1DA          284          ;
C1DA          285          dfs $E0-*&NEGONE,NEGONE
C1E0          286          ;
C1E0          287          ;
C1E0          288          ; Interface entrance in order to process a BINEOS command
C1E0          289          ; that is contained within a BINEOS DCB. The Y-reg and the
C1E0          290          ; A-reg must contain the address of the eight byte DCB.
C1E0          291          ; The X-reg is initialized with the slot number of this
C1E0          292          ; EPROM card before calling BINEOS. BINEOS first calls
C1E0          293          ; DOZCOFF and MOVEEPBM before processing the BINEOS DCB.
C1E0          294          ; The X-reg must be used in order to turn this EPROM card
C1E0          295          ; ON.
C1E0          296          ;
C1E0          297          .if LABEL
C1E0          298 EPBINEOS  ldx #EPONVAL
C1E0          299          .el
C1E0 A2 00    300          ldx #EPONVAL
C1E2          301          .fi
C1E2          302          ;
C1E2 8E 90 C0 303          stx EPSELC+HWSLOT16
C1E5          304          ;

```



```

C1E5      305 ;
C1E5      306 ; Initialize X-reg with this slot number and then enter
C1E5      307 ; BINEOS.
C1E5      308 ;
C1E5 A2 01 309      ldx #HWSLOT
C1E7      310 ;
C1E7 4C B0 F3 311      jmp BINEOS
C1EA      312 ;
C1EA      313 ;
C1EA      314      dfs $F0-*)&NEGONE,NEGONE
C1F0      315 ;
C1F0      316 ;
C1F0      317 ; Manual entrance into EOS for this EPROM card.  If this
C1F0      318 ; EPROM card is not the highest priority EPROM card,
C1F0      319 ; control will pass to the highest priority EPROM card
C1F0      320 ; after the EPROM card mapping function has completed.
C1F0      321 ; CTRL-N may be used in order to select the desired
C1F0      322 ; EPROM card if there are mulitple EPROM cards in an
C1F0      323 ; Apple computer.
C1F0      324 ;
C1F0      325      .if LABEL
C1F0      326 EPEOS    lda #EPONVAL
C1F0      327      .el
C1F0 A9 00 328      lda #EPONVAL
C1F2      329      .fi
C1F2      330 ;
C1F2 8D 90 C0 331      sta EPSELC+HWSLOT16
C1F5      332 ;
C1F5 4C 00 E8 333      jmp EOS
C1F8      334 ;
C1F8      335 ;
C1F8      336 ; This is the ASCII text that is compared to EPTEXT in
C1F8      337 ; order to determine if the slot that is being tested
C1F8      338 ; contains an EPROM card.
C1F8      339 ;
C1F8      340      .if LABEL
C1F8      341 EPBINTXT asc "EPBINEOS"
C1F8      342      .el
C1F8 C5 D0 C2 343      asc "EPBINEOS"
C1FB C9 CE C5
C1FE CF D3
C200      344      .fi
C200      345 ;
C200      346 ;
C200      347      icl "SLOT2.L"

```

LLOAD SLOT2.L,A\$4000

```

C200          1          ttl "EOS+ Source Code, SLOT2.L"
C200          2          ;
C200          3          ;
C200          4          ; SLOT2.L
C200          5          ;
C200          6          ;
0002          7  HWSLOT    let 2
0020          8  HWSLOT16 let $20
00C2          9  HWSLOT16 let $C2
C200         10          ;
0000         11  LABEL     let 0
C200         12          ;
C200         13          ;
C200         14          ; This is the generic code that is assembled specifically
C200         15          ; for each of the seven slots in which an EPROM card may
C200         16          ; reside.
C200         17          ;
C200         18          ; Interface to process an ASEOS command.
C200         19          ;
C200         20          .if LABEL
C200         21  EPASEOS   sta ASAV
C200         22          .el
C200 85 18         23          sta ASAV
C202         24          .fi
C202         25          ;
C202 18          26          clc
C203 90 03        27          bcc >0                ; always taken
C205         28          ;
C205         29          ;
C205         30          ; Insert TESTROM verification code here. When the CXRESET
C205         31          ; routine is entered, it calls TSTROMCD to test for a ROM
C205         32          ; card. TSTROMCD calls TESTROM that looks for 0x38xx18 at
C205         33          ; 0xC305. If found, C3ROMOFF is enabled.
C205         34          ;
C205 38          35          sec
C206 90 00        36          bcc *+2
C208          37          dfs !-1
C207 18          38          clc
C208          39          ;
C208          40          ;
C208 86 16        41  ^0          stx XSAV
C20A 84 17        42          sty YSAV
C20C          43          ;
C20C A5 76        44          lda CURLIN+1
C20E C9 FF        45          cmp #RUNMODE
C210 D0 03        46          bne >2
C212          47          ;
C212 A5 18        48  ^1          lda ASAV                ; recall A-reg
C214          49          ;
C214 60          50          rts
C215          51          ;
C215 A2 00        52  ^2          ldx #ZERO
C217          53          ;
C217 8E A0 02     54  ^3          stx ASPRNUM
C21A          55          ;
C21A 20 B7 00     56          jsr CHRGOT
C21D F0 21        57          beq >4
C21F          58          ;
C21F 20 BE DE     59          jsr CHKCOM
C222          60          ;

```

```

C222 20 B7 00      61      jsr CHRGOT
C225 F0 19        62      beq >4
C227              63      ;
C227 C9 2C        64      cmp #COMMA&MSBCLR
C229 F0 15        65      beq >4
C22B              66      ;
C22B 20 E3 DF     67      jsr PTRGET
C22E              68      ;
C22E AE A0 02     69      ldx ASPRNUM
C231 E0 0C        70      cpx #MAXASNUM      ; too many parameters
C233 F0 DD        71      beq <1
C235              72      ;
C235 9D D0 02     73      sta ASPRADRS,X
C238              74      ;
C238 98          75      tya
C239 9D D1 02     76      sta ASPRADRS+1,X
C23C              77      ;
C23C E8          78      inx
C23D E8          79      inx
C23E              80      ;
C23E D0 D7       81      bne <3      ; always taken
C240              82      ;
C240 4E A0 02     83      ^4      lsr ASPRNUM
C243 F0 CD        84      beq <1      ; no parameters
C245              85      ;
C245 A9 00        86      lda #EPONVAL
C247 8D A0 C0     87      sta EPSELC+HWSLOT16
C24A              88      ;
C24A              89      ;
C24A              90      ; Initialize X-reg with this slot number and enter ASEOS.
C24A              91      ;
C24A A2 02        92      ldx #HWSLOT
C24C              93      ;
C24C 4C 10 F2     94      jmp ASEOS
C24F              95      ;
C24F              96      ;
C24F              97      dfs $50-*)&NEGONE,NEGONE
C250              98      ;
C250              99      ;
C250            100      ; Exit for ASEOS and to RUN Applesoft files. If a ZipChip
C250            101      ; is present, flush cache, enable ZipChip, and return to
C250            102      ; the external caller or address in MEMJMP.
C250            103      ;
C250            104      .if LABEL
C250            105      ASEXIT  lda ZSTATUS
C250            106      .el
C250 AD 9A 02     107      lda ZSTATUS
C253            108      .fi
C253            109      ;
C253 30 14       110      bmi >2
C255            111      ;
C255 A0 00       112      ldy #ZERO
C257 A9 60       113      lda /PAGE60
C259            114      ;
C259 84 CE       115      sty GENPTR
C25B 85 CF       116      sta GENPTR+1
C25D            117      ;
C25D B1 CE       118      ^1      lda (GENPTR),Y
C25F            119      ;
C25F C8          120      iny
C260 D0 FB       121      bne <1

```

```

C262          122 ;
C262 E6 CF    123      inc GENPTR+1
C264 10 F7    124      bpl <1
C266          125 ;
C266 20 DB F4 126      jsr DOZCON
C269          127 ;
C269          128      .if HWCARD
C269          129 ^2      lda #EPOFFVAL
C269          130      .el
C269 A9 10    131 ^2      lda #QLOFFVAL
C26B          132      .fi
C26B          133 ;
C26B 8D A0 C0 134      sta EPSELC+HWSLOT16
C26E          135 ;
C26E A5 18    136      lda ASAV
C270 A6 16    137      ldx XSAV
C272 A4 17    138      ldy YSAV
C274          139 ;
C274 6C A6 02 140      jmp (MEMJMP)
C277          141 ;
C277          142 ;
C277          143      dfs $80-*)&NEGONE,NEGONE
C280          144 ;
C280          145 ;
C280          146 ; Exit for BINEOS and to RUN binary files. If a ZipChip is
C280          147 ; present, flush cache, enable ZipChip, push DOSWARM onto
C280          148 ; the stack, load the address for EPBINEOS, and return to
C280          149 ; the external caller or address in MEMJMP.
C280          150 ;
C280          151 ; Load the Y-reg/A-reg with the address of EPBINEOS in
C280          152 ; order for the current file in memory to process more
C280          153 ; DCBs if other EPROM files need to be loaded into memory.
C280          154 ;
C280          155      .if LABEL
C280          156 BINEXIT  lda ZSTATUS
C280          157      .el
C280 AD 9A 02 158      lda ZSTATUS
C283          159      .fi
C283          160 ;
C283 30 14    161      bmi >2
C285          162 ;
C285 A0 00    163      ldy #ZERO
C287 A9 60    164      lda /PAGE60
C289          165 ;
C289 84 CE    166      sty GENPTR
C28B 85 CF    167      sta GENPTR+1
C28D          168 ;
C28D B1 CE    169 ^1      lda (GENPTR),Y
C28F          170 ;
C28F C8       171      iny
C290 D0 FB    172      bne <1
C292          173 ;
C292 E6 CF    174      inc GENPTR+1
C294 10 F7    175      bpl <1
C296          176 ;
C296 20 DB F4 177      jsr DOZCON
C299          178 ;
C299 A2 FF    179 ^2      ldx #NEGONE
C29B 9A       180      txs
C29C          181 ;
C29C A9 03    182      lda /DOSWARM-1

```

```

C29E 48          183          pha
C29F          184          ;
C29F A9 CF      185          lda #DOSWARM-1
C2A1 48          186          pha
C2A2          187          ;
C2A2 A0 E0      188          ld y #EPBINEOS
C2A4 A9 C2      189          lda #HWSLOT CX
C2A6          190          ;
C2A6          191          ;
C2A6 EA        192          nop
C2A7 EA        193          nop
C2A8          194          ;
C2A8          195          ;
C2A8          196          ; Exit from EOS with the assumption that a return to EOS
C2A8          197          ; will be made by means of EPMAPEOS. The X-reg must be
C2A8          198          ; used in order to turn this EPROM card OFF.
C2A8          199          ;
C2A8          200          .if LABEL
C2A8          201          RTNEXIT:
C2A8          202          .fi
C2A8          203          ;
C2A8          204          .if HWCARD
C2A8          205          ldx #EPOFFVAL
C2A8          206          .el
C2A8 A2 10      207          ldx #QLOFFVAL
C2AA          208          .fi
C2AA          209          ;
C2AA 8E A0 C0   210          stx EPSELC+HWSLOT16
C2AD          211          ;
C2AD A2 20      212          ldx #HWSLOT16
C2AF          213          ;
C2AF 6C A6 02   214          jmp (MEMJMP)
C2B2          215          ;
C2B2          216          ;
C2B2          217          dfs $B8-*&NEGONE,NEGONE
C2B8          218          ;
C2B8          219          ;
C2B8          220          ; Special entrance in order to turn this EPROM card OFF.
C2B8          221          ;
C2B8          222          .if LABEL
C2B8          223          EPOFF:
C2B8          224          .fi
C2B8          225          ;
C2B8          226          .if HWCARD
C2B8          227          lda #EPOFFVAL
C2B8          228          .el
C2B8 A9 10      229          lda #QLOFFVAL
C2BA          230          .fi
C2BA          231          ;
C2BA 8D A0 C0   232          sta EPSELC+HWSLOT16
C2BD          233          ;
C2BD 60         234          rts
C2BE          235          ;
C2BE          236          ;
C2BE          237          dfs $C0-*&NEGONE,NEGONE
C2C0          238          ;
C2C0          239          ;
C2C0          240          ; Return from DOS CMDUSER 1 command.
C2C0          241          ;
C2C0          242          .if LABEL
C2C0          243          EPUSER1 lda #EPONVAL

```

```

C2C0          244          .el
C2C0 A9 00     245          lda #EPONVAL
C2C2          246          .fi
C2C2          247          ;
C2C2 8D A0 C0  248          sta EPSELC+HWSLOT16
C2C5          249          ;
C2C5 4C 12 E9  250          jmp USERRTN1
C2C8          251          ;
C2C8          252          ;
C2C8          253          ; Return from DOS CMDUSER 2 command.
C2C8          254          ;
C2C8          255          .if LABEL
C2C8          256 EPUSER2  lda #EPONVAL
C2C8          257          .el
C2C8 A9 00     258          lda #EPONVAL
C2CA          259          .fi
C2CA          260          ;
C2CA 8D A0 C0  261          sta EPSELC+HWSLOT16
C2CD          262          ;
C2CD 4C 10 EC  263          jmp USERRTN2
C2D0          264          ;
C2D0          265          ;
C2D0          266          ; Entrance into the EOS+ card at the EPMAPEOS location by
C2D0          267          ; using SLOTMAP from the EOS mapping function.
C2D0          268          ;
C2D0          269          .if LABEL
C2D0          270 EPMAPEOS  lda #EPONVAL
C2D0          271          .el
C2D0 A9 00     272          lda #EPONVAL
C2D2          273          .fi
C2D2          274          ;
C2D2 8D A0 C0  275          sta EPSELC+HWSLOT16
C2D5          276          ;
C2D5          277          ;
C2D5          278          ; Initialize X-reg with this slot number and enter MAPEOS.
C2D5          279          ;
C2D5 A2 02     280          ldx #HWSLOT
C2D7          281          ;
C2D7 4C AD E8  282          jmp MAPEOS
C2DA          283          ;
C2DA          284          ;
C2DA          285          dfs $E0-*&NEGONE,NEGONE
C2E0          286          ;
C2E0          287          ;
C2E0          288          ; Interface entrance in order to process a BINEOS command
C2E0          289          ; that is contained within a BINEOS DCB. The Y-reg and the
C2E0          290          ; A-reg must contain the address of the eight byte DCB.
C2E0          291          ; The X-reg is initialized with the slot number of this
C2E0          292          ; EPROM card before calling BINEOS. BINEOS first calls
C2E0          293          ; DOZCOFF and MOVEEPBM before processing the BINEOS DCB.
C2E0          294          ; The X-reg must be used in order to turn this EPROM card
C2E0          295          ; ON.
C2E0          296          ;
C2E0          297          .if LABEL
C2E0          298 EPBINEOS  ldx #EPONVAL
C2E0          299          .el
C2E0 A2 00     300          ldx #EPONVAL
C2E2          301          .fi
C2E2          302          ;
C2E2 8E A0 C0  303          stx EPSELC+HWSLOT16
C2E5          304          ;

```

```

C2E5      305 ;
C2E5      306 ; Initialize X-reg with this slot number and then enter
C2E5      307 ; BINEOS.
C2E5      308 ;
C2E5 A2 02 309      ldx #HWSLOT
C2E7      310 ;
C2E7 4C B0 F3 311      jmp BINEOS
C2EA      312 ;
C2EA      313 ;
C2EA      314      dfs $F0-*)&NEGONE,NEGONE
C2F0      315 ;
C2F0      316 ;
C2F0      317 ; Manual entrance into EOS for this EPROM card.  If this
C2F0      318 ; EPROM card is not the highest priority EPROM card,
C2F0      319 ; control will pass to the highest priority EPROM card
C2F0      320 ; after the EPROM card mapping function has completed.
C2F0      321 ; CTRL-N may be used in order to select the desired
C2F0      322 ; EPROM card if there are mulitple EPROM cards in an
C2F0      323 ; Apple computer.
C2F0      324 ;
C2F0      325      .if LABEL
C2F0      326 EPEOS    lda #EPONVAL
C2F0      327      .el
C2F0 A9 00 328      lda #EPONVAL
C2F2      329      .fi
C2F2      330 ;
C2F2 8D A0 C0 331      sta EPSELC+HWSLOT16
C2F5      332 ;
C2F5 4C 00 E8 333      jmp EOS
C2F8      334 ;
C2F8      335 ;
C2F8      336 ; This is the ASCII text that is compared to EPTEXT in
C2F8      337 ; order to determine if the slot that is being tested
C2F8      338 ; contains an EPROM card.
C2F8      339 ;
C2F8      340      .if LABEL
C2F8      341 EPBINTXT asc "EPBINEOS"
C2F8      342      .el
C2F8 C5 D0 C2 343      asc "EPBINEOS"
C2FB C9 CE C5
C2FE CF D3
C300      344      .fi
C300      345 ;
C300      346 ;
C300      347      icl "SLOT3.L"

```

```

LLOAD SLOT3.L,A$4000

```

```

C300          1          ttl "EOS+ Source Code, SLOT3.L"
C300          2          ;
C300          3          ;
C300          4          ; SLOT3.L
C300          5          ;
C300          6          ;
0003          7          HWSLOT    let 3
0030          8          HWSLOT16 let $30
00C3          9          HWSLOTCX let $C3
C300         10          ;
0000         11          LABEL    let 0
C300         12          ;
C300         13          ;
C300         14          ; This is the generic code that is assembled specifically
C300         15          ; for each of the seven slots in which an EPROM card may
C300         16          ; reside.
C300         17          ;
C300         18          ; Interface to process an ASEOS command.
C300         19          ;
C300         20          .if LABEL
C300         21          EPASEOS   sta ASAV
C300         22          .el
C300 85 18     23          sta ASAV
C302         24          .fi
C302         25          ;
C302 18       26          clc
C303 90 03     27          bcc >0          ; always taken
C305         28          ;
C305         29          ;
C305         30          ; Insert TESTROM verification code here. When the CXRESET
C305         31          ; routine is entered, it calls TSTROMCD to test for a ROM
C305         32          ; card. TSTROMCD calls TESTROM that looks for 0x38xx18 at
C305         33          ; 0xC305. If found, C3ROMOFF is enabled.
C305         34          ;
C305 38       35          sec
C306 90 00     36          bcc *+2
C308         37          dfs !-1
C307 18       38          clc
C308         39          ;
C308         40          ;
C308 86 16     41          ^0          stx XSAV
C30A 84 17     42          sty YSAV
C30C         43          ;
C30C A5 76     44          lda CURLIN+1
C30E C9 FF     45          cmp #RUNMODE
C310 D0 03     46          bne >2
C312         47          ;
C312 A5 18     48          ^1          lda ASAV          ; recall A-reg
C314         49          ;
C314 60       50          rts
C315         51          ;
C315 A2 00     52          ^2          ldx #ZERO
C317         53          ;
C317 8E A0 02  54          ^3          stx ASPRNUM
C31A         55          ;
C31A 20 B7 00  56          jsr CHRGOT
C31D F0 21     57          beq >4
C31F         58          ;
C31F 20 BE DE  59          jsr CHKCOM
C322         60          ;

```



```

C322 20 B7 00      61      jsr CHRGOT
C325 F0 19         62      beq >4
C327              63      ;
C327 C9 2C         64      cmp #COMMA&MSBCLR
C329 F0 15         65      beq >4
C32B              66      ;
C32B 20 E3 DF      67      jsr PTRGET
C32E              68      ;
C32E AE A0 02      69      ldx ASPRNUM
C331 E0 0C         70      cpx #MAXASNUM      ; too many parameters
C333 F0 DD         71      beq <1
C335              72      ;
C335 9D D0 02      73      sta ASPRADRS,X
C338              74      ;
C338 98           75      tya
C339 9D D1 02      76      sta ASPRADRS+1,X
C33C              77      ;
C33C E8           78      inx
C33D E8           79      inx
C33E              80      ;
C33E D0 D7         81      bne <3      ; always taken
C340              82      ;
C340 4E A0 02      83      ^4      lsr ASPRNUM
C343 F0 CD         84      beq <1      ; no parameters
C345              85      ;
C345 A9 00         86      lda #EPONVAL
C347 8D B0 C0      87      sta EPSELC+HWSLOT16
C34A              88      ;
C34A              89      ;
C34A              90      ; Initialize X-reg with this slot number and enter ASEOS.
C34A              91      ;
C34A A2 03         92      ldx #HWSLOT
C34C              93      ;
C34C 4C 10 F2      94      jmp ASEOS
C34F              95      ;
C34F              96      ;
C34F              97      dfs $50-*)&NEGONE,NEGONE
C350              98      ;
C350              99      ;
C350             100      ; Exit for ASEOS and to RUN Applesoft files. If a ZipChip
C350             101      ; is present, flush cache, enable ZipChip, and return to
C350             102      ; the external caller or address in MEMJMP.
C350             103      ;
C350             104      .if LABEL
C350             105      ASEXIT  lda ZSTATUS
C350             106      .el
C350 AD 9A 02      107      lda ZSTATUS
C353             108      .fi
C353             109      ;
C353 30 14         110      bmi >2
C355             111      ;
C355 A0 00         112      ldy #ZERO
C357 A9 60         113      lda /PAGE60
C359             114      ;
C359 84 CE         115      sty GENPTR
C35B 85 CF         116      sta GENPTR+1
C35D             117      ;
C35D B1 CE         118      ^1      lda (GENPTR),Y
C35F             119      ;
C35F C8           120      iny
C360 D0 FB         121      bne <1

```

```

C362          122 ;
C362 E6 CF    123      inc GENPTR+1
C364 10 F7    124      bpl <1
C366          125 ;
C366 20 DB F4 126      jsr DOZCON
C369          127 ;
C369          128      .if HWCARD
C369          129 ^2      lda #EPOFFVAL
C369          130      .el
C369 A9 10    131 ^2      lda #QLOFFVAL
C36B          132      .fi
C36B          133 ;
C36B 8D B0 C0 134      sta EPSELC+HWSLOT16
C36E          135 ;
C36E A5 18    136      lda ASAV
C370 A6 16    137      ldx XSAV
C372 A4 17    138      ldy YSAV
C374          139 ;
C374 6C A6 02 140      jmp (MEMJMP)
C377          141 ;
C377          142 ;
C377          143      dfs $80-*)&NEGONE,NEGONE
C380          144 ;
C380          145 ;
C380          146 ; Exit for BINEOS and to RUN binary files. If a ZipChip is
C380          147 ; present, flush cache, enable ZipChip, push DOSWARM onto
C380          148 ; the stack, load the address for EPBINEOS, and return to
C380          149 ; the external caller or address in MEMJMP.
C380          150 ;
C380          151 ; Load the Y-reg/A-reg with the address of EPBINEOS in
C380          152 ; order for the current file in memory to process more
C380          153 ; DCBs if other EPROM files need to be loaded into memory.
C380          154 ;
C380          155      .if LABEL
C380          156 BINEXIT  lda ZSTATUS
C380          157      .el
C380 AD 9A 02 158      lda ZSTATUS
C383          159      .fi
C383          160 ;
C383 30 14    161      bmi >2
C385          162 ;
C385 A0 00    163      ldy #ZERO
C387 A9 60    164      lda /PAGE60
C389          165 ;
C389 84 CE    166      sty GENPTR
C38B 85 CF    167      sta GENPTR+1
C38D          168 ;
C38D B1 CE    169 ^1      lda (GENPTR),Y
C38F          170 ;
C38F C8       171      iny
C390 D0 FB    172      bne <1
C392          173 ;
C392 E6 CF    174      inc GENPTR+1
C394 10 F7    175      bpl <1
C396          176 ;
C396 20 DB F4 177      jsr DOZCON
C399          178 ;
C399 A2 FF    179 ^2      ldx #NEGONE
C39B 9A       180      txs
C39C          181 ;
C39C A9 03    182      lda /DOSWARM-1

```

```

C39E 48          183          pha
C39F           184          ;
C39F A9 CF      185          lda #DOSWARM-1
C3A1 48          186          pha
C3A2           187          ;
C3A2 A0 E0      188          ld y #EPBINEOS
C3A4 A9 C3      189          lda #HWSLOT CX
C3A6           190          ;
C3A6           191          ;
C3A6 EA         192          nop
C3A7 EA         193          nop
C3A8           194          ;
C3A8           195          ;
C3A8           196          ; Exit from EOS with the assumption that a return to EOS
C3A8           197          ; will be made by means of EPMAPEOS. The X-reg must be
C3A8           198          ; used in order to turn this EPROM card OFF.
C3A8           199          ;
C3A8           200          .if LABEL
C3A8           201          RTNEXIT:
C3A8           202          .fi
C3A8           203          ;
C3A8           204          .if HWCARD
C3A8           205          ldx #EPOFFVAL
C3A8           206          .el
C3A8 A2 10      207          ldx #QLOFFVAL
C3AA           208          .fi
C3AA           209          ;
C3AA 8E B0 C0   210          stx EPSELC+HWSLOT16
C3AD           211          ;
C3AD A2 30      212          ldx #HWSLOT16
C3AF           213          ;
C3AF 6C A6 02   214          jmp (MEMJMP)
C3B2           215          ;
C3B2           216          ;
C3B2           217          dfs $B8-*&NEGONE,NEGONE
C3B8           218          ;
C3B8           219          ;
C3B8           220          ; Special entrance in order to turn this EPROM card OFF.
C3B8           221          ;
C3B8           222          .if LABEL
C3B8           223          EPOFF:
C3B8           224          .fi
C3B8           225          ;
C3B8           226          .if HWCARD
C3B8           227          lda #EPOFFVAL
C3B8           228          .el
C3B8 A9 10      229          lda #QLOFFVAL
C3BA           230          .fi
C3BA           231          ;
C3BA 8D B0 C0   232          sta EPSELC+HWSLOT16
C3BD           233          ;
C3BD 60         234          rts
C3BE           235          ;
C3BE           236          ;
C3BE           237          dfs $C0-*&NEGONE,NEGONE
C3C0           238          ;
C3C0           239          ;
C3C0           240          ; Return from DOS CMDUSER 1 command.
C3C0           241          ;
C3C0           242          .if LABEL
C3C0           243          EPUSER1 lda #EPONVAL

```

```

C3C0          244          .el
C3C0 A9 00     245          lda #EPONVAL
C3C2          246          .fi
C3C2          247          ;
C3C2 8D B0 C0  248          sta EPSELC+HWSLOT16
C3C5          249          ;
C3C5 4C 12 E9  250          jmp USERRTN1
C3C8          251          ;
C3C8          252          ;
C3C8          253          ; Return from DOS CMDUSER 2 command.
C3C8          254          ;
C3C8          255          .if LABEL
C3C8          256 EPUSER2  lda #EPONVAL
C3C8          257          .el
C3C8 A9 00     258          lda #EPONVAL
C3CA          259          .fi
C3CA          260          ;
C3CA 8D B0 C0  261          sta EPSELC+HWSLOT16
C3CD          262          ;
C3CD 4C 10 EC  263          jmp USERRTN2
C3D0          264          ;
C3D0          265          ;
C3D0          266          ; Entrance into the EOS+ card at the EPMAPEOS location by
C3D0          267          ; using SLOTMAP from the EOS mapping function.
C3D0          268          ;
C3D0          269          .if LABEL
C3D0          270 EPMAPEOS  lda #EPONVAL
C3D0          271          .el
C3D0 A9 00     272          lda #EPONVAL
C3D2          273          .fi
C3D2          274          ;
C3D2 8D B0 C0  275          sta EPSELC+HWSLOT16
C3D5          276          ;
C3D5          277          ;
C3D5          278          ; Initialize X-reg with this slot number and enter MAPEOS.
C3D5          279          ;
C3D5 A2 03     280          ldx #HWSLOT
C3D7          281          ;
C3D7 4C AD E8  282          jmp MAPEOS
C3DA          283          ;
C3DA          284          ;
C3DA          285          dfs $E0-*&NEGONE,NEGONE
C3E0          286          ;
C3E0          287          ;
C3E0          288          ; Interface entrance in order to process a BINEOS command
C3E0          289          ; that is contained within a BINEOS DCB. The Y-reg and the
C3E0          290          ; A-reg must contain the address of the eight byte DCB.
C3E0          291          ; The X-reg is initialized with the slot number of this
C3E0          292          ; EPROM card before calling BINEOS. BINEOS first calls
C3E0          293          ; DOZCOFF and MOVEEPBM before processing the BINEOS DCB.
C3E0          294          ; The X-reg must be used in order to turn this EPROM card
C3E0          295          ; ON.
C3E0          296          ;
C3E0          297          .if LABEL
C3E0          298 EPBINEOS  ldx #EPONVAL
C3E0          299          .el
C3E0 A2 00     300          ldx #EPONVAL
C3E2          301          .fi
C3E2          302          ;
C3E2 8E B0 C0  303          stx EPSELC+HWSLOT16
C3E5          304          ;

```

```

C3E5      305 ;
C3E5      306 ; Initialize X-reg with this slot number and then enter
C3E5      307 ; BINEOS.
C3E5      308 ;
C3E5 A2 03 309      ldx #HWSLOT
C3E7      310 ;
C3E7 4C B0 F3 311      jmp BINEOS
C3EA      312 ;
C3EA      313 ;
C3EA      314      dfs $F0-*)&NEGONE,NEGONE
C3F0      315 ;
C3F0      316 ;
C3F0      317 ; Manual entrance into EOS for this EPROM card.  If this
C3F0      318 ; EPROM card is not the highest priority EPROM card,
C3F0      319 ; control will pass to the highest priority EPROM card
C3F0      320 ; after the EPROM card mapping function has completed.
C3F0      321 ; CTRL-N may be used in order to select the desired
C3F0      322 ; EPROM card if there are mulitple EPROM cards in an
C3F0      323 ; Apple computer.
C3F0      324 ;
C3F0      325      .if LABEL
C3F0      326 EPEOS    lda #EPONVAL
C3F0      327      .el
C3F0 A9 00 328      lda #EPONVAL
C3F2      329      .fi
C3F2      330 ;
C3F2 8D B0 C0 331      sta EPSELC+HWSLOT16
C3F5      332 ;
C3F5 4C 00 E8 333      jmp EOS
C3F8      334 ;
C3F8      335 ;
C3F8      336 ; This is the ASCII text that is compared to EPTEXT in
C3F8      337 ; order to determine if the slot that is being tested
C3F8      338 ; contains an EPROM card.
C3F8      339 ;
C3F8      340      .if LABEL
C3F8      341 EPBINTXT asc "EPBINEOS"
C3F8      342      .el
C3F8 C5 D0 C2 343      asc "EPBINEOS"
C3FB C9 CE C5
C3FE CF D3
C400      344      .fi
C400      345 ;
C400      346 ;
C400      347      icl "SLOT4.L"

```

```

LLOAD SLOT4.L,A$4000

```

```

C400          1          ttl "EOS+ Source Code, SLOT4.L"
C400          2          ;
C400          3          ;
C400          4          ; SLOT4.L
C400          5          ;
C400          6          ;
0004          7          HWSLOT    let 4
0040          8          HWSLOT16 let $40
00C4          9          HWSLOT16 let $C4
C400         10          ;
0001         11          LABEL    let 1
C400         12          ;
C400         13          ;
C400         14          ; This is the generic code that is assembled specifically
C400         15          ; for each of the seven slots in which an EPROM card may
C400         16          ; reside.
C400         17          ;
C400         18          ; Interface to process an ASEOS command.
C400         19          ;
C400         20          .if LABEL
C400 85 18     21          EPASEOS  sta ASAV
C402         22          .el
C402         23          sta ASAV
C402         24          .fi
C402         25          ;
C402 18       26          clc
C403 90 03     27          bcc >0                ; always taken
C405         28          ;
C405         29          ;
C405         30          ; Insert TESTROM verification code here. When the CXRESET
C405         31          ; routine is entered, it calls TSTROMCD to test for a ROM
C405         32          ; card. TSTROMCD calls TESTROM that looks for 0x38xx18 at
C405         33          ; 0xC305. If found, C3ROMOFF is enabled.
C405         34          ;
C405 38       35          sec
C406 90 00     36          bcc *+2
C408         37          dfs !-1
C407 18       38          clc
C408         39          ;
C408         40          ;
C408 86 16     41          ^0      stx XSAV
C40A 84 17     42          sty YSAV
C40C         43          ;
C40C A5 76     44          lda CURLIN+1
C40E C9 FF     45          cmp #RUNMODE
C410 D0 03     46          bne >2
C412         47          ;
C412 A5 18     48          ^1      lda ASAV                ; recall A-reg
C414         49          ;
C414 60       50          rts
C415         51          ;
C415 A2 00     52          ^2      ldx #ZERO
C417         53          ;
C417 8E A0 02  54          ^3      stx ASPRNUM
C41A         55          ;
C41A 20 B7 00  56          jsr CHRGOT
C41D F0 21     57          beq >4
C41F         58          ;
C41F 20 BE DE  59          jsr CHKCOM
C422         60          ;

```

```

C422 20 B7 00      61      jsr CHRGOT
C425 F0 19        62      beq >4
C427              63      ;
C427 C9 2C        64      cmp #COMMA&MSBCLR
C429 F0 15        65      beq >4
C42B              66      ;
C42B 20 E3 DF     67      jsr PTRGET
C42E              68      ;
C42E AE A0 02     69      ldx ASPRNUM
C431 E0 0C        70      cpx #MAXASNUM      ; too many parameters
C433 F0 DD        71      beq <1
C435              72      ;
C435 9D D0 02     73      sta ASPRADRS,X
C438              74      ;
C438 98           75      tya
C439 9D D1 02     76      sta ASPRADRS+1,X
C43C              77      ;
C43C E8           78      inx
C43D E8           79      inx
C43E              80      ;
C43E D0 D7        81      bne <3      ; always taken
C440              82      ;
C440 4E A0 02     83      ^4      lsr ASPRNUM
C443 F0 CD        84      beq <1      ; no parameters
C445              85      ;
C445 A9 00        86      lda #EPONVAL
C447 8D C0 C0     87      sta EPSELC+HWSLOT16
C44A              88      ;
C44A              89      ;
C44A              90      ; Initialize X-reg with this slot number and enter ASEOS.
C44A              91      ;
C44A A2 04        92      ldx #HWSLOT
C44C              93      ;
C44C 4C 10 F2     94      jmp ASEOS
C44F              95      ;
C44F              96      ;
C44F              97      dfs $50-*)&NEGONE,NEGONE
C450              98      ;
C450              99      ;
C450             100      ; Exit for ASEOS and to RUN Applesoft files. If a ZipChip
C450             101      ; is present, flush cache, enable ZipChip, and return to
C450             102      ; the external caller or address in MEMJMP.
C450             103      ;
C450             104      .if LABEL
C450 AD 9A 02     105      ASEXIT  lda ZSTATUS
C453             106      .el
C453             107      lda ZSTATUS
C453             108      .fi
C453             109      ;
C453 30 14        110      bmi >2
C455             111      ;
C455 A0 00        112      ldy #ZERO
C457 A9 60        113      lda /PAGE60
C459             114      ;
C459 84 CE        115      sty GENPTR
C45B 85 CF        116      sta GENPTR+1
C45D             117      ;
C45D B1 CE        118      ^1      lda (GENPTR),Y
C45F             119      ;
C45F C8           120      iny
C460 D0 FB        121      bne <1

```

```

C462          122 ;
C462 E6 CF    123      inc GENPTR+1
C464 10 F7    124      bpl <1
C466          125 ;
C466 20 DB F4 126      jsr DOZCON
C469          127 ;
C469          128      .if HWCARD
C469          129 ^2      lda #EPOFFVAL
C469          130      .el
C469 A9 10    131 ^2      lda #QLOFFVAL
C46B          132      .fi
C46B          133 ;
C46B 8D C0 C0 134      sta EPSELC+HWSLOT16
C46E          135 ;
C46E A5 18    136      lda ASAV
C470 A6 16    137      ldx XSAV
C472 A4 17    138      ldy YSAV
C474          139 ;
C474 6C A6 02 140      jmp (MEMJMP)
C477          141 ;
C477          142 ;
C477          143      dfs $80-*)&NEGONE,NEGONE
C480          144 ;
C480          145 ;
C480          146 ; Exit for BINEOS and to RUN binary files. If a ZipChip is
C480          147 ; present, flush cache, enable ZipChip, push DOSWARM onto
C480          148 ; the stack, load the address for EPBINEOS, and return to
C480          149 ; the external caller or address in MEMJMP.
C480          150 ;
C480          151 ; Load the Y-reg/A-reg with the address of EPBINEOS in
C480          152 ; order for the current file in memory to process more
C480          153 ; DCBs if other EPROM files need to be loaded into memory.
C480          154 ;
C480          155      .if LABEL
C480 AD 9A 02 156 BINEXIT lda ZSTATUS
C483          157      .el
C483          158      lda ZSTATUS
C483          159      .fi
C483          160 ;
C483 30 14    161      bmi >2
C485          162 ;
C485 A0 00    163      ldy #ZERO
C487 A9 60    164      lda /PAGE60
C489          165 ;
C489 84 CE    166      sty GENPTR
C48B 85 CF    167      sta GENPTR+1
C48D          168 ;
C48D B1 CE    169 ^1      lda (GENPTR),Y
C48F          170 ;
C48F C8       171      iny
C490 D0 FB    172      bne <1
C492          173 ;
C492 E6 CF    174      inc GENPTR+1
C494 10 F7    175      bpl <1
C496          176 ;
C496 20 DB F4 177      jsr DOZCON
C499          178 ;
C499 A2 FF    179 ^2      ldx #NEGONE
C49B 9A       180      txs
C49C          181 ;
C49C A9 03    182      lda /DOSWARM-1

```



```

C49E 48          183          pha
C49F            184          ;
C49F A9 CF      185          lda #DOSWARM-1
C4A1 48          186          pha
C4A2            187          ;
C4A2 A0 E0      188          ldy #EPBINEOS
C4A4 A9 C4      189          lda #HWSLOTCTX
C4A6            190          ;
C4A6            191          ;
C4A6 EA         192          nop
C4A7 EA         193          nop
C4A8            194          ;
C4A8            195          ;
C4A8            196          ; Exit from EOS with the assumption that a return to EOS
C4A8            197          ; will be made by means of EPMAPEOS. The X-reg must be
C4A8            198          ; used in order to turn this EPROM card OFF.
C4A8            199          ;
C4A8            200          .if LABEL
C4A8            201          RTNEXIT:
C4A8            202          .fi
C4A8            203          ;
C4A8            204          .if HWCARD
C4A8            205          ldx #EPOFFVAL
C4A8            206          .el
C4A8 A2 10       207          ldx #QLOFFVAL
C4AA            208          .fi
C4AA            209          ;
C4AA 8E C0 C0    210          stx EPSELC+HWSLOT16
C4AD            211          ;
C4AD A2 40       212          ldx #HWSLOT16
C4AF            213          ;
C4AF 6C A6 02    214          jmp (MEMJMP)
C4B2            215          ;
C4B2            216          ;
C4B2            217          dfs $B8-*&NEGONE,NEGONE
C4B8            218          ;
C4B8            219          ;
C4B8            220          ; Special entrance in order to turn this EPROM card OFF.
C4B8            221          ;
C4B8            222          .if LABEL
C4B8            223          EPOFF:
C4B8            224          .fi
C4B8            225          ;
C4B8            226          .if HWCARD
C4B8            227          lda #EPOFFVAL
C4B8            228          .el
C4B8 A9 10       229          lda #QLOFFVAL
C4BA            230          .fi
C4BA            231          ;
C4BA 8D C0 C0    232          sta EPSELC+HWSLOT16
C4BD            233          ;
C4BD 60          234          rts
C4BE            235          ;
C4BE            236          ;
C4BE            237          dfs $C0-*&NEGONE,NEGONE
C4C0            238          ;
C4C0            239          ;
C4C0            240          ; Return from DOS CMDUSER 1 command.
C4C0            241          ;
C4C0            242          .if LABEL
C4C0 A9 00       243          EPUSER1 lda #EPONVAL

```

```

C4C2      244      .el
C4C2      245      lda #EPONVAL
C4C2      246      .fi
C4C2      247      ;
C4C2 8D C0 C0 248      sta EPSELC+HWSLOT16
C4C5      249      ;
C4C5 4C 12 E9 250      jmp USERRTN1
C4C8      251      ;
C4C8      252      ;
C4C8      253      ; Return from DOS CMDUSER 2 command.
C4C8      254      ;
C4C8      255      .if LABEL
C4C8 A9 00 256  EPUSER2  lda #EPONVAL
C4CA      257      .el
C4CA      258      lda #EPONVAL
C4CA      259      .fi
C4CA      260      ;
C4CA 8D C0 C0 261      sta EPSELC+HWSLOT16
C4CD      262      ;
C4CD 4C 10 EC 263      jmp USERRTN2
C4D0      264      ;
C4D0      265      ;
C4D0      266      ; Entrance into the EOS+ card at the EPMAPEOS location by
C4D0      267      ; using SLOTMAP from the EOS mapping function.
C4D0      268      ;
C4D0      269      .if LABEL
C4D0 A9 00 270  EPMAPEOS  lda #EPONVAL
C4D2      271      .el
C4D2      272      lda #EPONVAL
C4D2      273      .fi
C4D2      274      ;
C4D2 8D C0 C0 275      sta EPSELC+HWSLOT16
C4D5      276      ;
C4D5      277      ;
C4D5      278      ; Initialize X-reg with this slot number and enter MAPEOS.
C4D5      279      ;
C4D5 A2 04 280      ldx #HWSLOT
C4D7      281      ;
C4D7 4C AD E8 282      jmp MAPEOS
C4DA      283      ;
C4DA      284      ;
C4DA      285      dfs $E0-*&NEGONE,NEGONE
C4E0      286      ;
C4E0      287      ;
C4E0      288      ; Interface entrance in order to process a BINEOS command
C4E0      289      ; that is contained within a BINEOS DCB. The Y-reg and the
C4E0      290      ; A-reg must contain the address of the eight byte DCB.
C4E0      291      ; The X-reg is initialized with the slot number of this
C4E0      292      ; EPROM card before calling BINEOS. BINEOS first calls
C4E0      293      ; DOZCOFF and MOVEEPBM before processing the BINEOS DCB.
C4E0      294      ; The X-reg must be used in order to turn this EPROM card
C4E0      295      ; ON.
C4E0      296      ;
C4E0      297      .if LABEL
C4E0 A2 00 298  EPBINEOS  ldx #EPONVAL
C4E2      299      .el
C4E2      300      ldx #EPONVAL
C4E2      301      .fi
C4E2      302      ;
C4E2 8E C0 C0 303      stx EPSELC+HWSLOT16
C4E5      304      ;

```

```

C4E5      305 ;
C4E5      306 ; Initialize X-reg with this slot number and then enter
C4E5      307 ; BINEOS.
C4E5      308 ;
C4E5 A2 04 309      ldx #HWSLOT
C4E7      310 ;
C4E7 4C B0 F3 311      jmp BINEOS
C4EA      312 ;
C4EA      313 ;
C4EA      314      dfs $F0-*)&NEGONE,NEGONE
C4F0      315 ;
C4F0      316 ;
C4F0      317 ; Manual entrance into EOS for this EPROM card.  If this
C4F0      318 ; EPROM card is not the highest priority EPROM card,
C4F0      319 ; control will pass to the highest priority EPROM card
C4F0      320 ; after the EPROM card mapping function has completed.
C4F0      321 ; CTRL-N may be used in order to select the desired
C4F0      322 ; EPROM card if there are mulitple EPROM cards in an
C4F0      323 ; Apple computer.
C4F0      324 ;
C4F0      325      .if LABEL
C4F0 A9 00 326 EPEOS   lda #EPONVAL
C4F2      327      .el
C4F2      328      lda #EPONVAL
C4F2      329      .fi
C4F2      330 ;
C4F2 8D C0 C0 331      sta EPSELC+HWSLOT16
C4F5      332 ;
C4F5 4C 00 E8 333      jmp EOS
C4F8      334 ;
C4F8      335 ;
C4F8      336 ; This is the ASCII text that is compared to EPTEXT in
C4F8      337 ; order to determine if the slot that is being tested
C4F8      338 ; contains an EPROM card.
C4F8      339 ;
C4F8      340      .if LABEL
C4F8 C5 D0 C2 341 EPBINTXT asc "EPBINEOS"
C4FB C9 CE C5
C4FE CF D3
C500      342      .el
C500      343      asc "EPBINEOS"
C500      344      .fi
C500      345 ;
C500      346 ;
C500      347      icl "SLOT5.L"

```

LLOAD SLOT5.L,A\$4000

```

C500          1          ttl "EOS+ Source Code, SLOT5.L"
C500          2          ;
C500          3          ;
C500          4          ; SLOT5.L
C500          5          ;
C500          6          ;
0005          7  HWSLOT   let 5
0050          8  HWSLOT16 let $50
00C5          9  HWSLOT16 let $C5
C500         10          ;
0000         11  LABEL    let 0
C500         12          ;
C500         13          ;
C500         14          ; This is the generic code that is assembled specifically
C500         15          ; for each of the seven slots in which an EPROM card may
C500         16          ; reside.
C500         17          ;
C500         18          ; Interface to process an ASEOS command.
C500         19          ;
C500         20          .if LABEL
C500         21  EPASEOS  sta ASAV
C500         22          .el
C500 85 18      23          sta ASAV
C502         24          .fi
C502         25          ;
C502 18         26          clc
C503 90 03      27          bcc >0                ; always taken
C505         28          ;
C505         29          ;
C505         30          ; Insert TESTROM verification code here. When the CXRESET
C505         31          ; routine is entered, it calls TSTROMCD to test for a ROM
C505         32          ; card. TSTROMCD calls TESTROM that looks for 0x38xx18 at
C505         33          ; 0xC305. If found, C3ROMOFF is enabled.
C505         34          ;
C505 38         35          sec
C506 90 00      36          bcc *+2
C508         37          dfs !-1
C507 18         38          clc
C508         39          ;
C508         40          ;
C508 86 16      41  ^0      stx XSAV
C50A 84 17      42          sty YSAV
C50C         43          ;
C50C A5 76      44          lda CURLIN+1
C50E C9 FF      45          cmp #RUNMODE
C510 D0 03      46          bne >2
C512         47          ;
C512 A5 18      48  ^1      lda ASAV                ; recall A-reg
C514         49          ;
C514 60         50          rts
C515         51          ;
C515 A2 00      52  ^2      ldx #ZERO
C517         53          ;
C517 8E A0 02    54  ^3      stx ASPRNUM
C51A         55          ;
C51A 20 B7 00    56          jsr CHRGOT
C51D F0 21      57          beq >4
C51F         58          ;
C51F 20 BE DE    59          jsr CHKCOM
C522         60          ;

```

```

C522 20 B7 00      61      jsr CHRGOT
C525 F0 19         62      beq >4
C527              63      ;
C527 C9 2C         64      cmp #COMMA&MSBCLR
C529 F0 15         65      beq >4
C52B              66      ;
C52B 20 E3 DF      67      jsr PTRGET
C52E              68      ;
C52E AE A0 02      69      ldx ASPRNUM
C531 E0 0C         70      cpx #MAXASNUM      ; too many parameters
C533 F0 DD         71      beq <1
C535              72      ;
C535 9D D0 02      73      sta ASPRADRS,X
C538              74      ;
C538 98           75      tya
C539 9D D1 02      76      sta ASPRADRS+1,X
C53C              77      ;
C53C E8           78      inx
C53D E8           79      inx
C53E              80      ;
C53E D0 D7         81      bne <3      ; always taken
C540              82      ;
C540 4E A0 02      83      ^4      lsr ASPRNUM
C543 F0 CD         84      beq <1      ; no parameters
C545              85      ;
C545 A9 00         86      lda #EPONVAL
C547 8D D0 C0      87      sta EPSELC+HWSLOT16
C54A              88      ;
C54A              89      ;
C54A              90      ; Initialize X-reg with this slot number and enter ASEOS.
C54A              91      ;
C54A A2 05         92      ldx #HWSLOT
C54C              93      ;
C54C 4C 10 F2      94      jmp ASEOS
C54F              95      ;
C54F              96      ;
C54F              97      dfs $50-*)&NEGONE,NEGONE
C550              98      ;
C550              99      ;
C550             100      ; Exit for ASEOS and to RUN Applesoft files. If a ZipChip
C550             101      ; is present, flush cache, enable ZipChip, and return to
C550             102      ; the external caller or address in MEMJMP.
C550             103      ;
C550             104      .if LABEL
C550             105      ASEXIT  lda ZSTATUS
C550             106      .el
C550 AD 9A 02      107      lda ZSTATUS
C553             108      .fi
C553             109      ;
C553 30 14         110      bmi >2
C555             111      ;
C555 A0 00         112      ldy #ZERO
C557 A9 60         113      lda /PAGE60
C559             114      ;
C559 84 CE         115      sty GENPTR
C55B 85 CF         116      sta GENPTR+1
C55D             117      ;
C55D B1 CE         118      ^1      lda (GENPTR),Y
C55F             119      ;
C55F C8           120      iny
C560 D0 FB         121      bne <1

```

```

C562          122 ;
C562 E6 CF    123      inc GENPTR+1
C564 10 F7    124      bpl <1
C566          125 ;
C566 20 DB F4 126      jsr DOZCON
C569          127 ;
C569          128      .if HWCARD
C569          129 ^2      lda #EPOFFVAL
C569          130      .el
C569 A9 10    131 ^2      lda #QLOFFVAL
C56B          132      .fi
C56B          133 ;
C56B 8D D0 C0 134      sta EPSELC+HWSLOT16
C56E          135 ;
C56E A5 18    136      lda ASAV
C570 A6 16    137      ldx XSAV
C572 A4 17    138      ldy YSAV
C574          139 ;
C574 6C A6 02 140      jmp (MEMJMP)
C577          141 ;
C577          142 ;
C577          143      dfs $80-*)&NEGONE,NEGONE
C580          144 ;
C580          145 ;
C580          146 ; Exit for BINEOS and to RUN binary files. If a ZipChip is
C580          147 ; present, flush cache, enable ZipChip, push DOSWARM onto
C580          148 ; the stack, load the address for EPBINEOS, and return to
C580          149 ; the external caller or address in MEMJMP.
C580          150 ;
C580          151 ; Load the Y-reg/A-reg with the address of EPBINEOS in
C580          152 ; order for the current file in memory to process more
C580          153 ; DCBs if other EPROM files need to be loaded into memory.
C580          154 ;
C580          155      .if LABEL
C580          156 BINEXIT  lda ZSTATUS
C580          157      .el
C580 AD 9A 02 158      lda ZSTATUS
C583          159      .fi
C583          160 ;
C583 30 14    161      bmi >2
C585          162 ;
C585 A0 00    163      ldy #ZERO
C587 A9 60    164      lda /PAGE60
C589          165 ;
C589 84 CE    166      sty GENPTR
C58B 85 CF    167      sta GENPTR+1
C58D          168 ;
C58D B1 CE    169 ^1      lda (GENPTR),Y
C58F          170 ;
C58F C8       171      iny
C590 D0 FB    172      bne <1
C592          173 ;
C592 E6 CF    174      inc GENPTR+1
C594 10 F7    175      bpl <1
C596          176 ;
C596 20 DB F4 177      jsr DOZCON
C599          178 ;
C599 A2 FF    179 ^2      ldx #NEGONE
C59B 9A       180      txs
C59C          181 ;
C59C A9 03    182      lda /DOSWARM-1

```

```

C59E 48          183          pha
C59F           184          ;
C59F A9 CF      185          lda #DOSWARM-1
C5A1 48          186          pha
C5A2           187          ;
C5A2 A0 E0      188          ld y #EPBINEOS
C5A4 A9 C5      189          lda #HWSLOT CX
C5A6           190          ;
C5A6           191          ;
C5A6 EA         192          nop
C5A7 EA         193          nop
C5A8           194          ;
C5A8           195          ;
C5A8           196          ; Exit from EOS with the assumption that a return to EOS
C5A8           197          ; will be made by means of EPMAPEOS. The X-reg must be
C5A8           198          ; used in order to turn this EPROM card OFF.
C5A8           199          ;
C5A8           200          .if LABEL
C5A8           201          RTNEXIT:
C5A8           202          .fi
C5A8           203          ;
C5A8           204          .if HWCARD
C5A8           205          ldx #EPOFFVAL
C5A8           206          .el
C5A8 A2 10      207          ldx #QLOFFVAL
C5AA           208          .fi
C5AA           209          ;
C5AA 8E D0 C0   210          stx EPSELC+HWSLOT16
C5AD           211          ;
C5AD A2 50      212          ldx #HWSLOT16
C5AF           213          ;
C5AF 6C A6 02   214          jmp (MEMJMP)
C5B2           215          ;
C5B2           216          ;
C5B2           217          dfs $B8-*&NEGONE,NEGONE
C5B8           218          ;
C5B8           219          ;
C5B8           220          ; Special entrance in order to turn this EPROM card OFF.
C5B8           221          ;
C5B8           222          .if LABEL
C5B8           223          EPOFF:
C5B8           224          .fi
C5B8           225          ;
C5B8           226          .if HWCARD
C5B8           227          lda #EPOFFVAL
C5B8           228          .el
C5B8 A9 10      229          lda #QLOFFVAL
C5BA           230          .fi
C5BA           231          ;
C5BA 8D D0 C0   232          sta EPSELC+HWSLOT16
C5BD           233          ;
C5BD 60         234          rts
C5BE           235          ;
C5BE           236          ;
C5BE           237          dfs $C0-*&NEGONE,NEGONE
C5C0           238          ;
C5C0           239          ;
C5C0           240          ; Return from DOS CMDUSER 1 command.
C5C0           241          ;
C5C0           242          .if LABEL
C5C0           243          EPUSER1 lda #EPONVAL

```

```

C5C0          244          .el
C5C0 A9 00     245          lda #EPONVAL
C5C2          246          .fi
C5C2          247          ;
C5C2 8D D0 C0  248          sta EPSELC+HWSLOT16
C5C5          249          ;
C5C5 4C 12 E9  250          jmp USERRTN1
C5C8          251          ;
C5C8          252          ;
C5C8          253          ; Return from DOS CMDUSER 2 command.
C5C8          254          ;
C5C8          255          .if LABEL
C5C8          256 EPUSER2  lda #EPONVAL
C5C8          257          .el
C5C8 A9 00     258          lda #EPONVAL
C5CA          259          .fi
C5CA          260          ;
C5CA 8D D0 C0  261          sta EPSELC+HWSLOT16
C5CD          262          ;
C5CD 4C 10 EC  263          jmp USERRTN2
C5D0          264          ;
C5D0          265          ;
C5D0          266          ; Entrance into the EOS+ card at the EPMAPEOS location by
C5D0          267          ; using SLOTMAP from the EOS mapping function.
C5D0          268          ;
C5D0          269          .if LABEL
C5D0          270 EPMAPEOS lda #EPONVAL
C5D0          271          .el
C5D0 A9 00     272          lda #EPONVAL
C5D2          273          .fi
C5D2          274          ;
C5D2 8D D0 C0  275          sta EPSELC+HWSLOT16
C5D5          276          ;
C5D5          277          ;
C5D5          278          ; Initialize X-reg with this slot number and enter MAPEOS.
C5D5          279          ;
C5D5 A2 05     280          ldx #HWSLOT
C5D7          281          ;
C5D7 4C AD E8  282          jmp MAPEOS
C5DA          283          ;
C5DA          284          ;
C5DA          285          dfs $E0-*&NEGONE,NEGONE
C5E0          286          ;
C5E0          287          ;
C5E0          288          ; Interface entrance in order to process a BINEOS command
C5E0          289          ; that is contained within a BINEOS DCB. The Y-reg and the
C5E0          290          ; A-reg must contain the address of the eight byte DCB.
C5E0          291          ; The X-reg is initialized with the slot number of this
C5E0          292          ; EPROM card before calling BINEOS. BINEOS first calls
C5E0          293          ; DOZCOFF and MOVEEPBM before processing the BINEOS DCB.
C5E0          294          ; The X-reg must be used in order to turn this EPROM card
C5E0          295          ; ON.
C5E0          296          ;
C5E0          297          .if LABEL
C5E0          298 EPBINEOS ldx #EPONVAL
C5E0          299          .el
C5E0 A2 00     300          ldx #EPONVAL
C5E2          301          .fi
C5E2          302          ;
C5E2 8E D0 C0  303          stx EPSELC+HWSLOT16
C5E5          304          ;

```



```

C5E5      305 ;
C5E5      306 ; Initialize X-reg with this slot number and then enter
C5E5      307 ; BINEOS.
C5E5      308 ;
C5E5 A2 05 309      ldx #HWSLOT
C5E7      310 ;
C5E7 4C B0 F3 311      jmp BINEOS
C5EA      312 ;
C5EA      313 ;
C5EA      314      dfs $F0-*)&NEGONE,NEGONE
C5F0      315 ;
C5F0      316 ;
C5F0      317 ; Manual entrance into EOS for this EPROM card.  If this
C5F0      318 ; EPROM card is not the highest priority EPROM card,
C5F0      319 ; control will pass to the highest priority EPROM card
C5F0      320 ; after the EPROM card mapping function has completed.
C5F0      321 ; CTRL-N may be used in order to select the desired
C5F0      322 ; EPROM card if there are mulitple EPROM cards in an
C5F0      323 ; Apple computer.
C5F0      324 ;
C5F0      325      .if LABEL
C5F0      326 EPEOS    lda #EPONVAL
C5F0      327      .el
C5F0 A9 00 328      lda #EPONVAL
C5F2      329      .fi
C5F2      330 ;
C5F2 8D D0 C0 331      sta EPSELC+HWSLOT16
C5F5      332 ;
C5F5 4C 00 E8 333      jmp EOS
C5F8      334 ;
C5F8      335 ;
C5F8      336 ; This is the ASCII text that is compared to EPTEXT in
C5F8      337 ; order to determine if the slot that is being tested
C5F8      338 ; contains an EPROM card.
C5F8      339 ;
C5F8      340      .if LABEL
C5F8      341 EPBINTXT asc "EPBINEOS"
C5F8      342      .el
C5F8 C5 D0 C2 343      asc "EPBINEOS"
C5FB C9 CE C5
C5FE CF D3
C600      344      .fi
C600      345 ;
C600      346 ;
C600      347      icl "SLOT6.L"

```

```

LLOAD SLOT6.L,A$4000

```

```

C600          1          ttl "EOS+ Source Code, SLOT6.L"
C600          2          ;
C600          3          ;
C600          4          ; SLOT6.L
C600          5          ;
C600          6          ;
0006          7          HWSLOT    let 6
0060          8          HWSLOT16 let $60
00C6          9          HWSLOT16 let $C6
C600         10          ;
0000         11          LABEL    let 0
C600         12          ;
C600         13          ;
C600         14          ; This is the generic code that is assembled specifically
C600         15          ; for each of the seven slots in which an EPROM card may
C600         16          ; reside.
C600         17          ;
C600         18          ; Interface to process an ASEOS command.
C600         19          ;
C600         20          .if LABEL
C600         21          EPASEOS  sta ASAV
C600         22          .el
C600 85 18     23          sta ASAV
C602         24          .fi
C602         25          ;
C602 18       26          clc
C603 90 03    27          bcc >0          ; always taken
C605         28          ;
C605         29          ;
C605         30          ; Insert TESTROM verification code here. When the CXRESET
C605         31          ; routine is entered, it calls TSTROMCD to test for a ROM
C605         32          ; card. TSTROMCD calls TESTROM that looks for 0x38xx18 at
C605         33          ; 0xC305. If found, C3ROMOFF is enabled.
C605         34          ;
C605 38       35          sec
C606 90 00    36          bcc *+2
C608         37          dfs !-1
C607 18       38          clc
C608         39          ;
C608         40          ;
C608 86 16    41          ^0          stx XSAV
C60A 84 17    42          sty YSAV
C60C         43          ;
C60C A5 76    44          lda CURLIN+1
C60E C9 FF    45          cmp #RUNMODE
C610 D0 03    46          bne >2
C612         47          ;
C612 A5 18    48          ^1          lda ASAV          ; recall A-reg
C614         49          ;
C614 60       50          rts
C615         51          ;
C615 A2 00    52          ^2          ldx #ZERO
C617         53          ;
C617 8E A0 02 54          ^3          stx ASPRNUM
C61A         55          ;
C61A 20 B7 00 56          jsr CHRGOT
C61D F0 21    57          beq >4
C61F         58          ;
C61F 20 BE DE 59          jsr CHKCOM
C622         60          ;

```

```

C622 20 B7 00      61      jsr CHRGOT
C625 F0 19        62      beq >4
C627              63      ;
C627 C9 2C        64      cmp #COMMA&MSBCLR
C629 F0 15        65      beq >4
C62B              66      ;
C62B 20 E3 DF     67      jsr PTRGET
C62E              68      ;
C62E AE A0 02     69      ldx ASPRNUM
C631 E0 0C        70      cpx #MAXASNUM      ; too many parameters
C633 F0 DD        71      beq <1
C635              72      ;
C635 9D D0 02     73      sta ASPRADRS,X
C638              74      ;
C638 98           75      tya
C639 9D D1 02     76      sta ASPRADRS+1,X
C63C              77      ;
C63C E8           78      inx
C63D E8           79      inx
C63E              80      ;
C63E D0 D7        81      bne <3      ; always taken
C640              82      ;
C640 4E A0 02     83      ^4      lsr ASPRNUM
C643 F0 CD        84      beq <1      ; no parameters
C645              85      ;
C645 A9 00        86      lda #EPONVAL
C647 8D E0 C0     87      sta EPSELC+HWSLOT16
C64A              88      ;
C64A              89      ;
C64A              90      ; Initialize X-reg with this slot number and enter ASEOS.
C64A              91      ;
C64A A2 06        92      ldx #HWSLOT
C64C              93      ;
C64C 4C 10 F2     94      jmp ASEOS
C64F              95      ;
C64F              96      ;
C64F              97      dfs $50-*)&NEGONE,NEGONE
C650              98      ;
C650              99      ;
C650            100      ; Exit for ASEOS and to RUN Applesoft files. If a ZipChip
C650            101      ; is present, flush cache, enable ZipChip, and return to
C650            102      ; the external caller or address in MEMJMP.
C650            103      ;
C650            104      .if LABEL
C650            105      ASEXIT  lda ZSTATUS
C650            106      .el
C650 AD 9A 02     107      lda ZSTATUS
C653            108      .fi
C653            109      ;
C653 30 14        110      bmi >2
C655            111      ;
C655 A0 00        112      ldy #ZERO
C657 A9 60        113      lda /PAGE60
C659            114      ;
C659 84 CE        115      sty GENPTR
C65B 85 CF        116      sta GENPTR+1
C65D            117      ;
C65D B1 CE        118      ^1      lda (GENPTR),Y
C65F            119      ;
C65F C8           120      iny
C660 D0 FB        121      bne <1

```

```

C662          122 ;
C662 E6 CF    123      inc GENPTR+1
C664 10 F7    124      bpl <1
C666          125 ;
C666 20 DB F4 126      jsr DOZCON
C669          127 ;
C669          128      .if HWCARD
C669          129 ^2      lda #EPOFFVAL
C669          130      .el
C669 A9 10    131 ^2      lda #QLOFFVAL
C66B          132      .fi
C66B          133 ;
C66B 8D E0 C0 134      sta EPSELC+HWSLOT16
C66E          135 ;
C66E A5 18    136      lda ASAV
C670 A6 16    137      ldx XSAV
C672 A4 17    138      ldy YSAV
C674          139 ;
C674 6C A6 02 140      jmp (MEMJMP)
C677          141 ;
C677          142 ;
C677          143      dfs $80-*)&NEGONE,NEGONE
C680          144 ;
C680          145 ;
C680          146 ; Exit for BINEOS and to RUN binary files. If a ZipChip is
C680          147 ; present, flush cache, enable ZipChip, push DOSWARM onto
C680          148 ; the stack, load the address for EPBINEOS, and return to
C680          149 ; the external caller or address in MEMJMP.
C680          150 ;
C680          151 ; Load the Y-reg/A-reg with the address of EPBINEOS in
C680          152 ; order for the current file in memory to process more
C680          153 ; DCBs if other EPROM files need to be loaded into memory.
C680          154 ;
C680          155      .if LABEL
C680          156 BINEXIT  lda ZSTATUS
C680          157      .el
C680 AD 9A 02 158      lda ZSTATUS
C683          159      .fi
C683          160 ;
C683 30 14    161      bmi >2
C685          162 ;
C685 A0 00    163      ldy #ZERO
C687 A9 60    164      lda /PAGE60
C689          165 ;
C689 84 CE    166      sty GENPTR
C68B 85 CF    167      sta GENPTR+1
C68D          168 ;
C68D B1 CE    169 ^1      lda (GENPTR),Y
C68F          170 ;
C68F C8       171      iny
C690 D0 FB    172      bne <1
C692          173 ;
C692 E6 CF    174      inc GENPTR+1
C694 10 F7    175      bpl <1
C696          176 ;
C696 20 DB F4 177      jsr DOZCON
C699          178 ;
C699 A2 FF    179 ^2      ldx #NEGONE
C69B 9A       180      txs
C69C          181 ;
C69C A9 03    182      lda /DOSWARM-1

```

```

C69E 48          183          pha
C69F           184          ;
C69F A9 CF      185          lda #DOSWARM-1
C6A1 48          186          pha
C6A2           187          ;
C6A2 A0 E0      188          ldy #EPBINEOS
C6A4 A9 C6      189          lda #HWSLOT16
C6A6           190          ;
C6A6           191          ;
C6A6 EA         192          nop
C6A7 EA         193          nop
C6A8           194          ;
C6A8           195          ;
C6A8           196          ; Exit from EOS with the assumption that a return to EOS
C6A8           197          ; will be made by means of EPMAPEOS. The X-reg must be
C6A8           198          ; used in order to turn this EPROM card OFF.
C6A8           199          ;
C6A8           200          .if LABEL
C6A8           201          RTNEXIT:
C6A8           202          .fi
C6A8           203          ;
C6A8           204          .if HWCARD
C6A8           205          ldx #EPOFFVAL
C6A8           206          .el
C6A8 A2 10      207          ldx #QLOFFVAL
C6AA           208          .fi
C6AA           209          ;
C6AA 8E E0 C0   210          stx EPSELC+HWSLOT16
C6AD           211          ;
C6AD A2 60      212          ldx #HWSLOT16
C6AF           213          ;
C6AF 6C A6 02   214          jmp (MEMJMP)
C6B2           215          ;
C6B2           216          ;
C6B2           217          dfs $B8-*&NEGONE,NEGONE
C6B8           218          ;
C6B8           219          ;
C6B8           220          ; Special entrance in order to turn this EPROM card OFF.
C6B8           221          ;
C6B8           222          .if LABEL
C6B8           223          EPOFF:
C6B8           224          .fi
C6B8           225          ;
C6B8           226          .if HWCARD
C6B8           227          lda #EPOFFVAL
C6B8           228          .el
C6B8 A9 10      229          lda #QLOFFVAL
C6BA           230          .fi
C6BA           231          ;
C6BA 8D E0 C0   232          sta EPSELC+HWSLOT16
C6BD           233          ;
C6BD 60         234          rts
C6BE           235          ;
C6BE           236          ;
C6BE           237          dfs $C0-*&NEGONE,NEGONE
C6C0           238          ;
C6C0           239          ;
C6C0           240          ; Return from DOS CMDUSER 1 command.
C6C0           241          ;
C6C0           242          .if LABEL
C6C0           243          EPUSER1 lda #EPONVAL

```

```

C6C0          244          .el
C6C0 A9 00     245          lda #EPONVAL
C6C2          246          .fi
C6C2          247          ;
C6C2 8D E0 C0  248          sta EPSELC+HWSLOT16
C6C5          249          ;
C6C5 4C 12 E9  250          jmp USERRTN1
C6C8          251          ;
C6C8          252          ;
C6C8          253          ; Return from DOS CMDUSER 2 command.
C6C8          254          ;
C6C8          255          .if LABEL
C6C8          256 EPUSER2  lda #EPONVAL
C6C8          257          .el
C6C8 A9 00     258          lda #EPONVAL
C6CA          259          .fi
C6CA          260          ;
C6CA 8D E0 C0  261          sta EPSELC+HWSLOT16
C6CD          262          ;
C6CD 4C 10 EC  263          jmp USERRTN2
C6D0          264          ;
C6D0          265          ;
C6D0          266          ; Entrance into the EOS+ card at the EPMAPEOS location by
C6D0          267          ; using SLOTMAP from the EOS mapping function.
C6D0          268          ;
C6D0          269          .if LABEL
C6D0          270 EPMAPEOS lda #EPONVAL
C6D0          271          .el
C6D0 A9 00     272          lda #EPONVAL
C6D2          273          .fi
C6D2          274          ;
C6D2 8D E0 C0  275          sta EPSELC+HWSLOT16
C6D5          276          ;
C6D5          277          ;
C6D5          278          ; Initialize X-reg with this slot number and enter MAPEOS.
C6D5          279          ;
C6D5 A2 06     280          ldx #HWSLOT
C6D7          281          ;
C6D7 4C AD E8  282          jmp MAPEOS
C6DA          283          ;
C6DA          284          ;
C6DA          285          dfs $E0-*&NEGONE,NEGONE
C6E0          286          ;
C6E0          287          ;
C6E0          288          ; Interface entrance in order to process a BINEOS command
C6E0          289          ; that is contained within a BINEOS DCB. The Y-reg and the
C6E0          290          ; A-reg must contain the address of the eight byte DCB.
C6E0          291          ; The X-reg is initialized with the slot number of this
C6E0          292          ; EPROM card before calling BINEOS. BINEOS first calls
C6E0          293          ; DOZCOFF and MOVEEPBM before processing the BINEOS DCB.
C6E0          294          ; The X-reg must be used in order to turn this EPROM card
C6E0          295          ; ON.
C6E0          296          ;
C6E0          297          .if LABEL
C6E0          298 EPBINEOS ldx #EPONVAL
C6E0          299          .el
C6E0 A2 00     300          ldx #EPONVAL
C6E2          301          .fi
C6E2          302          ;
C6E2 8E E0 C0  303          stx EPSELC+HWSLOT16
C6E5          304          ;

```

```

C6E5      305 ;
C6E5      306 ; Initialize X-reg with this slot number and then enter
C6E5      307 ; BINEOS.
C6E5      308 ;
C6E5 A2 06 309      ldx #HWSLOT
C6E7      310 ;
C6E7 4C B0 F3 311      jmp BINEOS
C6EA      312 ;
C6EA      313 ;
C6EA      314      dfs $F0-*&NEGONE,NEGONE
C6F0      315 ;
C6F0      316 ;
C6F0      317 ; Manual entrance into EOS for this EPROM card.  If this
C6F0      318 ; EPROM card is not the highest priority EPROM card,
C6F0      319 ; control will pass to the highest priority EPROM card
C6F0      320 ; after the EPROM card mapping function has completed.
C6F0      321 ; CTRL-N may be used in order to select the desired
C6F0      322 ; EPROM card if there are mulitple EPROM cards in an
C6F0      323 ; Apple computer.
C6F0      324 ;
C6F0      325      .if LABEL
C6F0      326 EPEOS    lda #EPONVAL
C6F0      327      .el
C6F0 A9 00 328      lda #EPONVAL
C6F2      329      .fi
C6F2      330 ;
C6F2 8D E0 C0 331      sta EPSELC+HWSLOT16
C6F5      332 ;
C6F5 4C 00 E8 333      jmp EOS
C6F8      334 ;
C6F8      335 ;
C6F8      336 ; This is the ASCII text that is compared to EPTEXT in
C6F8      337 ; order to determine if the slot that is being tested
C6F8      338 ; contains an EPROM card.
C6F8      339 ;
C6F8      340      .if LABEL
C6F8      341 EPBINTXT asc "EPBINEOS"
C6F8      342      .el
C6F8 C5 D0 C2 343      asc "EPBINEOS"
C6FB C9 CE C5
C6FE CF D3
C700      344      .fi
C700      345 ;
C700      346 ;
C700      347      icl "SLOT7.L"

```

LLOAD SLOT7.L,A\$4000

```

C700          1          ttl "EOS+ Source Code, SLOT7.L"
C700          2          ;
C700          3          ;
C700          4          ; SLOT7.L
C700          5          ;
C700          6          ;
0007          7          HWSLOT    let 7
0070          8          HWSLOT16 let $70
00C7          9          HWSLOT16 let $C7
C700         10          ;
0000         11          LABEL     let 0
C700         12          ;
C700         13          ;
C700         14          ; This is the generic code that is assembled specifically
C700         15          ; for each of the seven slots in which an EPROM card may
C700         16          ; reside.
C700         17          ;
C700         18          ; Interface to process an ASEOS command.
C700         19          ;
C700         20          .if LABEL
C700         21          EPASEOS   sta ASAV
C700         22          .el
C700 85 18     23          sta ASAV
C702         24          .fi
C702         25          ;
C702 18       26          clc
C703 90 03    27          bcc >0          ; always taken
C705         28          ;
C705         29          ;
C705         30          ; Insert TESTROM verification code here. When the CXRESET
C705         31          ; routine is entered, it calls TSTROMCD to test for a ROM
C705         32          ; card. TSTROMCD calls TESTROM that looks for 0x38xx18 at
C705         33          ; 0xC305. If found, C3ROMOFF is enabled.
C705         34          ;
C705 38       35          sec
C706 90 00    36          bcc *+2
C708         37          dfs !-1
C707 18       38          clc
C708         39          ;
C708         40          ;
C708 86 16    41          ^0          stx XSAV
C70A 84 17    42          sty YSAV
C70C         43          ;
C70C A5 76    44          lda CURLIN+1
C70E C9 FF    45          cmp #RUNMODE
C710 D0 03    46          bne >2
C712         47          ;
C712 A5 18    48          ^1          lda ASAV          ; recall A-reg
C714         49          ;
C714 60       50          rts
C715         51          ;
C715 A2 00    52          ^2          ldx #ZERO
C717         53          ;
C717 8E A0 02 54          ^3          stx ASPRNUM
C71A         55          ;
C71A 20 B7 00 56          jsr CHRGOT
C71D F0 21    57          beq >4
C71F         58          ;
C71F 20 BE DE 59          jsr CHKCOM
C722         60          ;

```



```

C722 20 B7 00      61      jsr CHRGOT
C725 F0 19        62      beq >4
C727              63      ;
C727 C9 2C        64      cmp #COMMA&MSBCLR
C729 F0 15        65      beq >4
C72B              66      ;
C72B 20 E3 DF     67      jsr PTRGET
C72E              68      ;
C72E AE A0 02     69      ldx ASPRNUM
C731 E0 0C        70      cpx #MAXASNUM      ; too many parameters
C733 F0 DD        71      beq <1
C735              72      ;
C735 9D D0 02     73      sta ASPRADRS,X
C738              74      ;
C738 98           75      tya
C739 9D D1 02     76      sta ASPRADRS+1,X
C73C              77      ;
C73C E8           78      inx
C73D E8           79      inx
C73E              80      ;
C73E D0 D7        81      bne <3      ; always taken
C740              82      ;
C740 4E A0 02     83      ^4      lsr ASPRNUM
C743 F0 CD        84      beq <1      ; no parameters
C745              85      ;
C745 A9 00        86      lda #EPONVAL
C747 8D F0 C0     87      sta EPSELC+HWSLOT16
C74A              88      ;
C74A              89      ;
C74A              90      ; Initialize X-reg with this slot number and enter ASEOS.
C74A              91      ;
C74A A2 07        92      ldx #HWSLOT
C74C              93      ;
C74C 4C 10 F2     94      jmp ASEOS
C74F              95      ;
C74F              96      ;
C74F              97      dfs $50-*)&NEGONE,NEGONE
C750              98      ;
C750              99      ;
C750            100      ; Exit for ASEOS and to RUN Applesoft files. If a ZipChip
C750            101      ; is present, flush cache, enable ZipChip, and return to
C750            102      ; the external caller or address in MEMJMP.
C750            103      ;
C750            104      .if LABEL
C750            105      ASEXIT  lda ZSTATUS
C750            106      .el
C750 AD 9A 02     107      lda ZSTATUS
C753            108      .fi
C753            109      ;
C753 30 14        110      bmi >2
C755            111      ;
C755 A0 00        112      ldy #ZERO
C757 A9 60        113      lda /PAGE60
C759            114      ;
C759 84 CE        115      sty GENPTR
C75B 85 CF        116      sta GENPTR+1
C75D            117      ;
C75D B1 CE        118      ^1      lda (GENPTR),Y
C75F            119      ;
C75F C8           120      iny
C760 D0 FB        121      bne <1

```

```

C762          122 ;
C762 E6 CF    123      inc GENPTR+1
C764 10 F7    124      bpl <1
C766          125 ;
C766 20 DB F4 126      jsr DOZCON
C769          127 ;
C769          128      .if HWCARD
C769          129 ^2      lda #EPOFFVAL
C769          130      .el
C769 A9 10     131 ^2      lda #QLOFFVAL
C76B          132      .fi
C76B          133 ;
C76B 8D F0 C0 134      sta EPSELC+HWSLOT16
C76E          135 ;
C76E A5 18     136      lda ASAV
C770 A6 16     137      ldx XSAV
C772 A4 17     138      ldy YSAV
C774          139 ;
C774 6C A6 02  140      jmp (MEMJMP)
C777          141 ;
C777          142 ;
C777          143      dfs $80-*)&NEGONE,NEGONE
C780          144 ;
C780          145 ;
C780          146 ; Exit for BINEOS and to RUN binary files. If a ZipChip is
C780          147 ; present, flush cache, enable ZipChip, push DOSWARM onto
C780          148 ; the stack, load the address for EPBINEOS, and return to
C780          149 ; the external caller or address in MEMJMP.
C780          150 ;
C780          151 ; Load the Y-reg/A-reg with the address of EPBINEOS in
C780          152 ; order for the current file in memory to process more
C780          153 ; DCBs if other EPROM files need to be loaded into memory.
C780          154 ;
C780          155      .if LABEL
C780          156 BINEXIT  lda ZSTATUS
C780          157      .el
C780 AD 9A 02  158      lda ZSTATUS
C783          159      .fi
C783          160 ;
C783 30 14     161      bmi >2
C785          162 ;
C785 A0 00     163      ldy #ZERO
C787 A9 60     164      lda /PAGE60
C789          165 ;
C789 84 CE     166      sty GENPTR
C78B 85 CF     167      sta GENPTR+1
C78D          168 ;
C78D B1 CE     169 ^1      lda (GENPTR),Y
C78F          170 ;
C78F C8        171      iny
C790 D0 FB     172      bne <1
C792          173 ;
C792 E6 CF     174      inc GENPTR+1
C794 10 F7     175      bpl <1
C796          176 ;
C796 20 DB F4  177      jsr DOZCON
C799          178 ;
C799 A2 FF     179 ^2      ldx #NEGONE
C79B 9A        180      txs
C79C          181 ;
C79C A9 03     182      lda /DOSWARM-1

```

```

C79E 48          183          pha
C79F          184          ;
C79F A9 CF      185          lda #DOSWARM-1
C7A1 48          186          pha
C7A2          187          ;
C7A2 A0 E0      188          ldy #EPBINEOS
C7A4 A9 C7      189          lda #HWSLOTCTX
C7A6          190          ;
C7A6          191          ;
C7A6 EA        192          nop
C7A7 EA        193          nop
C7A8          194          ;
C7A8          195          ;
C7A8          196          ; Exit from EOS with the assumption that a return to EOS
C7A8          197          ; will be made by means of EPMAPEOS. The X-reg must be
C7A8          198          ; used in order to turn this EPROM card OFF.
C7A8          199          ;
C7A8          200          .if LABEL
C7A8          201          RTNEXIT:
C7A8          202          .fi
C7A8          203          ;
C7A8          204          .if HWCARD
C7A8          205          ldx #EPOFFVAL
C7A8          206          .el
C7A8 A2 10      207          ldx #QLOFFVAL
C7AA          208          .fi
C7AA          209          ;
C7AA 8E F0 C0   210          stx EPSELC+HWSLOT16
C7AD          211          ;
C7AD A2 70      212          ldx #HWSLOT16
C7AF          213          ;
C7AF 6C A6 02   214          jmp (MEMJMP)
C7B2          215          ;
C7B2          216          ;
C7B2          217          dfs $B8-*&NEGONE,NEGONE
C7B8          218          ;
C7B8          219          ;
C7B8          220          ; Special entrance in order to turn this EPROM card OFF.
C7B8          221          ;
C7B8          222          .if LABEL
C7B8          223          EPOFF:
C7B8          224          .fi
C7B8          225          ;
C7B8          226          .if HWCARD
C7B8          227          lda #EPOFFVAL
C7B8          228          .el
C7B8 A9 10      229          lda #QLOFFVAL
C7BA          230          .fi
C7BA          231          ;
C7BA 8D F0 C0   232          sta EPSELC+HWSLOT16
C7BD          233          ;
C7BD 60         234          rts
C7BE          235          ;
C7BE          236          ;
C7BE          237          dfs $C0-*&NEGONE,NEGONE
C7C0          238          ;
C7C0          239          ;
C7C0          240          ; Return from DOS CMDUSER 1 command.
C7C0          241          ;
C7C0          242          .if LABEL
C7C0          243          EPUSER1 lda #EPONVAL

```

```

C7C0          244          .el
C7C0 A9 00     245          lda #EPONVAL
C7C2          246          .fi
C7C2          247          ;
C7C2 8D F0 C0  248          sta EPSELC+HWSLOT16
C7C5          249          ;
C7C5 4C 12 E9  250          jmp USERRTN1
C7C8          251          ;
C7C8          252          ;
C7C8          253          ; Return from DOS CMDUSER 2 command.
C7C8          254          ;
C7C8          255          .if LABEL
C7C8          256 EPUSER2  lda #EPONVAL
C7C8          257          .el
C7C8 A9 00     258          lda #EPONVAL
C7CA          259          .fi
C7CA          260          ;
C7CA 8D F0 C0  261          sta EPSELC+HWSLOT16
C7CD          262          ;
C7CD 4C 10 EC  263          jmp USERRTN2
C7D0          264          ;
C7D0          265          ;
C7D0          266          ; Entrance into the EOS+ card at the EPMAPEOS location by
C7D0          267          ; using SLOTMAP from the EOS mapping function.
C7D0          268          ;
C7D0          269          .if LABEL
C7D0          270 EPMAPEOS  lda #EPONVAL
C7D0          271          .el
C7D0 A9 00     272          lda #EPONVAL
C7D2          273          .fi
C7D2          274          ;
C7D2 8D F0 C0  275          sta EPSELC+HWSLOT16
C7D5          276          ;
C7D5          277          ;
C7D5          278          ; Initialize X-reg with this slot number and enter MAPEOS.
C7D5          279          ;
C7D5 A2 07     280          ldx #HWSLOT
C7D7          281          ;
C7D7 4C AD E8  282          jmp MAPEOS
C7DA          283          ;
C7DA          284          ;
C7DA          285          dfs $E0-*&NEGONE,NEGONE
C7E0          286          ;
C7E0          287          ;
C7E0          288          ; Interface entrance in order to process a BINEOS command
C7E0          289          ; that is contained within a BINEOS DCB. The Y-reg and the
C7E0          290          ; A-reg must contain the address of the eight byte DCB.
C7E0          291          ; The X-reg is initialized with the slot number of this
C7E0          292          ; EPROM card before calling BINEOS. BINEOS first calls
C7E0          293          ; DOZCOFF and MOVEEPBM before processing the BINEOS DCB.
C7E0          294          ; The X-reg must be used in order to turn this EPROM card
C7E0          295          ; ON.
C7E0          296          ;
C7E0          297          .if LABEL
C7E0          298 EPBINEOS  ldx #EPONVAL
C7E0          299          .el
C7E0 A2 00     300          ldx #EPONVAL
C7E2          301          .fi
C7E2          302          ;
C7E2 8E F0 C0  303          stx EPSELC+HWSLOT16
C7E5          304          ;

```

```

C7E5      305 ;
C7E5      306 ; Initialize X-reg with this slot number and then enter
C7E5      307 ; BINEOS.
C7E5      308 ;
C7E5 A2 07 309      ldx #HWSLOT
C7E7      310 ;
C7E7 4C B0 F3 311      jmp BINEOS
C7EA      312 ;
C7EA      313 ;
C7EA      314      dfs $F0-*)&NEGONE,NEGONE
C7F0      315 ;
C7F0      316 ;
C7F0      317 ; Manual entrance into EOS for this EPROM card.  If this
C7F0      318 ; EPROM card is not the highest priority EPROM card,
C7F0      319 ; control will pass to the highest priority EPROM card
C7F0      320 ; after the EPROM card mapping function has completed.
C7F0      321 ; CTRL-N may be used in order to select the desired
C7F0      322 ; EPROM card if there are multiple EPROM cards in an
C7F0      323 ; Apple computer.
C7F0      324 ;
C7F0      325      .if LABEL
C7F0      326 EPEOS    lda #EPONVAL
C7F0      327      .el
C7F0 A9 00 328      lda #EPONVAL
C7F2      329      .fi
C7F2      330 ;
C7F2 8D F0 C0 331      sta EPSELC+HWSLOT16
C7F5      332 ;
C7F5 4C 00 E8 333      jmp EOS
C7F8      334 ;
C7F8      335 ;
C7F8      336 ; This is the ASCII text that is compared to EPTEXT in
C7F8      337 ; order to determine if the slot that is being tested
C7F8      338 ; contains an EPROM card.
C7F8      339 ;
C7F8      340      .if LABEL
C7F8      341 EPBINTXT asc "EPBINEOS"
C7F8      342      .el
C7F8 C5 D0 C2 343      asc "EPBINEOS"
C7FB C9 CE C5
C7FE CF D3
C800      344      .fi
C800      345 ;
C800      346 ;
C800      347      icl "MENU.L"

```

```

LLOAD MENU.L,A$4000

```

```

C800          1          ttl "EOS+ Source Code, MENU.L"
C800          2          ;
C800          3          ;
C800          4          ; MENU.L
C800          5          ;
C800          6          ;
C800          7          .if DEBUG
C800          8          .el
C800          9          phs PAGEE8
E800         10          .fi
E800         11          ;
E800         12          ;
E800         13          ; This is the entrance for an EPROM card after a RESET has
E800         14          ; been pressed.
E800         15          ;
E800 D8       16 EOS      cld
E801         17          ;
E801         18          ;
E801         19          ; Initialize stack pointer and general state of computer.
E801         20          ;
E801 A2 FF    21          ldx #NEGONE
E803 9A       22          txs
E804         23          ;
E804 8E FB 04  24          stx XMODE
E807         25          ;
E807 8D 00 C0  26          sta STR80OFF
E80A 8D 02 C0  27          sta RAMRDOFF
E80D 8D 04 C0  28          sta RAMWROFF
E810 8D 08 C0  29          sta AUXZPOFF
E813 8D 0C C0  30          sta VID80OFF
E816 8D 0E C0  31          sta ALTCHOFF
E819         32          ;
E819         33          ;
E819         34          ; Determine if this is an Apple ][ or an Apple //e. Set
E819         35          ; APPLTYPE to 0x00 for Apple ][ and 0xFF for an Apple //e.
E819         36          ; The Apple ][ will fail either test because it does not
E819         37          ; have this particular hardware. Run test for 32 times
E819         38          ; in order to avoid random false positives.
E819         39          ;
E819 A2 00     40          ldx #ZERO
E81B A0 20     41          ldy #TESTCNT
E81D         42          ;
E81D 8D 07 C0  43 ^1      sta CXROMON          ; enable internal CX ROM
E820         44          ;
E820 AD 15 C0  45          lda RDCXROM          ; read which ROM is enabled
E823 10 0C     46          bpl >2          ; BPL is wrong, so Apple ][
E825         47          ;
E825 8D 06 C0  48          sta CXROMOFF        ; enable slot ROM
E828         49          ;
E828 AD 15 C0  50          lda RDCXROM
E82B 30 04     51          bmi >2          ; BMI is wrong, Apple ][
E82D         52          ;
E82D 88       53          dey
E82E D0 ED     54          bne <1
E830         55          ;
E830 CA       56          dex
E831         57          ;
E831 8E 93 02  58 ^2      stx APPLTYPE
E834 8D 06 C0  59          sta CXROMOFF
E837         60          ;

```

```

E837      61 ;
E837      62 ; Set initial video and Language Card.
E837      63 ;
E837 AD 56 C0      64      lda HIRESOFF
E83A AD 54 C0      65      lda PAGE1ON
E83D AD 51 C0      66      lda TEXTON
E840      67 ;
E840 A9 00      68      lda #ZERO
E842 85 22      69      sta WNDTOP
E844 85 20      70      sta WNDLFT
E846      71 ;
E846 A9 18      72      lda #24
E848 85 23      73      sta WNDBTM
E84A      74 ;
E84A A9 28      75      lda #40
E84C 85 21      76      sta WNDWDTH
E84E      77 ;
E84E AD 82 C0      78      lda ROM2WP
E851      79 ;
E851 AD FF CF      80      lda CLRROM
E854      81 ;
E854      82 ;
E854      83 ; If there is a ZipChip then disable it and turn it off.
E854      84 ; The annunciators are set by these two routines.
E854      85 ;
E854 20 9F F4      86      jsr DOZCOFF
E857 20 F6 F4      87      jsr DOZCRSET
E85A      88 ;
E85A AD 9A 02      89      lda ZSTATUS
E85D 49 10      90      eor #ZCSTAT      ; toggle ZipChip OFF
E85F 8D 9A 02      91      sta ZSTATUS
E862      92 ;
E862      93 ;
E862      94 ; Initialize the page-zero pointers in order to discover
E862      95 ; in which slot this EPROM card resides.
E862      96 ;
E862 A0 00      97      ldy #ZERO
E864      98 ;
E864 8C 94 02      99      sty EPNMBR
E867 8C 95 02     100      sty EPBANK
E86A      101 ;
E86A A9 C1      102      lda /PAGEC1
E86C      103 ;
E86C 84 2A      104      sty SRCPTR
E86E 85 2B      105      sta SRCPTR+1
E870      106 ;
E870 A9 E1      107      lda /PAGEE1
E872      108 ;
E872 84 2E      109      sty DSTPTR
E874 85 2F      110      sta DSTPTR+1
E876      111 ;
E876 8D 0B C0     112      sta C3ROMON      ; enable slot 3 ROM
E879      113 ;
E879 A2 01      114      ldx #1      ; begin with Slot 1
E87B      115 ;
E87B      116 ;
E87B      117 ; Test for an EPROM card by comparing the EPTEXT in the
E87B      118 ; Slot Page and in the EPROM page.
E87B      119 ;
E87B A0 F8      120 ^3      ldy #EPBINTXT
E87D      121 ;

```

```

E87D B9 3E FD    122  ^4      lda EPTEXT-EPBINTXT&NEGONE,Y
E880              123  ;
E880 D1 2A      124      cmp (SRCPTR),Y          ; compare slot page
E882 D0 09      125      bne >5
E884              126  ;
E884 D1 2E      127      cmp (DSTPTR),Y          ; compare EPROM page
E886 D0 05      128      bne >5
E888              129  ;
E888 C8          130      iny
E889 D0 F2      131      bne <4
E88B              132  ;
E88B F0 0D      133      beq >6                ; always taken
E88D              134  ;
E88D              135  ;
E88D              136  ; Go to the next slot to test.
E88D              137  ;
E88D 2C FF CF   138  ^5      bit CLRROM
E890              139  ;
E890 E6 2B      140      inc SRCPTR+1          ; next slot page
E892 E6 2F      141      inc DSTPTR+1          ; next EPROM page
E894              142  ;
E894 E8          143      inx
E895              144  ;
E895 E0 08      145      cpx #8
E897 D0 E2      146      bne <3
E899              147  ;
E899              148  ;
E899              149  ; A horrible situation -- the hardware must be broken.
E899              150  ;
E899              151      .if DEBUG
E899              152      ldx #4
E899              153      nop
E899              154      nop
E899              155      .el
E899 00          156      brk
E89A              157      .fi
E89A              158  ;
E89A              159  ;
E89A              160  ; Found this EPROM card. Build the SLOTMAP for the
E89A              161  ; computer.
E89A              162  ;
E89A 2C FF CF   163  ^6      bit CLRROM
E89D              164  ;
E89D 8E 90 02   165      stx PRISLOT
E8A0              166  ;
E8A0 20 E1 FA   167      jsr BUILDMAP
E8A3              168  ;
E8A3 AE 90 02   169      ldx PRISLOT
E8A6 E0 03      170      cpx #SLOT3
E8A8 F0 03      171      beq MAPEOS
E8AA              172  ;
E8AA 8D 0A C0   173      sta C3ROMOFF          ; disable slot 3 ROM
E8AD              174  ;
E8AD              175  ;
E8AD              176  ; This is the general entrance for an EPROM card.
E8AD              177  ; Install the EP interface management routines.
E8AD              178  ;
E8AD 8E 91 02   179  MAPEOS  stx EPSLOT
E8B0              180  ;
E8B0 20 5C FB   181      jsr MOVEEPBM
E8B3              182  ;

```



```

E8B3      183 ;
E8B3      184 ; If the ESC key is pressed then exit to the monitor.
E8B3      185 ;
E8B3 AD 00 C0 186      lda KEY
E8B6 C9 9B 187      cmp #ESCAPE
E8B8 D0 07 188      bne >1
E8BA      189 ;
E8BA A0 65 190      ldy #MONITOR
E8BC A9 FF 191      lda /MONITOR
E8BE      192 ;
E8BE 4C 1E ED 193      jmp EXITBIN
E8C1      194 ;
E8C1      195 ;
E8C1      196 ; Initialize the screen.
E8C1      197 ;
E8C1 2C 10 C0 198      ^1      bit CLRKEY
E8C4      199 ;
E8C4 20 C2 F8 200      jsr PRINT
E8C7 51 201      byt NORMCMD
E8C8 52 202      byt INITCMD
E8C9 53 203      byt VIDCMD
E8CA 54 204      byt KBDCMD
E8CB 55 205      byt HOMECMD
E8CC 50 206      byt RTNCMD
E8CD      207 ;
E8CD A9 8D 208      lda #RETURN
E8CF 8D 00 02 209      sta INPUT
E8D2      210 ;
E8D2      211 ;
E8D2      212 ; Initialize the SDV values and ring the bell.
E8D2      213 ;
E8D2 A9 07 214      lda #7
E8D4 85 EB 215      sta MSLOT
E8D6      216 ;
E8D6 A9 01 217      lda #1
E8D8 85 EC 218      sta DRIVE
E8DA      219 ;
E8DA A9 00 220      lda #ZERO
E8DC 85 ED 221      sta VOLUME
E8DE      222 ;
E8DE 20 2C F7 223      jsr EOSBELL
E8E1      224 ;
E8E1      225 ;
E8E1      226 ; Check private slot variables for power up bytes. Load
E8E1      227 ; DOS if in power up state.
E8E1      228 ;
E8E1      229      .if DEBUG
E8E1      230      jmp MAIN
E8E1      231      .fi
E8E1      232 ;
E8E1 18 233      clc
E8E2      234 ;
E8E2 A9 78 235      lda #PWRUP3
E8E4 6D 90 02 236      adc PRISLOT
E8E7 85 2A 237      sta SRCPTR
E8E9      238 ;
E8E9 A9 07 239      lda /PWRUP3
E8EB 85 2B 240      sta SRCPTR+1
E8ED      241 ;
E8ED A2 03 242      ldx #SYNC.L-1
E8EF A0 00 243      ldy #ZERO

```

```

E8F1      244 ;
E8F1 B1 2A 245 ^2      lda (SRCPTR),Y
E8F3 DD 70 FE 246      cmp SYNCBYTES,X
E8F6 D0 07 247      bne >3
E8F8      248 ;
E8F8 C6 2B 249      dec SRCPTR+1
E8FA      250 ;
E8FA CA 251      dex
E8FB 10 F4 252      bpl <2
E8FD      253 ;
E8FD 30 30 254      bmi MAIN ; always taken
E8FF      255 ;
E8FF      256 ;
E8FF      257 ; If this is an Apple ][, load DOS 4.5.05L into memory.
E8FF      258 ; If this is an Apple //e, load DOS 4.5.06H into memory.
E8FF      259 ;
E8FF 2C 93 02 260 ^3      bit APPLTYPE
E902 30 06 261      bmi >4
E904      262 ;
E904 20 4C F9 263      jsr LOADOSL
E907      264 ;
E907 4C 0D E9 265      jmp >5
E90A      266 ;
E90A 20 67 F9 267 ^4      jsr LOADOSH
E90D      268 ;
E90D      269 ;
E90D      270 ; Initialize DOS. EPUSER1 will direct program flow to
E90D      271 ; USERRTN1 before MAIN.
E90D      272 ;
E90D A0 C0 273 ^5      ldy #EPUSER1
E90F      274 ;
E90F 4C F8 EB 275      jmp BHNDLR2
E912      276 ;
E912      277 ;
E912      278 ; Install power up bytes into private slot variables.
E912      279 ;
E912 20 B2 F8 280 USERRTN1 jsr CLRUSER
E915      281 ;
E915 18 282      clc
E916      283 ;
E916 A9 78 284      lda #PWRUP3
E918 6D 90 02 285      adc PRISLOT
E91B 85 2A 286      sta SRCPTR
E91D      287 ;
E91D A9 07 288      lda /PWRUP3
E91F 85 2B 289      sta SRCPTR+1
E921      290 ;
E921 A2 03 291      ldx #SYNC.L-1
E923 A0 00 292      ldy #ZERO
E925      293 ;
E925 BD 70 FE 294 ^6      lda SYNCBYTES,X
E928 91 2A 295      sta (SRCPTR),Y
E92A      296 ;
E92A C6 2B 297      dec SRCPTR+1
E92C      298 ;
E92C CA 299      dex
E92D 10 F6 300      bpl <6
E92F      301 ;
E92F      302 ;
E92F      303 ; EOS main menu display.
E92F      304 ;

```

```

E92F 20 1E F9      305  MAIN      jsr DOSHOOK
E932                306  ;
E932 20 C2 F8      307          jsr PRINT
E935 55            308          byt HOMECMD
E936 8D            309          byt RETURN
E937 02            310          hex 02
E938 C5 CF D3      311          asc "EOS+ Main Menu Selection for Slot "
E93B AB A0 CD
E93E E1 E9 EE
E941 A0 CD E5
E944 EE F5 A0
E947 D3 E5 EC
E94A E5 E3 F4
E94D E9 EF EE
E950 A0 E6 EF
E953 F2 A0 D3
E956 EC EF F4
E959 A0
E95A 50            312          byt RTNCMD
E95B                313  ;
E95B AD 91 02      314          lda EPSLOT
E95E 20 2C F9      315          jsr DOPRHEX
E961                316  ;
E961 20 C2 F8      317          jsr PRINT
E964 8D 8D          318          byt RETURN,RETURN
E966 C1 A0 C4      319          asc "A DOS 4.5.05L"
E969 CF D3 A0
E96C B4 AE B5
E96F AE B0 B5
E972 CC
E973 13            320          hex 13
E974 CE A0 C3      321          asc "N Copy ROM->RAM"
E977 EF F0 F9
E97A A0 D2 CF
E97D CD AD BE
E980 D2 C1 CD
E983 8D            322          byt RETURN
E984 C2 A0 C4      323          asc "B DOS 4.5.06H"
E987 CF D3 A0
E98A B4 AE B5
E98D AE B0 B6
E990 C8
E991 13            324          hex 13
E992 CF A0 D2      325          asc "O Run HELLO on SDV"
E995 F5 EE A0
E998 C8 C5 CC
E99B CC CF A0
E99E EF EE A0
E9A1 D3 C4 D6
E9A4 8D            326          byt RETURN
E9A5 C3 A0 C3      327          asc "C Coldstart DOS"
E9A8 EF EC E4
E9AB F3 F4 E1
E9AE F2 F4 A0
E9B1 C4 CF D3
E9B4 13            328          hex 13
E9B5 D0 A0 C3      329          asc "P CATALOG this SDV"
E9B8 C1 D4 C1
E9BB CC CF C7
E9BE A0 F4 E8
E9C1 E9 F3 A0

```

```
E9C4 D3 C4 D6
E9C7 8D          330      byt RETURN
E9C8 C4 A0 D7    331      asc "D Warmstart DOS"
E9CB E1 F2 ED
E9CE F3 F4 E1
E9D1 F2 F4 A0
E9D4 C4 CF D3
E9D7 13          332      hex 13
E9D8 D1 A0 C2    333      asc "Q BigMac"
E9DB E9 E7 CD
E9DE E1 E3
E9E0 8D          334      byt RETURN
E9E1 C5 A0 C2    335      asc "E Boot the Slot S"
E9E4 EF EF F4
E9E7 A0 F4 E8
E9EA E5 A0 D3
E9ED EC EF F4
E9F0 A0 D3
E9F2 13          336      hex 13
E9F3 D2 A0 D3    337      asc "R Scan Disk"
E9F6 E3 E1 EE
E9F9 A0 C4 E9
E9FC F3 EB
E9FE 8D          338      byt RETURN
E9FF C6 A0 C8    339      asc "F Hook the Slot S"
EA02 EF EF EB
EA05 A0 F4 E8
EA08 E5 A0 D3
EA0B EC EF F4
EA0E A0 D3
EA10 13          340      hex 13
EA11 D3 A0 C1    341      asc "S Applesoft List"
EA14 F0 F0 EC
EA17 E5 F3 EF
EA1A E6 F4 A0
EA1D CC E9 F3
EA20 F4
EA21 8D          342      byt RETURN
EA22 C7 A0 D5    343      asc "G Unhook a Slot S"
EA25 EE E8 EF
EA28 EF EB A0
EA2B E1 A0 D3
EA2E EC EF F4
EA31 A0 D3
EA33 13          344      hex 13
EA34 D4 A0 C2    345      asc "T Binary Install"
EA37 E9 EE E1
EA3A F2 F9 A0
EA3D C9 EE F3
EA40 F4 E1 EC
EA43 EC
EA44 8D          346      byt RETURN
EA45 C8 A0 D3    347      asc "H Sourceror"
EA48 EF F5 F2
EA4B E3 E5 F2
EA4E EF F2
EA50 13          348      hex 13
EA51 D5 A0 D6    349      asc "U VTOC Manager"
EA54 D4 CF C3
EA57 A0 CD E1
EA5A EE E1 E7
```

```

EA5D E5 F2
EA5F 8D          350      byt RETURN
EA60 C9 A0 C5    351      asc "I EPROM Burner"
EA63 D0 D2 CF
EA66 CD A0 C2
EA69 F5 F2 EE
EA6C E5 F2
EA6E 13          352      hex 13
EA6F D6 A0 D6    353      asc "V Volume Manager"
EA72 EF EC F5
EA75 ED E5 A0
EA78 CD E1 EE
EA7B E1 E7 E5
EA7E F2
EA7F 8D          354      byt RETURN
EA80 CA A0 C6    355      asc "J FID"
EA83 C9 C4
EA85 13          356      hex 13
EA86 D7 A0 D6    357      asc "W Volume Duplicate"
EA89 EF EC F5
EA8C ED E5 A0
EA8F C4 F5 F0
EA92 EC E9 E3
EA95 E1 F4 E5
EA98 8D          358      byt RETURN
EA99 CB A0 C1    359      asc "K ADT2"
EA9C C4 D4 B2
EA9F 13          360      hex 13
EAA0 D8 A0 C4    361      asc "X Disk Window"
EAA3 E9 F3 EB
EAA6 A0 D7 E9
EAA9 EE E4 EF
EAAC F7
EAAD 8D          362      byt RETURN
EAAE CC A0 CC    363      asc "L Lisa80"
EAB1 E9 F3 E1
EAB4 B8 B0
EAB6 13          364      hex 13
EAB7 D9 A0 D2    365      asc "Y Real Time Clock"
EABA E5 E1 EC
EABD A0 D4 E9
EAC0 ED E5 A0
EAC3 C3 EC EF
EAC6 E3 EB
EAC8 8D          366      byt RETURN
EAC9 CD A0 D2    367      asc "M RamDisk Config"
EACC E1 ED C4
EACF E9 F3 EB
EAD2 A0 C3 EF
EAD5 EE E6 E9
EAD8 E7
EAD9 13          368      hex 13
EADA DA A0 DA    369      asc "Z ZipChip Config"
EADD E9 F0 C3
EAE0 E8 E9 F0
EAE3 A0 C3 EF
EAE6 EE E6 E9
EAE9 E7
EAEA 8D 8D        370      byt RETURN,RETURN
EAEC A0 A0 D3     371      asc "  S=  D=  V="
EAEF BD A0 A0

```

```

EAF2 C4 BD A0
EAF5 A0 A0 D6
EAF8 BD
EAF9 13          372      hex 13
EAFA          373      ;
EAFA          374      .if HWCARD
EAFA          375      asc "00-0F EPROM Catalog"
EAFA          376      .el
EAFA B0 B0 AD    377      asc "00-07 EPROM Catalog"
EAFD B0 B7 A0
EB00 C5 D0 D2
EB03 CF CD A0
EB06 C3 E1 F4
EB09 E1 EC EF
EB0C E7
EB0D          378      .fi
EB0D          379      ;
EB0D 8D 8D      380      byt RETURN,RETURN
EB0F DE C3 A0   381      asc "^C Configure SDV"
EB12 C3 EF EE
EB15 E6 E9 E7
EB18 F5 F2 E5
EB1B A0 D3 C4
EB1E D6
EB1F 13          382      hex 13
EB20 D2 D4 CE   383      asc "RTN Toggle ZipChip"
EB23 A0 D4 EF
EB26 E7 E7 EC
EB29 E5 A0 DA
EB2C E9 F0 C3
EB2F E8 E9 F0
EB32 8D          384      byt RETURN
EB33 DE D3 A0   385      asc "^S Skip EPROM Card"
EB36 D3 EB E9
EB39 F0 A0 C5
EB3C D0 D2 CF
EB3F CD A0 C3
EB42 E1 F2 E4
EB45 50          386      byt RTNCMD
EB46          387      ;
EB46          388      ;
EB46 20 62 F8    389      MAIN2      jsr PRTSDV
EB49          390      ;
EB49          391      ;
EB49 20 C2 F8    392      MAIN3      jsr PRINT
EB4C 17 74      393      hex 1774
EB4E AD BE A0   394      asc "-> 0"
EB51 CF
EB52 1B          395      hex 1B
EB53 50          396      byt RTNCMD
EB54          397      ;
EB54 AD 9A 02    398      lda ZSTATUS
EB57 30 0D      399      bmi >1
EB59          400      ;
EB59 D0 0B      401      bne >1
EB5B          402      ;
EB5B 20 C2 F8    403      jsr PRINT
EB5E EE A0 BC    404      asc "n <- "
EB61 AD A0
EB63 50          405      byt RTNCMD
EB64          406      ;

```

```

EB64 90 09      407      bcc SELC                ; always taken
EB66           408      ;
EB66 20 C2 F8   409      ^1      jsr PRINT
EB69 E6 E6 A0   410      asc "ff <-"
EB6C BC AD
EB6E 50         411      byt RTNCMD
EB6F           412      ;
EB6F           413      ;
EB6F           414      ; EOS main menu selection.
EB6F           415      ;
EB6F 20 C2 F8   416      SELC      jsr PRINT
EB72 00 76      417      hex 0076
EB74 C5 EE F4   418      asc "Enter Selection:  "
EB77 E5 F2 A0
EB7A D3 E5 EC
EB7D E5 E3 F4
EB80 E9 EF EE
EB83 BA A0 A0
EB86 57         419      byt EOLCMD
EB87 50         420      byt RTNCMD
EB88           421      ;
EB88 20 3C F7   422      jsr RDKEY
EB8B           423      ;
EB8B C9 83      424      cmp #CTRLC
EB8D D0 06      425      bne >2
EB8F           426      ;
EB8F 20 FA F7   427      jsr EDITS DV
EB92           428      ;
EB92 4C 46 EB   429      jmp MAIN2
EB95           430      ;
EB95           431      ;
EB95           432      ; Confirm that SLOTMAP contains a valid value.  If not, do
EB95           433      ; not process SLOTMAP.
EB95           434      ;
EB95 C9 93      435      ^2      cmp #CTRLS
EB97 D0 0D      436      bne >3
EB99           437      ;
EB99 AD 92 02   438      lda SLOTMAP
EB9C C9 02      439      cmp #2                ; value if card in slot 1
EB9E 90 37      440      bcc SELCERR
EBA0           441      ;
EBA0 20 9A F8   442      jsr NEXTMAP
EBA3           443      ;
EBA3 4C 21 ED   444      jmp EXITRTN
EBA6           445      ;
EBA6 C9 8D      446      ^3      cmp #RETURN
EBA8 D0 0C      447      bne >4
EBAA           448      ;
EBAA AD 9A 02   449      lda ZSTATUS
EBAD 30 28      450      bmi SELCERR
EBAF           451      ;
EBAF 49 10      452      eor #ZCSTAT
EBB1 8D 9A 02   453      sta ZSTATUS
EBB4           454      ;
EBB4 10 93      455      bpl MAIN3                ; always taken
EBB6           456      ;
EBB6 C9 B0      457      ^4      cmp #"0"
EBB8 D0 15      458      bne >5
EBBA           459      ;
EBBA 20 35 F9   460      jsr DOCOUT
EBBD 20 3C F7   461      jsr RDKEY

```

```

EBC0          462 ;
EBC0 20 59 F7 463      jsr GETHEX
EBC3 B0 12    464      bcs SELCERR
EBC5          465 ;
EBC5 20 35 F9 466      jsr DOCOUT
EBC8          467 ;
EBC8 8A       468      txa
EBC9 20 63 ED 469      jsr EOSCAT
EBCC          470 ;
EBCC 4C 2F E9 471      jmp MAIN
EBCF          472 ;
EBCF C9 C1    473 ^5    cmp #"A"
EBD1 90 04    474      bcc SELCERR
EBD3          475 ;
EBD3 C9 DB    476      cmp #"Z"+1
EBD5 90 05    477      bcc >6
EBD7          478 ;
EBD7 20 2C F7 479 SELCERR jsr EOSBELL
EBDA F0 93    480      beq SELC          ; always taken
EBDC          481 ;
EBDC 20 35 F9 482 ^6    jsr DOCOUT
EBDF          483 ;
EBDF 29 1F    484      and #MENUMASK
EBE1          485 ;
EBE1 0A       486      asl
EBE2 A8       487      tay
EBE3          488 ;
EBE3 B9 1C FD 489      lda CMDTBL-1,Y
EBE6 48       490      pha
EBE7          491 ;
EBE7 B9 1B FD 492      lda CMDTBL-2,Y
EBEA 48       493      pha
EBEB          494 ;
EBEB 60       495      rts
EBEC          496 ;
EBEC          497 ;
EBEC          498 ; EOS main menu selection handlers.
EBEC          499 ;
EBEC          500 ; EPUSER2 will direct program flow to USERRTN2 before
EBEC          501 ; DHNDLR.
EBEC          502 ;
EBEC 20 4C F9 503 AHNDLR  jsr LOADOSL
EBEF          504 ;
EBEF A0 C8    505      ldy #EPUSER2
EBF1 D0 05    506      bne BHNDLR2          ; always taken
EBF3          507 ;
EBF3          508 ;
EBF3 20 67 F9 509 BHNDLR  jsr LOADOSH
EBF6          510 ;
EBF6 A0 C8    511      ldy #EPUSER2
EBF8          512 ;
EBF8 AD 91 02 513 BHNDLR2  lda EPSLOT
EBFB 09 C0    514      ora /PAGEC0
EBFD          515 ;
EBFD 20 B4 F8 516      jsr SETUSER
EC00          517 ;
EC00 AC F8 BF 518      ldy INITDOS
EC03 AD F9 BF 519      lda INITDOS+1
EC06          520 ;
EC06 4C 21 ED 521      jmp EXITRTN
EC09          522 ;

```



```

EC09          523 ;
EC09 A0 D3    524 CHNDLR    ldy #DOSCOLD
EC0B A9 03    525          lda /DOSCOLD
EC0D          526 ;
EC0D 4C 1E ED 527          jmp EXITBIN
EC10          528 ;
EC10          529 ;
EC10 20 B2 F8 530 USERRTN2 jsr CLRUSER
EC13          531 ;
EC13 20 33 F9 532 DHNDLR    jsr DOCROUT
EC16          533 ;
EC16 A0 D0    534          ldy #DOSWARM
EC18 A9 03    535          lda /DOSWARM
EC1A          536 ;
EC1A 4C 1E ED 537          jmp EXITBIN
EC1D          538 ;
EC1D          539 ;
EC1D A5 EB    540 EHNDLR    lda MSLOT
EC1F 85 18    541          sta ASAV
EC21          542 ;
EC21 A0 95    543          ldy #OUTPORT
EC23 A9 FE    544          lda /OUTPORT
EC25          545 ;
EC25 4C 1B ED 546          jmp EXITAS
EC28          547 ;
EC28          548 ;
EC28 A0 10    549 FHNDLR    ldy #HOOKSLT
EC2A          550 ;
EC2A 2C 00 00 551          bit *-*
EC2D          552          dfs !-2
EC2B          553 ;
EC2B          554 ;
EC2B A0 18    555 GHNDLR    ldy #UHOOKSLT
EC2D          556 ;
EC2D A5 EB    557          lda MSLOT
EC2F 09 C0    558          ora /PAGEC0
EC31          559 ;
EC31 20 43 F9 560          jsr JSRMEM
EC34 20 43 F7 561          jsr GETKEY
EC37          562 ;
EC37 4C 2F E9 563          jmp MAIN
EC3A          564 ;
EC3A          565 ;
EC3A A2 16    566 HHNDLR    ldx #LSDLEN
EC3C          567 ;
EC3C A0 64    568          ldy #LSDCB
EC3E A9 FC    569          lda /LSDCB
EC40          570 ;
EC40 4C 04 ED 571          jmp DOEOSDCB
EC43          572 ;
EC43          573 ;
EC43 A2 14    574 IHNDLR    ldx #EBDLLEN
EC45          575 ;
EC45 A0 7A    576          ldy #EBDCB
EC47 A9 FC    577          lda /EBDCB
EC49          578 ;
EC49 4C 04 ED 579          jmp DOEOSDCB
EC4C          580 ;
EC4C          581 ;
EC4C A2 0B    582 JHNDLR    ldx #FDDLLEN
EC4E          583 ;

```

```

EC4E A0 21      584      ldy #FDDCB
EC50 A9 FC      585      lda /FDDCB
EC52           586      ;
EC52 4C 04 ED    587      jmp DOEOSDCB
EC55           588      ;
EC55           589      ;
EC55 A2 0C      590      KHNDLR    ldx #ADDLEN
EC57           591      ;
EC57 A0 2C      592      ldy #ADDCB
EC59 A9 FC      593      lda /ADDCB
EC5B           594      ;
EC5B 4C 04 ED    595      jmp DOEOSDCB
EC5E           596      ;
EC5E           597      ;
EC5E A2 12      598      LHNDLR    ldx #LLDLEN
EC60           599      ;
EC60 A0 F9      600      ldy #LLDCB
EC62 A9 FB      601      lda /LLDCB
EC64           602      ;
EC64 4C 04 ED    603      jmp DOEOSDCB
EC67           604      ;
EC67           605      ;
EC67 A2 16      606      MHNDLR    ldx #RDDLEN
EC69           607      ;
EC69 A0 0B      608      ldy #RDDCB
EC6B A9 FC      609      lda /RDDCB
EC6D           610      ;
EC6D 4C 04 ED    611      jmp DOEOSDCB
EC70           612      ;
EC70           613      ;
EC70 20 AC F9    614      NHNDLR    jsr COPYROM
EC73           615      ;
EC73 4C D7 EB    616      jmp SELCERR
EC76           617      ;
EC76           618      ;
EC76 A2 01      619      OHNDLR    ldx #1                ; HELOTEXT
EC78           620      ;
EC78 2C 00 00    621      bit *-*
EC7B           622      dfs !-2
EC79           623      ;
EC79           624      ;
EC79 A2 00      625      PHNDLR    ldx #ZERO                ; CATTEXT
EC7B           626      ;
EC7B 20 4B ED    627      jsr DOEXEC
EC7E           628      ;
EC7E A2 02      629      ldx #2                ; SLTTEXT
EC80 20 4B ED    630      jsr DOEXEC
EC83           631      ;
EC83 A5 EB      632      lda MSLOT
EC85 20 2C F9    633      jsr DOPRHEX
EC88           634      ;
EC88 A2 03      635      ldx #3                ; DRVTEXT
EC8A 20 4B ED    636      jsr DOEXEC
EC8D           637      ;
EC8D A5 EC      638      lda DRIVE
EC8F 20 28 F9    639      jsr DOPRBYTE
EC92           640      ;
EC92 A2 04      641      ldx #4                ; VOLTEXT
EC94 20 4B ED    642      jsr DOEXEC
EC97           643      ;
EC97 A5 ED      644      lda VOLUME

```

```

EC99 20 28 F9      645      jsr DOPRBYTE
EC9C              646      ;
EC9C 20 33 F9      647      jsr DOCROUT
EC9F 20 43 F7      648      jsr GETKEY
ECA2              649      ;
ECA2 4C 2F E9      650      jmp MAIN
ECA5              651      ;
ECA5              652      ;
ECA5 A2 13         653      QHNDLR    ldx #BMDLEN
ECA7              654      ;
ECA7 A0 FE         655      ldy #BMDCB
ECA9 A9 FC         656      lda /BMDCB
ECAB              657      ;
ECAB 4C 04 ED      658      jmp DOEOSDCB
ECAE              659      ;
ECAE              660      ;
ECAE A2 0C         661      RHNDLR    ldx #SDDLEN
ECB0              662      ;
ECB0 A0 11         663      ldy #SDDCB
ECB2 A9 FD         664      lda /SDDCB
ECB4              665      ;
ECB4 4C 04 ED      666      jmp DOEOSDCB
ECB7              667      ;
ECB7              668      ;
ECB7 A2 1B         669      SHNDLR    ldx #AFDLEN
ECB9              670      ;
ECB9 A0 49         671      ldy #AFDCB
ECBB A9 FC         672      lda /AFDCB
ECBD              673      ;
ECBD 4C 04 ED      674      jmp DOEOSDCB
ECC0              675      ;
ECC0              676      ;
ECC0 A2 1B         677      THNDLR    ldx #BFDLEN
ECC2              678      ;
ECC2 A0 E3         679      ldy #BFDLCB
ECC4 A9 FC         680      lda /BFDLCB
ECC6              681      ;
ECC6 4C 04 ED      682      jmp DOEOSDCB
ECC9              683      ;
ECC9              684      ;
ECC9 A2 14         685      UHNDLR    ldx #VTDLEN
ECCB              686      ;
ECCB A0 8E         687      ldy #VTDCB
ECCD A9 FC         688      lda /VTDCB
ECCF              689      ;
ECCF 4C 04 ED      690      jmp DOEOSDCB
ECD2              691      ;
ECD2              692      ;
ECD2 A2 16         693      VHNDLR    ldx #VODLEN
ECD4              694      ;
ECD4 A0 A2         695      ldy #VODCB
ECD6 A9 FC         696      lda /VODCB
ECD8              697      ;
ECD8 4C 04 ED      698      jmp DOEOSDCB
ECDB              699      ;
ECDB              700      ;
ECDB A2 18         701      WHNDLR    ldx #VDDLEN
ECDD              702      ;
ECDD A0 B8         703      ldy #VDDCB
ECDF A9 FC         704      lda /VDDCB
ECE1              705      ;

```

```

ECE1 4C 04 ED    706          jmp DOEOSDCB
ECE4              707      ;
ECE4              708      ;
ECE4 A2 13      709  XHNDLR    ldx #DWDLEN
ECE6              710      ;
ECE6 A0 D0      711          ldy #DWDCB
ECE8 A9 FC      712          lda /DWDCB
ECEA              713      ;
ECEA 4C 04 ED    714          jmp DOEOSDCB
ECED              715      ;
ECED              716      ;
ECED A2 11      717  YHNDLR    ldx #SCDLEN
ECF7              718      ;
ECF7 A0 38      719          ldy #SCDCB
ECF1 A9 FC      720          lda /SCDCB
ECF3              721      ;
ECF3 4C 04 ED    722          jmp DOEOSDCB
ECF6              723      ;
ECF6              724      ;
ECF6 AD 9A 02    725  ZHNDLR    lda ZSTATUS
ECF9              726      ;
ECF9              727          .if DEBUG
ECF9              728          nop
ECF9              729          nop
ECF9              730          .el
ECF9 30 06      731          bmi >1
ECFB              732          .fi
ECFB              733      ;
ECFB 20 97 F5    734          jsr ZCONFIG
ECFE              735      ;
ECFE 4C 2F E9    736          jmp MAIN
ED01              737      ;
ED01 4C D7 EB    738      ^1      jmp SELCERR
ED04              739      ;
ED04              740      ;
ED04              741      ; Save the address of the selected DCB and copy the DCB
ED04              742      ; contents to the DCBBUFR. The X-reg contains the number
ED04              743      ; of bytes to copy. Call BINEOS2 to process the DCB.
ED04              744      ;
ED04              745      ; A RUN DCB will exit through EXITBIN and a LOAD DCB will
ED04              746      ; return to MAIN.
ED04              747      ;
ED04 84 EE      748  DOEOSDCB  sty CMDPTR
ED06 85 EF      749          sta CMDPTR+1
ED08              750      ;
ED08 A0 00      751          ldy #ZERO
ED0A              752      ;
ED0A B1 EE      753      ^1      lda (CMDPTR),Y
ED0C 99 D0 02    754          sta DCBBUFR,Y
ED0F              755      ;
ED0F C8          756          iny
ED10              757      ;
ED10 CA          758          dex
ED11 D0 F7      759          bne <1
ED13              760      ;
ED13 A9 01      761          lda #INTERNAL
ED15 20 CC F3    762          jsr BINEOS2
ED18              763      ;
ED18 4C 2F E9    764          jmp MAIN
ED1B              765      ;
ED1B              766      ;

```

```

ED1B      767 ; EXIT for all ASEOS utilization other than RUNMODE and
ED1B      768 ; OUTPORT.
ED1B      769 ;
ED1B A2 50  770 EXITAS   ldx #ASEXIT
ED1D      771 ;
ED1D 2C 00 00 772         bit *-*
ED20      773         dfs !-2
ED1E      774 ;
ED1E      775 ;
ED1E      776 ; EXIT for all external BINEOS utilization and RUNMODE.
ED1E      777 ;
ED1E A2 80  778 EXITBIN   ldx #BINEXIT
ED20      779 ;
ED20 2C 00 00 780         bit *-*
ED23      781         dfs !-2
ED21      782 ;
ED21      783 ;
ED21      784 ; EXIT for the routines that return to EOS using USERRTN
ED21      785 ; or the mapping function.
ED21      786 ;
ED21 A2 A8  787 EXITRTN   ldx #RTNEXIT
ED23      788 ;
ED23 8C A6 02 789         sty MEMJMP
ED26 8D A7 02 790         sta MEMJMP+1
ED29      791 ;
ED29 AD 91 02 792         lda EPSLOT
ED2C 09 C0    793         ora /PAGEC0
ED2E      794 ;
ED2E 8E A8 02 795         stx SLOTJMP
ED31 8D A9 02 796         sta SLOTJMP+1
ED34      797 ;
ED34 2C FF CF 798         bit CLRROM
ED37      799 ;
ED37 6C A8 02 800         jmp (SLOTJMP)
ED3A      801 ;
ED3A      802 ;
ED3A A9 FF    803 EXECTEXT  lda #RUNMODE
ED3C 85 76    804         sta CURLIN+1
ED3E      805 ;
ED3E 20 92 FA 806         jsr SELCBANK
ED41      807 ;
ED41 A4 2A    808         ldy SRCPTR
ED43 A5 2B    809         lda SRCPTR+1
ED45      810 ;
ED45 20 59 ED 811         jsr DOEXEC2
ED48      812 ;
ED48 4C 13 EC 813         jmp DHNDLR
ED4B      814 ;
ED4B      815 ;
ED4B BC 0B FE 816 DOEXEC   ldy TEXTTBLL,X
ED4E BD 10 FE 817         lda TEXTTBLH,X
ED51      818 ;
ED51 A2 00    819         ldx #ZERO
ED53 8E 94 02 820         stx EPNMBR
ED56 8E 95 02 821         stx EPBANK
ED59      822 ;
ED59      823 ;
ED59 84 FA    824 DOEXEC2   sty EXECPTR
ED5B 85 FB    825         sta EXECPTR+1
ED5D      826 ;
ED5D 20 1E F9 827         jsr DOSHOOK

```

```
ED60          828  ;
ED60 4C 65 01  829      jmp EPEXEC
ED63          830  ;
ED63          831  ;
ED63          832      icl "CAT.L"

LLOAD CAT.L,A$4000
```

```

ED63          1          ttl "EOS+ Source Code, CAT.L"
ED63          2          ;
ED63          3          ;
ED63          4          ; CAT.L
ED63          5          ;
ED63          6          ;
ED63 8D 94 02   7  EOSCAT   sta  EPNMBR
ED66          8          ;
ED66 AD 94 02   9  EOSCAT2  lda  EPNMBR
ED69          10         ;
ED69          11         .if  HWCARD
ED69          12         and  #EPMASK
ED69          13         .el
ED69 29 07      14         and  #QLMASK
ED6B          15         .fi
ED6B          16         ;
ED6B 8D 94 02   17         sta  EPNMBR
ED6E          18         ;
ED6E 20 13 EE   19         jsr  CATHDR
ED71          20         ;
ED71 20 CC F9   21         jsr  INITCAT
ED74 90 75      22         bcc  >3
ED76          23         ;
ED76 20 C2 F8   24         jsr  PRINT
ED79 00 64      25         hex  0064
ED7B 58         26         byt  EOPCMD
ED7C 00 69      27         hex  0069
ED7E 59         28         byt  CNTRCMD
ED7F D4 E8 E5   29         asc  "There is no EPROM"
ED82 F2 E5 A0
ED85 E9 F3 A0
ED88 EE EF A0
ED8B C5 D0 D2
ED8E CF CD
ED90 8D 8D      30         byt  RETURN,RETURN
ED92 59         31         byt  CNTRCMD
ED93 EF F2 A0   32         asc  "or EPROM Catalog"
ED96 C5 D0 D2
ED99 CF CD A0
ED9C C3 E1 F4
ED9F E1 EC EF
EDA2 E7
EDA3 8D 8D      33         byt  RETURN,RETURN
EDA5 59         34         byt  CNTRCMD
EDA6 E9 EE A0   35         asc  "in this Socket."
EDA9 F4 E8 E9
EDAC F3 A0 D3
EDAF EF E3 EB
EDB2 E5 F4 AE
EDB5 00 76      36         hex  0076
EDB7 59         37         byt  CNTRCMD
EDB8 D0 F2 E5   38         asc  "Press any key for next EPROM"
EDBB F3 F3 A0
EDBE E1 EE F9
EDC1 A0 EB E5
EDC4 F9 A0 E6
EDC7 EF F2 A0
EDCA EE E5 F8
EDCD F4 A0 C5
EDD0 D0 D2 CF

```

```

EDD3 CD
EDD4 50          39          byt RTNCMD
EDD5          40          ;
EDD5 20 43 F7    41          jsr GETKEY
EDD8          42          ;
EDD8 C9 D1       43          cmp #"Q"
EDDA F0 29       44          beq >5
EDDC          45          ;
EDDC C9 8B       46          cmp #UARROW
EDDE D0 06       47          bne >2
EDE0          48          ;
EDE0 CE 94 02    49          dec EPNMBR
EDE3          50          ;
EDE3 4C 66 ED    51          ^1      jmp EOSCAT2
EDE6          52          ;
EDE6 EE 94 02    53          ^2      inc EPNMBR
EDE9 D0 F8       54          bne <1          ; always taken
EDEB          55          ;
EDEB 20 AE EE    56          ^3      jsr CHKCAT
EDEE          57          ;
EDEE A9 00       58          lda #ZERO
EDF0 8D 9D 02    59          sta NUMSELC
EDF3          60          ;
EDF3 20 57 EE    61          ^4      jsr CATFTR
EDF6 20 D8 EE    62          jsr SHOWCAT
EDF9          63          ;
EDF9 A9 FF       64          lda #NEGONE
EDFB 8D D1 02    65          sta FIRSTIME
EDFE          66          ;
EDFE 20 34 EF    67          jsr SELCFILE
EE01 90 F0       68          bcc <4
EE03          69          ;
EE03 D0 DE       70          bne <1
EE05          71          ;
EE05 20 C2 F8    72          ^5      jsr PRINT
EE08 52          73          byt INITCMD
EE09 50          74          byt RTNCMD
EE0A          75          ;
EE0A A9 00       76          lda #ZERO
EE0C 8D 94 02    77          sta EPNMBR
EE0F 8D 95 02    78          sta EPBANK
EE12          79          ;
EE12 60          80          rts
EE13          81          ;
EE13          82          ;
EE13 20 C2 F8    83          CATHDR jsr PRINT
EE16 52          84          byt INITCMD
EE17 55          85          byt HOMECMD
EE18 07          86          hex 07
EE19 C5 D0 D2    87          asc "EPROM Card - EPROM Catalog"
EE1C CF CD A0
EE1F C3 E1 F2
EE22 E4 A0 AD
EE25 A0 C5 D0
EE28 D2 CF CD
EE2B A0 C3 E1
EE2E F4 E1 EC
EE31 EF E7
EE33 8D          88          byt RETURN
EE34 09          89          hex 09
EE35 D3 EC EF    90          asc "Slot "

```



```

EE38 F4 A0
EE3A 50          91      byt RTNCMD
EE3B          92      ;
EE3B AD 91 02    93      lda EPSLOT
EE3E 20 2C F9    94      jsr DOPRHEX
EE41          95      ;
EE41 20 C2 F8    96      jsr PRINT
EE44 17          97      hex 17
EE45 C5 D0 D2    98      asc "EPROM "
EE48 CF CD A0
EE4B 50          99      byt RTNCMD
EE4C          100     ;
EE4C AD 94 02    101     lda EPNMBR
EE4F 20 2C F9    102     jsr DOPRHEX
EE52          103     ;
EE52 A9 04       104     lda #4
EE54 85 22       105     sta WNDTOP
EE56          106     ;
EE56 60          107     rts
EE57          108     ;
EE57          109     ;
EE57 20 C2 F8    110     CATFTR jsr PRINT
EE5A 52          111     byt INITCMD
EE5B 00 75       112     hex 0075
EE5D 58          113     byt EOPCMD
EE5E 05          114     hex 05
EE5F D2 D4 CE    115     asc "RTN - File Info"
EE62 A0 AD A0
EE65 C6 E9 EC
EE68 E5 A0 C9
EE6B EE E6 EF
EE6E 1D          116     hex 1D
EE6F A8 CC A9    117     asc "(L)oad"
EE72 EF E1 E4
EE75 8D          118     byt RETURN
EE76 05          119     hex 05
EE77 D3 D0 C3    120     asc "SPC - Next EPROM"
EE7A A0 AD A0
EE7D CE E5 F8
EE80 F4 A0 C5
EE83 D0 D2 CF
EE86 CD
EE87 1D          121     hex 1D
EE88 A8 D2 A9    122     asc "(R)un"
EE8B F5 EE
EE8D 8D          123     byt RETURN
EE8E 05          124     hex 05
EE8F          125     ;
EE8F          126     .if HWCARD
EE8F          127     asc "0-F - Select EPROM"
EE8F          128     .el
EE8F B0 AD B7    129     asc "0-7 - Select EPROM"
EE92 A0 AD A0
EE95 D3 E5 EC
EE98 E5 E3 F4
EE9B A0 C5 D0
EE9E D2 CF CD
EEA1          130     .fi
EEA1          131     ;
EEA1 1D          132     hex 1D
EEA2 A8 D1 A9    133     asc "(Q)uit"

```

```

EEA5 F5 E9 F4
EEA8 50          134          byt RTNCMD
EEA9          135          ;
EEA9 A9 15      136          lda #21
EEAB 85 23      137          sta WNDBTM
EEAD          138          ;
EEAD 60         139          rts
EEAE          140          ;
EEAE          141          ;
EEAE A9 00      142  CHKCAT   lda #ZERO
EEB0 8D D3 02   143          sta NUMNTRYS
EEB3 8D D4 02   144          sta LSTOPNTY
EEB6 8D D5 02   145          sta NTRYSTRT
EEB9          146          ;
EEB9 20 E5 F9   147  ^1      jsr GETENTRY
EEBC          148          ;
EEBC AD B1 02   149          lda FILETYPE
EEBF F0 05      150          beq >2
EEC1          151          ;
EEC1 EE D3 02   152          inc NUMNTRYS
EEC4 D0 F3      153          bne <1
EEC6          154          ;
EEC6 AD D3 02   155  ^2      lda NUMNTRYS
EEC9 C9 09      156          cmp #9
EECB 90 07      157          bcc >3
EECD          158          ;
EECD E9 08      159          sbc #8
EECF 8D D4 02   160          sta LSTOPNTY
EED2          161          ;
EED2 A9 08      162          lda #8
EED4          163          ;
EED4 8D D6 02   164  ^3      sta NTRYEND
EED7          165          ;
EED7 60         166          rts
EED8          167          ;
EED8          168          ;
EED8 20 C2 F8   169  SHOWCAT jsr PRINT
EEDB 00 64      170          hex 0064
EEDD 58         171          byt EOPCMD
EEDE 50         172          byt RTNCMD
EEDF          173          ;
EEDF A9 00      174          lda #ZERO
EEE1 8D D0 02   175          sta NUMSCRN
EEE4          176          ;
EEE4 AD D5 02   177          lda NTRYSTRT
EEE7 20 0B F0   178          jsr GETFILE
EEEE          179          ;
EEEE 20 E5 F9   180  ^1      jsr GETENTRY
EEED          181          ;
EEED A9 05      182          lda #INDENT
EEEF 85 24      183          sta CH
EEF1          184          ;
EEF1 A0 07      185          ldY #7
EEF3          186          ;
EEF3 AD B1 02   187          lda FILETYPE
EEF6          188          ;
EEF6 0A         189  ^2      asl
EEF7 B0 03      190          bcs >3
EEF9          191          ;
EEF9 88         192          dey
EEFA D0 FA      193          bne <2

```

```

EEFC      194 ;
EEFC B9 88 FD 195 ^3      lda PARMTYPE,Y
EEFF 20 35 F9 196      jsr DOCOUT
EF02      197 ;
EF02 20 C2 F8 198      jsr PRINT
EF05 A0 B0 F8 199      asc " 0x"
EF08 50      200      byt RTNCMD
EF09      201 ;
EF09 AD B1 02 202      lda FILETYPE
EF0C 20 28 F9 203      jsr DOPRBYTE
EF0F      204 ;
EF0F 20 30 F9 205      jsr DOSPACE
EF12 20 30 F9 206      jsr DOSPACE
EF15      207 ;
EF15 A0 00      208      ldy #ZERO
EF17      209 ;
EF17 B9 B8 02 210 ^4      lda FILENAME,Y
EF1A 20 35 F9 211      jsr DOCOUT
EF1D      212 ;
EF1D C8      213      iny
EF1E      214 ;
EF1E C0 18      215      cpy #NAMESIZE
EF20 D0 F5      216      bne <4
EF22      217 ;
EF22 20 33 F9 218      jsr DOCROUT
EF25 20 33 F9 219      jsr DOCROUT
EF28      220 ;
EF28 EE D0 02 221      inc NUMSCRN
EF2B      222 ;
EF2B AD D0 02 223      lda NUMSCRN
EF2E CD D6 02 224      cmp NTRYEND
EF31 D0 B7      225      bne <1
EF33      226 ;
EF33 60      227      rts
EF34      228 ;
EF34      229 ;
EF34 2C D1 02 230 SELCFILE bit FIRSTIME
EF37 30 07      231      bmi >1
EF39      232 ;
EF39 A9 0C      233      lda #INDENT+7
EF3B 85 24      234      sta CH
EF3D      235 ;
EF3D 20 30 F9 236      jsr DOSPACE
EF40      237 ;
EF40 A9 00      238 ^1      lda #ZERO
EF42 8D D1 02 239      sta FIRSTIME
EF45      240 ;
EF45 AD 9D 02 241      lda NUMSELC
EF48 0A      242      asl
EF49 69 04      243      adc #4
EF4B 85 25      244      sta CV
EF4D      245 ;
EF4D 20 1A F9 246      jsr DOVTAB
EF50      247 ;
EF50 A9 0C      248      lda #INDENT+7
EF52 85 24      249      sta CH
EF54      250 ;
EF54 A9 BE      251      lda #">"
EF56 20 35 F9 252      jsr DOCOUT
EF59      253 ;
EF59 20 43 F7 254 SELCFIL2 jsr GETKEY

```

```

EF5C          255 ;
EF5C C9 88    256      cmp #LARROW
EF5E F0 04    257      beq >3
EF60          258 ;
EF60 C9 8B    259      cmp #UARROW
EF62 D0 12    260      bne >4
EF64          261 ;
EF64 CE 9D 02 262 ^3    dec NUMSELC
EF67 10 CB    263      bpl SELCFILE
EF69          264 ;
EF69 EE 9D 02 265      inc NUMSELC
EF6C          266 ;
EF6C AD D5 02 267      lda NTRYSTRT
EF6F F0 C3    268      beq SELCFILE
EF71          269 ;
EF71 CE D5 02 270      dec NTRYSTRT
EF74 10 50    271      bpl >7 ; always taken
EF76          272 ;
EF76 C9 95    273 ^4    cmp #RARROW
EF78 F0 04    274      beq >5
EF7A          275 ;
EF7A C9 8A    276      cmp #DARROW
EF7C D0 1B    277      bne >6
EF7E          278 ;
EF7E EE 9D 02 279 ^5    inc NUMSELC
EF81          280 ;
EF81 AD 9D 02 281      lda NUMSELC
EF84 CD D6 02 282      cmp NTRYEND
EF87 D0 AB    283      bne SELCFILE
EF89          284 ;
EF89 CE 9D 02 285      dec NUMSELC
EF8C          286 ;
EF8C AD D5 02 287      lda NTRYSTRT
EF8F CD D4 02 288      cmp LSTOPNTY
EF92 F0 A0    289      beq SELCFILE
EF94          290 ;
EF94 EE D5 02 291      inc NTRYSTRT
EF97 D0 2D    292      bne >7 ; always taken
EF99          293 ;
EF99 C9 8D    294 ^6    cmp #RETURN
EF9B D0 2D    295      bne CKCATCMD
EF9D          296 ;
EF9D 20 04 F0 297      jsr GETFILE2
EFA0 20 1E F0 298      jsr SHOWFILE
EFA3          299 ;
EFA3 20 C2 F8 300      jsr PRINT
EFA6 00 76    301      hex 0076
EFA8 59       302      byt CNTRCMD
EFA9 D0 F2 E5 303      asc "Press any key to Continue"
EFAC F3 F3 A0
EFAF E1 EE F9
EFB2 A0 EB E5
EFB5 F9 A0 F4
EFB8 EF A0 C3
EFBB EF EE F4
EFBE E9 EE F5
EFC1 E5
EFC2 50       304      byt RTNCMD
EFC3          305 ;
EFC3 20 43 F7 306      jsr GETKEY
EFC6          307 ;

```

```

EFC6 18          308 ^7      clc
EFC7          309 ;
EFC7 60          310      rts
EFC8          311 ;
EFC8 38          312 ^8      sec
EFC9          313 ;
EFC9 60          314      rts
EFCA          315 ;
EFCA          316 ;
EFCA C9 D1      317 CKCATCMD  cmp #"Q"
EFCC F0 FA      318      beq <8
EFCE          319 ;
EFCE 20 59 F7   320      jsr GETHEX
EFD1 B0 07      321      bcs >1
EFD3          322 ;
EFD3 8E 94 02   323      stx EPNMBR
EFD6          324 ;
EFD6 A9 01      325      lda #1
EFD8 D0 EE      326      bne <8                ; always taken
EFDA          327 ;
EFDA C9 A0      328 ^1      cmp #SPACE
EFDC D0 05      329      bne >2
EFDE          330 ;
EFDE EE 94 02   331      inc EPNMBR
EFE1 D0 E5      332      bne <8                ; always taken
EFE3          333 ;
EFE3 C9 CC      334 ^2      cmp #"L"
EFE5 D0 04      335      bne >3
EFE7          336 ;
EFE7 A9 00      337      lda #ZERO
EFE9 F0 06      338      beq >4                ; always taken
EFEB          339 ;
EFEB C9 D2      340 ^3      cmp #"R"
EFED D0 0D      341      bne >5
EFEF          342 ;
EFEF A9 FF      343      lda #RUNMODE
EFF1          344 ;
EFF1 8D 9F 02   345 ^4      sta RUNFLAG
EFF4          346 ;
EFF4 20 04 F0   347      jsr GETFILE2
EFF7          348 ;
EFF7 2C B1 02   349      bit FILETYPE
EFFA 50 03      350      bvc >6
EFFC          351 ;
EFFC 4C 59 EF   352 ^5      jmp SELCFIL2
EFFF          353 ;
EFFF 20 E9 F0   354 ^6      jsr RUNLOAD
F002          355 ;
F002 18          356      clc
F003          357 ;
F003 60          358      rts
F004          359 ;
F004          360 ;
F004          361 ; Read the selected file's parameters from the Catalog.
F004          362 ;
F004 38          363 GETFILE2 sec
F005          364 ;
F005 AD 9D 02   365      lda NUMSELC
F008 6D D5 02   366      adc NTRYSTRT
F00B          367 ;
F00B          368 ;

```

```

F00B 48          369  GETFILE pha
F00C          370  ;
F00C 20 CC F9    371          jsr INITCAT
F00F          372  ;
F00F 68          373          pla
F010 F0 0B       374          beq >2
F012          375  ;
F012 8D D2 02    376          sta FILECNT
F015          377  ;
F015 20 E5 F9    378 ^1      jsr GETENTRY
F018          379  ;
F018 CE D2 02    380          dec FILECNT
F01B D0 F8       381          bne <1
F01D          382  ;
F01D 60          383 ^2      rts
F01E          384  ;
F01E          385  ;
F01E A9 18       386 SHOWFILE lda #24
F020 85 23       387          sta WNDBTM
F022          388  ;
F022 20 C2 F8    389          jsr PRINT
F025 00 64       390          hex 0064
F027 58          391          byt EOPCMD
F028 8D          392          byt RETURN
F029 C6 E9 EC    393          asc "File Name - "
F02C E5 A0 CE
F02F E1 ED E5
F032 A0 AD A0
F035 50          394          byt RTNCMD
F036          395  ;
F036 A0 00       396          ldy #ZERO
F038          397  ;
F038 B9 B8 02    398 ^1      lda FILENAME,Y
F03B 20 35 F9    399          jsr DOCOUT
F03E          400  ;
F03E C8          401          iny
F03F          402  ;
F03F CC 9E 02    403          cpy FLENGTH
F042 D0 F4       404          bne <1
F044          405  ;
F044 20 C2 F8    406          jsr PRINT
F047 8D 8D       407          byt RETURN,RETURN
F049 C6 E9 EC    408          asc "File Type - "
F04C E5 A0 D4
F04F F9 F0 E5
F052 A0 AD A0
F055 50          409          byt RTNCMD
F056          410  ;
F056 A9 07       411          lda #7
F058 8D D8 02    412          sta INDEX
F05B          413  ;
F05B AD B1 02    414          lda FILETYPE
F05E 8D D7 02    415          sta FILTYPE
F061          416  ;
F061 0E D7 02    417 ^2      asl FILTYPE
F064 90 17       418          bcc >5
F066          419  ;
F066 AE D8 02    420          ldx INDEX
F069          421  ;
F069 BC A4 FD    422          ldy TYPETBL,X
F06C          423  ;

```

```

F06C B9 AC FD      424  ^3      lda TYPTEXTS,Y
F06F F0 06         425          beq >4
F071              426  ;
F071 20 35 F9      427          jsr DOCOUT
F074              428  ;
F074 C8            429          iny
F075 D0 F5         430          bne <3
F077              431  ;
F077 20 C2 F8      432  ^4      jsr PRINT
F07A 8D            433          byt RETURN
F07B 0C            434          hex 0C
F07C 50            435          byt RTNCMD
F07D              436  ;
F07D CE D8 02      437  ^5      dec INDEX
F080 10 DF         438          bpl <2
F082              439  ;
F082 20 C2 F8      440          jsr PRINT
F085 8D            441          byt RETURN
F086 C6 E9 EC      442          asc "File Size - 0x"
F089 E5 A0 D3
F08C E9 FA E5
F08F A0 AD A0
F092 B0 F8
F094 50            443          byt RTNCMD
F095              444  ;
F095 AE B4 02      445          ldx LENVAL
F098 AD B5 02      446          lda LENVAL+1
F09B              447  ;
F09B 20 22 F9      448          jsr DOPRNTAX
F09E              449  ;
F09E 20 C2 F8      450          jsr PRINT
F0A1 8D 8D 8D      451          byt RETURN,RETURN,RETURN
F0A4 C6 F2 EF      452          asc "From EPROM Offset - 0x"
F0A7 ED A0 C5
F0AA D0 D2 CF
F0AD CD A0 CF
F0B0 E6 E6 F3
F0B3 E5 F4 A0
F0B6 AD A0 B0
F0B9 F8
F0BA 50            453          byt RTNCMD
F0BB              454  ;
F0BB AE B2 02      455          ldx SRCVAL
F0BE AD B3 02      456          lda SRCVAL+1
F0C1              457  ;
F0C1 20 22 F9      458          jsr DOPRNTAX
F0C4              459  ;
F0C4 20 C2 F8      460          jsr PRINT
F0C7 8D 8D         461          byt RETURN,RETURN
F0C9 D4 EF A0      462          asc "To Memory Address - 0x"
F0CC CD E5 ED
F0CF EF F2 F9
F0D2 A0 C1 E4
F0D5 E4 F2 E5
F0D8 F3 F3 A0
F0DB AD A0 B0
F0DE F8
F0DF 50            463          byt RTNCMD
F0E0              464  ;
F0E0 AE B6 02      465          ldx DSTVAL
F0E3 AD B7 02      466          lda DSTVAL+1

```

```

F0E6          467 ;
F0E6 4C 22 F9 468      jmp DOPRNTAX
F0E9          469 ;
F0E9          470 ;
F0E9          471 ; Use FILETYPE and RUNFLAG to LOAD or RUN the file.
F0E9          472 ;
F0E9 A0 0E    473 RUNLOAD ldy #14
F0EB          474 ;
F0EB AD B1 02 475      lda FILETYPE
F0EE 29 1F    476      and #%00011111
F0F0 F0 1F    477      beq >3
F0F2          478 ;
F0F2 0A      479 ^1      asl
F0F3 B0 04    480      bcs >2
F0F5          481 ;
F0F5 88      482      dey
F0F6 88      483      dey
F0F7          484 ;
F0F7 D0 F9    485      bne <1
F0F9          486 ;
F0F9 2C 9F 02 487 ^2      bit RUNFLAG
F0FC 10 14    488      bpl LOADFILE
F0FE          489 ;
F0FE 8C 97 02 490      sty TEMPVAL
F101          491 ;
F101 20 C2 F8 492      jsr PRINT
F104 52      493      byt INITCMD
F105 50      494      byt RTNCMD
F106          495 ;
F106 AC 97 02 496      ldy TEMPVAL
F109          497 ;
F109          498 ;
F109 B9 9B FD 499 RUNFILE lda RUNTBL+1,Y
F10C 48      500      pha
F10D          501 ;
F10D B9 9A FD 502      lda RUNTBL,Y
F110 48      503      pha
F111          504 ;
F111 60      505 ^3      rts
F112          506 ;
F112          507 ;
F112 B9 91 FD 508 LOADFILE lda LOADTBL+1,Y
F115 48      509      pha
F116          510 ;
F116 B9 90 FD 511      lda LOADTBL,Y
F119 48      512      pha
F11A          513 ;
F11A 60      514      rts
F11B          515 ;
F11B          516 ;
F11B 20 04 F0 517 LOADTEXT jsr GETFILE2
F11E          518 ;
F11E A0 00    519      ldy #PAGE09
F120 A9 09    520      lda /PAGE09
F122          521 ;
F122 8C B6 02 522 ^1      sty DSTVAL
F125 8D B7 02 523      sta DSTVAL+1
F128          524 ;
F128 20 1E F0 525      jsr SHOWFILE
F12B          526 ;
F12B 20 C2 F8 527      jsr PRINT

```



```

F12E 73          528          hex 73
F12F 59          529          byt CNTRCMD
F130 C9 F3 A0    530          asc "Is this Memory Address okay?"
F133 F4 E8 E9
F136 F3 A0 CD
F139 E5 ED EF
F13C F2 F9 A0
F13F C1 E4 E4
F142 F2 E5 F3
F145 F3 A0 EF
F148 EB E1 F9
F14B BF
F14C 76          531          hex 76
F14D 59          532          byt CNTRCMD
F14E C5 D3 C3    533          asc "ESC      (Y)es      (N)o"
F151 A0 A0 A0
F154 A0 A8 D9
F157 A9 E5 F3
F15A A0 A0 A0
F15D A0 A8 CE
F160 A9 EF
F162 50          534          byt RTNCMD
F163          535          ;
F163 20 43 F7    536          ^2      jsr GETKEY
F166          537          ;
F166 C9 9B       538          cmp #ESCAPE
F168 F0 54       539          beq >8
F16A          540          ;
F16A C9 D9       541          cmp #"Y"
F16C F0 4D       542          beq >7
F16E          543          ;
F16E C9 CE       544          cmp #"N"
F170 F0 02       545          beq >3
F172          546          ;
F172 D0 EF       547          bne <2          ; always taken
F174          548          ;
F174 20 C2 F8    549          ^3      jsr PRINT
F177 00 73       550          hex 0073
F179 58          551          byt EOPCMD
F17A C5 EE F4    552          asc "Enter Memory Address:  0x"
F17D E5 F2 A0
F180 CD E5 ED
F183 EF F2 F9
F186 A0 C1 E4
F189 E4 F2 E5
F18C F3 F3 BA
F18F A0 A0 B0
F192 F8
F193 50          553          byt RTNCMD
F194          554          ;
F194 A9 00       555          lda #ZERO
F196 85 2E       556          sta DSTPTR
F198 85 2F       557          sta DSTPTR+1
F19A          558          ;
F19A 20 78 F7    559          ^4      jsr GETNUM
F19D B0 D5       560          bcs <3
F19F          561          ;
F19F F0 F9       562          beq <4
F1A1          563          ;
F1A1 20 78 F7    564          ^5      jsr GETNUM
F1A4 B0 F4       565          bcs <4

```

```

F1A6          566 ;
F1A6 F0 F9    567      beq <5
F1A8          568 ;
F1A8 20 78 F7 569 ^6      jsr GETNUM
F1AB B0 F4    570      bcs <5
F1AD          571 ;
F1AD F0 F9    572      beq <6
F1AF          573 ;
F1AF 20 78 F7 574      jsr GETNUM
F1B2 B0 F4    575      bcs <6
F1B4          576 ;
F1B4 A4 2E    577      ldy DSTPTR
F1B6 A5 2F    578      lda DSTPTR+1
F1B8          579 ;
F1B8 4C 22 F1 580      jmp <1
F1BB          581 ;
F1BB 20 AC FA 582 ^7      jsr READBLK
F1BE          583 ;
F1BE 60       584 ^8      rts
F1BF          585 ;
F1BF          586 ;
F1BF A0 01    587 EXECAS   ldy #STARTAS
F1C1 A9 08    588      lda /STARTAS
F1C3          589 ;
F1C3 8C B6 02 590      sty DSTVAL
F1C6 8D B7 02 591      sta DSTVAL+1
F1C9          592 ;
F1C9 20 AC FA 593      jsr READBLK
F1CC          594 ;
F1CC 18       595      clc
F1CD          596 ;
F1CD A9 01    597      lda #STARTAS
F1CF 85 67    598      sta ASPGMST
F1D1          599 ;
F1D1 6D B4 02 600      adc LENVAL
F1D4          601 ;
F1D4 85 69    602      sta ASVARS
F1D6 85 AF    603      sta ASPEND
F1D8          604 ;
F1D8 A9 08    605      lda /STARTAS
F1DA 85 68    606      sta ASPGMST+1
F1DC          607 ;
F1DC 6D B5 02 608      adc LENVAL+1
F1DF          609 ;
F1DF 85 6A    610      sta ASVARS+1
F1E1 85 B0    611      sta ASPEND+1
F1E3          612 ;
F1E3 2C 9F 02 613      bit RUNFLAG
F1E6 10 D6    614      bpl <8
F1E8          615 ;
F1E8 A0 66    616      ldy #RUNAS
F1EA A9 D5    617      lda /RUNAS
F1EC          618 ;
F1EC 4C 1B ED 619      jmp EXITAS
F1EF          620 ;
F1EF          621 ;
F1EF AD 89 C0 622 EXECBIN1   lda ROM1WE
F1F2 AD 89 C0 623      lda ROM1WE
F1F5          624 ;
F1F5 4C FE F1 625      jmp EXECBIN0
F1F8          626 ;

```

```
F1F8          627 ;
F1F8 AD 81 C0 628 EXECBIN2 lda ROM2WE
F1FB AD 81 C0 629          lda ROM2WE
F1FE          630 ;
F1FE          631 ;
F1FE 20 AC FA 632 EXECBIN0 jsr READBLK
F201          633 ;
F201 2C 9F 02 634          bit RUNFLAG
F204 10 09     635          bpl >1
F206          636 ;
F206 AC B6 02 637          ldy DSTVAL
F209 AD B7 02 638          lda DSTVAL+1
F20C          639 ;
F20C 4C 1E ED 640          jmp EXITBIN
F20F          641 ;
F20F 60       642 ^1      rts
F210          643 ;
F210          644 ;
F210          645      icl "ASEOS.L"
```

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

```

F210          1          ttl "EOS+ Source Code, ASEOS.L"
F210          2          ;
F210          3          ;
F210          4          ; ASEOS.L
F210          5          ;
F210          6          ;
F210          7          ; Entry for the external user of ASEOS. Pop the stack for
F210          8          ; the address of the external routine that called ASEOS.
F210          9          ;
F210 8E A5 02    10 ASEOS      stx SLOT
F213          11          ;
F213 20 9F F4    12          jsr DOZCOFF
F216          13          ;
F216 AE A5 02    14          ldx SLOT
F219 20 5C FB    15          jsr MOVEEPBM
F21C          16          ;
F21C          17          ;
F21C          18          ; Begin ASEOS processing.
F21C          19          ;
F21C A2 00      20          ldx #ASPCMD-ASPRADRS
F21E          21          ;
F21E 20 98 F3    22          jsr GETASVAL          ; value in X-reg
F221 D0 11      23          bne >1          ; value in A-reg must be zero
F223          24          ;
F223 AD A0 02    25          lda ASPRNUM
F226 A0 FF      26          ldy #RUNMODE          ; run flag
F228          27          ;
F228 E0 01      28          cpx #LOADCMD
F22A F0 2F      29          beq >5
F22C          30          ;
F22C E0 02      31          cpx #RUNCMD
F22E F0 2C      32          beq >6
F230          33          ;
F230 E0 03      34          cpx #CATCMD
F232 F0 3F      35          beq >8
F234          36          ;
F234          37          ;
F234          38          ; Unknown Command error.
F234          39          ;
F234 A9 01      40 ^1          lda #ERR01
F236          41          ;
F236 2C 00 00    42          bit *-*
F239          43          dfs !-2
F237          44          ;
F237          45          ;
F237          46          ; Wrong Number of Parameters error.
F237          47          ;
F237 A9 02      48 ^2          lda #ERR02
F239          49          ;
F239 2C 00 00    50          bit *-*
F23C          51          dfs !-2
F23A          52          ;
F23A          53          ;
F23A          54          ; Search Range Invalid error.
F23A          55          ;
F23A A9 03      56 ^3          lda #ERR03
F23C          57          ;
F23C 2C 00 00    58          bit *-*
F23F          59          dfs !-2
F23D          60          ;

```

```

F23D      61 ;
F23D      62 ; File Not Found error.
F23D      63 ;
F23D A9 04 64 ^4      lda #ERR04
F23F      65 ;
F23F 2C 00 00 66      bit *-*
F242      67      dfs !-2
F240      68 ;
F240      69 ;
F240      70 ; Return no error.
F240      71 ;
F240 A9 00 72 ^0      lda #ERR00
F242      73 ;
F242      74 ;
F242      75 ; Exit ASEOS through EXITAS. All RUN commands are handled
F242      76 ; by RUNFILE.
F242      77 ;
F242 8D A1 02 78      sta ASSTATUS
F245      79 ;
F245 A2 02 80      ldx #ASPSTAT-ASPRADRS
F247 20 A3 F3 81      jsr SETASPTR
F24A      82 ;
F24A AE A1 02 83      ldx ASSTATUS
F24D 20 8E F3 84      jsr SAVPARM2
F250      85 ;
F250 18 86      clc
F251      87 ;
F251 68 88      pla
F252 69 01 89      adc #1
F254 A8 90      tay
F255      91 ;
F255 68 92      pla
F256 69 00 93      adc #ZERO
F258      94 ;
F258 4C 1B ED 95      jmp EXITAS
F25B      96 ;
F25B      97 ;
F25B      98 ; Command #1 processing, Load file.
F25B      99 ;
F25B C8 100 ^5      iny
F25C      101 ;
F25C      102 ;
F25C      103 ; Command #2 processing, Run file. Check ASPRNUM.
F25C      104 ;
F25C 8C 9F 02 105 ^6      sty RUNFLAG
F25F      106 ;
F25F C9 04 107      cmp #ASPNUM4
F261 F0 04 108      beq >7
F263      109 ;
F263 C9 05 110      cmp #ASPNUM5
F265 D0 D0 111      bne <2
F267      112 ;
F267      113 ;
F267      114 ; Get EPROM search range.
F267      115 ;
F267 20 60 F3 116 ^7      jsr GETEPRNG
F26A B0 CE 117      bcs <3
F26C      118 ;
F26C      119 ;
F26C      120 ; Find the requested file and process.
F26C      121 ;

```

```

F26C 20 85 F2    122          jsr DOASFILE
F26F 90 CF      123          bcc <0
F271           124          ;
F271 B0 CA      125          bcs <4                ; always taken
F273           126          ;
F273           127          ;
F273           128          ; Command #3 processing, read Catalog.  Check ASPRNUM.
F273           129          ;
F273 C9 05      130          ^8      cmp #ASPNUM5
F275 F0 04      131          beq >9
F277           132          ;
F277 C9 06      133          cmp #ASPNUM6
F279 D0 BC      134          bne <2
F27B           135          ;
F27B           136          ;
F27B           137          ; Get EPROM search range.
F27B           138          ;
F27B 20 60 F3   139          ^9      jsr GETEPRNG
F27E B0 BA      140          bcs <3
F280           141          ;
F280 20 AD F2   142          jsr DOASCAT
F283 90 BB      143          bcc <0                ; always taken
F285           144          ;
F285           145          ;
F285           146          ; Load or run the requested EPROM file.  Applesoft string
F285           147          ; variables require three bytes:  length, low address, high
F285           148          ; address.  The first value returned in A-reg is length.
F285           149          ; X-reg contains the low address.  The third value is the
F285           150          ; high address of the string.
F285           151          ;
F285 A2 06      152          DOASFILE ldx #ASPFIL-ASPRADRS
F287 20 98 F3   153          jsr GETASVAL
F28A           154          ;
F28A 8D A3 02   155          sta FILELEN
F28D           156          ;
F28D A0 02      157          ldy #2
F28F           158          ;
F28F B1 CE      159          lda (GENPTR),Y        ; high address
F291           160          ;
F291 20 40 FA   161          jsr FINDFILE
F294 B0 16      162          bcs >2
F296           163          ;
F296           164          ;
F296           165          ; Found a matching filename.
F296           166          ;
F296 AE A0 02   167          ldx ASPRNUM
F299 E0 04      168          cpx #ASPNUM4
F29B F0 0B      169          beq >1
F29D           170          ;
F29D           171          ;
F29D           172          ; Get optional run/load address.
F29D           173          ;
F29D A2 08      174          ldx #ASPADR-ASPRADRS
F29F 20 98 F3   175          jsr GETASVAL
F2A2           176          ;
F2A2 8E B6 02   177          stx DSTVAL
F2A5 8D B7 02   178          sta DSTVAL+1
F2A8           179          ;
F2A8           180          ;
F2A8           181          ; If FILETYPE is zero, RUNLOAD will return, otherwise if
F2A8           182          ; the RUNFLAG is LOAD, a routine from LOADTBL will LOAD

```

```

F2A8      183 ; the file and return. If the RUNFLAG is RUN, a routine
F2A8      184 ; from RUNTBL will LOAD the file and exit EOS+ via EXITAS.
F2A8      185 ;
F2A8 20 E9 F0 186 ^1      jsr RUNLOAD
F2AB      187 ;
F2AB 18      188      clc
F2AC      189 ;
F2AC 60      190 ^2      rts
F2AD      191 ;
F2AD      192 ;
F2AD      193 ; Read the requested EPROM catalog.
F2AD      194 ;
F2AD AD 98 02 195 DOASCAT  lda EPSTRT
F2B0 8D 94 02 196      sta EPNMBR
F2B3      197 ;
F2B3 20 CC F9 198 ^1      jsr INITCAT
F2B6 B0 0D      199      bcs >3
F2B8      200 ;
F2B8      201 ;
F2B8      202 ; Get a Catalog entry.
F2B8      203 ;
F2B8 20 E5 F9 204 ^2      jsr GETENTRY
F2BB      205 ;
F2BB AD B1 02 206      lda FILETYPE
F2BE F0 05      207      beq >3
F2C0      208 ;
F2C0      209 ;
F2C0      210 ; Record the entry.
F2C0      211 ;
F2C0 20 D1 F2 212      jsr SAVFILE
F2C3 90 F3      213      bcc <2                ; always taken
F2C5      214 ;
F2C5      215 ;
F2C5      216 ; Check next EPROM.
F2C5      217 ;
F2C5 EE 94 02 218 ^3      inc EPNMBR
F2C8      219 ;
F2C8 AD 99 02 220      lda EPEND
F2CB CD 94 02 221      cmp EPNMBR
F2CE B0 E3      222      bcs <1
F2D0      223 ;
F2D0 60      224      rts
F2D1      225 ;
F2D1      226 ;
F2D1      227 ; Record the entry.
F2D1      228 ;
F2D1 A2 06      229 SAVFILE  ldx #ASPNUM-ASPRADRS
F2D3 20 A3 F3 230      jsr SETASPTR
F2D6      231 ;
F2D6 18      232      clc
F2D7      233 ;
F2D7 C8      234      iny
F2D8      235 ;
F2D8 B1 CE      236      lda (GENPTR),Y
F2DA 69 01      237      adc #1
F2DC 91 CE      238      sta (GENPTR),Y
F2DE      239 ;
F2DE 88      240      dey
F2DF      241 ;
F2DF B1 CE      242      lda (GENPTR),Y
F2E1 69 00      243      adc #ZERO

```

```

F2E3 91 CE      244      sta (GENPTR),Y
F2E5           245      ;
F2E5           246      ;
F2E5           247      ; Get memory space for the filename.
F2E5           248      ;
F2E5 AD 9E 02   249      lda FLENGTH
F2E8 85 08      250      sta AREG
F2EA           251      ;
F2EA A0 D5      252      ldy #STRINI
F2EC A9 E3      253      lda /STRINI
F2EE           254      ;
F2EE 20 43 F9   255      jsr JSRMEM
F2F1           256      ;
F2F1           257      ;
F2F1           258      ; Record the filename length and address.
F2F1           259      ;
F2F1 A2 08      260      ldx #ASPFILES-ASPRADRS
F2F3 20 A3 F3   261      jsr SETASPTR
F2F6           262      ;
F2F6 A0 02      263      ldy #2
F2F8           264      ;
F2F8 B9 9D 00   265      ^1  lda DSCTMP,Y
F2FB 91 CE      266      sta (GENPTR),Y
F2FD           267      ;
F2FD 88         268      dey
F2FE 10 F8      269      bpl <1
F300           270      ;
F300 18         271      clc
F301           272      ;
F301 A5 CE      273      lda GENPTR
F303 69 03      274      adc #3
F305 8D D8 02   275      sta ASPFILES
F308           276      ;
F308 A5 CF      277      lda GENPTR+1
F30A 69 00      278      adc #ZERO
F30C 8D D9 02   279      sta ASPFILES+1
F30F           280      ;
F30F           281      ;
F30F           282      ; Copy the filename.
F30F           283      ;
F30F A0 00      284      ldy #ZERO
F311           285      ;
F311 B9 B8 02   286      ^2  lda FILENAME,Y
F314 29 7F      287      and #MSBCLR
F316 91 9E      288      sta (DSCTMP+1),Y
F318           289      ;
F318 C8         290      iny
F319           291      ;
F319 C4 9D      292      cpy DSCTMP
F31B D0 F4      293      bne <2
F31D           294      ;
F31D AE A0 02   295      ldx ASPRNUM
F320 E0 05      296      cpx #ASPNUM5
F322 F0 3A      297      beq RTNCLC
F324           298      ;
F324           299      ;
F324           300      ; Initialize GENPTR with the ASPPARMS array.
F324           301      ;
F324 A2 0A      302      ldx #ASPPARMS-ASPRADRS
F326 20 A3 F3   303      jsr SETASPTR
F329           304      ;

```



```

F329          305 ;
F329          306 ; Save EPNMBR.
F329          307 ;
F329 AE 94 02 308         ldx EPNMBR
F32C 20 8E F3 309         jsr SAVPARM2
F32F          310 ;
F32F          311 ;
F32F          312 ; Save file type.
F32F          313 ;
F32F AE B1 02 314         ldx FILETYPE
F332 20 8E F3 315         jsr SAVPARM2
F335          316 ;
F335          317 ;
F335          318 ; Save EPROM offset.
F335          319 ;
F335 AE B2 02 320         ldx SRCVAL
F338 AD B3 02 321         lda SRCVAL+1
F33B          322 ;
F33B 20 90 F3 323         jsr SAVPARM
F33E          324 ;
F33E          325 ;
F33E          326 ; Save file length.
F33E          327 ;
F33E AE B4 02 328         ldx LENVAL
F341 AD B5 02 329         lda LENVAL+1
F344          330 ;
F344 20 90 F3 331         jsr SAVPARM
F347          332 ;
F347          333 ;
F347          334 ; Save memory address.
F347          335 ;
F347 AE B6 02 336         ldx DSTVAL
F34A AD B7 02 337         lda DSTVAL+1
F34D          338 ;
F34D 20 90 F3 339         jsr SAVPARM
F350          340 ;
F350          341 ;
F350          342 ; Point to next array entry.
F350          343 ;
F350 18        344         clc
F351          345 ;
F351 98        346         tya
F352 65 CE     347         adc GENPTR
F354 8D DA 02  348         sta ASPPARMS
F357          349 ;
F357 A5 CF     350         lda GENPTR+1
F359 69 00     351         adc #ZERO
F35B 8D DB 02  352         sta ASPPARMS+1
F35E          353 ;
F35E 18        354 RTNCLC   clc
F35F          355 ;
F35F 60        356         rts
F360          357 ;
F360          358 ;
F360          359 ; Get the EPROM search range in X-reg and verify.
F360          360 ;
F360 A2 04     361 GETEPRNG ldx #ASPSRCH-ASPRADRS
F362          362 ;
F362 20 98 F3  363         jsr GETASVAL
F365 D0 25     364         bne >2
F367          365 ;

```

; value in A-reg must be zero

```

F367 8E A2 02    366  GETRANGE stx EPSEARCH
F36A              367  ;
F36A 8A          368              txa
F36B              369  ;
F36B              370              .if HWCARD
F36B              371              cmp #EPMASK+1
F36B              372              .el
F36B C9 08        373              cmp #QLMASK+1
F36D              374              .fi
F36D              375  ;
F36D 90 12        376              bcc >1
F36F              377  ;
F36F 29 0F        378              and #EPMASK
F371              379  ;
F371              380              .if HWCARD
F371              381              cmp #EPMASK+1
F371              382              .el
F371 C9 08        383              cmp #QLMASK+1
F373              384              .fi
F373              385  ;
F373 B0 18        386              bcs >3
F375              387  ;
F375 AA           388              tax
F376              389  ;
F376 AD A2 02     390              lda EPSEARCH
F379              391  ;
F379 4A           392              lsr
F37A 4A           393              lsr
F37B 4A           394              lsr
F37C 4A           395              lsr
F37D              396  ;
F37D              397              .if HWCARD
F37D              398              cmp #EPMASK+1
F37D              399              .el
F37D C9 08        400              cmp #QLMASK+1
F37F              401              .fi
F37F              402  ;
F37F B0 0C        403              bcs >3
F381              404  ;
F381 8E 98 02     405  ^1          stx EPSTRT
F384 8D 99 02     406              sta EPEND
F387              407  ;
F387 CD 98 02     408              cmp EPSTRT
F38A B0 D2        409              bcs RTNCLC
F38C              410  ;
F38C 38           411  ^2          sec
F38D              412  ;
F38D 60           413  ^3          rts
F38E              414  ;
F38E              415  ;
F38E              416  ; Save Applesoft value in ASPPARMS array.
F38E              417  ;
F38E A9 00        418  SAVPARM2 lda #ZERO
F390              419  ;
F390 91 CE        420  SAVPARM  sta (GENPTR),Y
F392              421  ;
F392 C8           422              iny
F393              423  ;
F393 8A           424              txa
F394 91 CE        425              sta (GENPTR),Y
F396              426  ;

```

```
F396 C8          427          iny
F397            428          ;
F397 60          429          rts
F398            430          ;
F398            431          ;
F398            432          ; Get the requested Applesoft value.
F398            433          ;
F398 20 A3 F3    434  GETASVAL jsr SETASPTR
F39B            435          ;
F39B C8          436          iny
F39C            437          ;
F39C B1 CE       438          lda (GENPTR),Y
F39E AA          439          tax
F39F            440          ;
F39F 88          441          dey
F3A0            442          ;
F3A0 B1 CE       443          lda (GENPTR),Y
F3A2            444          ;
F3A2 60          445          rts
F3A3            446          ;
F3A3            447          ;
F3A3            448          ; Set GENPTR to the requested address.
F3A3            449          ;
F3A3 BC D0 02    450  SETASPTR ldy ASPRADRS,X
F3A6 BD D1 02    451          lda ASPRADRS+1,X
F3A9            452          ;
F3A9 84 CE       453          sty GENPTR
F3AB 85 CF       454          sta GENPTR+1
F3AD            455          ;
F3AD A0 00       456          ldy #ZERO
F3AF            457          ;
F3AF 60          458          rts
F3B0            459          ;
F3B0            460          ;
F3B0            461          icl "BINEOS.L"
```

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

```

F3B0          1          ttl "EOS+ Source Code, BINEOS.L"
F3B0          2          ;
F3B0          3          ;
F3B0          4          ; BINEOS.L
F3B0          5          ;
F3B0          6          ;
F3B0          7          ; Entry for the external user of BINEOS. Pop the stack for
F3B0          8          ; the address of the external routine that called BINEOS.
F3B0          9          ;
F3B0 8E A5 02 10 BINEOS stx SLOT
F3B3          11         ;
F3B3 84 EE    12         sty CMDPTR
F3B5 85 EF    13         sta CMDPTR+1
F3B7          14         ;
F3B7 20 9F F4 15         jsr DOZCOFF
F3BA          16         ;
F3BA AE A5 02 17         ldx SLOT
F3BD 20 5C FB 18         jsr MOVEEPBM
F3C0          19         ;
F3C0 A0 07    20         ldy #DCBSIZE-1
F3C2          21         ;
F3C2 B1 EE    22 ^1      lda (CMDPTR),Y
F3C4 99 D0 02 23         sta DCBBUFR,Y
F3C7          24         ;
F3C7 88       25         dey
F3C8 10 F8    26         bpl <1
F3CA          27         ;
F3CA A9 02    28         lda #EXTERNAL
F3CC          29         ;
F3CC          30         ;
F3CC          31         ; Initialize Y-reg with RUNMODE and process DCB command.
F3CC          32         ; Assume CMDPTR contains the address of DCBBUFR.
F3CC          33         ;
F3CC 8D 96 02 34 BINEOS2 sta RTNTYPE
F3CF          35         ;
F3CF A0 FF    36         ldy #RUNMODE
F3D1          37         ;
F3D1 AD D0 02 38         lda DCBCMD
F3D4          39         ;
F3D4 C9 01    40         cmp #1                ; load file
F3D6 F0 30    41         beq BINLOAD
F3D8          42         ;
F3D8 C9 02    43         cmp #2                ; run file
F3DA F0 2D    44         beq BINRUN
F3DC          45         ;
F3DC C9 03    46         cmp #3                ; catalog EPROM
F3DE F0 5F    47         beq BINCAT
F3E0          48         ;
F3E0          49         ;
F3E0          50         ; Return Unknown Command error.
F3E0          51         ;
F3E0 A9 01    52         lda #ERR01
F3E2          53         ;
F3E2 2C 00 00 54         bit *-*
F3E5          55         dfs !-2
F3E3          56         ;
F3E3          57         ;
F3E3          58         ; Return Filename Length Invalid error.
F3E3          59         ;
F3E3 A9 02    60 ^2      lda #ERR02

```

```

F3E5          61 ;
F3E5 2C 00 00 62          bit *-*
F3E8          63          dfs !-2
F3E6          64 ;
F3E6          65 ;
F3E6          66 ; Return Search Range Invalid error.
F3E6          67 ;
F3E6 A9 03     68 ^3      lda #ERR03
F3E8          69 ;
F3E8 2C 00 00 70          bit *-*
F3EB          71          dfs !-2
F3E9          72 ;
F3E9          73 ;
F3E9          74 ; Buffer/Filename Address error.
F3E9          75 ;
F3E9 A9 04     76 ^4      lda #ERR04
F3EB          77 ;
F3EB 2C 00 00 78          bit *-*
F3EE          79          dfs !-2
F3EC          80 ;
F3EC          81 ;
F3EC          82 ; Requested File Not Found error.
F3EC          83 ;
F3EC A9 05     84 ^5      lda #ERR05
F3EE          85 ;
F3EE 2C 00 00 86          bit *-*
F3F1          87          dfs !-2
F3EF          88 ;
F3EF          89 ;
F3EF          90 ; Exit BINEOS through EXITBIN for an external caller or
F3EF          91 ; return to MAIN for an internal caller for a LOAD DCB.
F3EF          92 ; All RUN DCBs are handled by RUNFILE.
F3EF          93 ;
F3EF A9 00     94 BINDONE  lda #ERR00
F3F1          95 ;
F3F1 A0 04     96          ldy #DCBSTAT-DCBCMD
F3F3          97 ;
F3F3 91 EE     98          sta (CMDPTR),Y
F3F5          99 ;
F3F5 AD 96 02 100         lda RTNTYPE
F3F8 C9 01     101         cmp #INTERNAL
F3FA F0 0B     102         beq >1
F3FC          103 ;
F3FC 18        104         clc
F3FD          105 ;
F3FD 68        106         pla
F3FE 69 01     107         adc #1
F400 A8        108         tay
F401          109 ;
F401 68        110         pla
F402 69 00     111         adc #ZERO
F404          112 ;
F404 4C 1E ED  113         jmp EXITBIN
F407          114 ;
F407 60        115 ^1      rts
F408          116 ;
F408          117 ;
F408          118 ; Command #1 processing, Load file.
F408          119 ;
F408 C8        120 BINLOAD  iny
F409          121 ;

```

```

F409      122 ;
F409      123 ; Command 2 processing, Run file.
F409      124 ;
F409 8C 9F 02 125 BINRUN    sty RUNFLAG
F40C      126 ;
F40C      127 ;
F40C      128 ; Get the filename length and verify.
F40C      129 ;
F40C AD D5 02 130          lda DCBFLEN
F40F C9 18    131          cmp #NAME SIZE
F411 B0 D0    132          bcs <2
F413      133 ;
F413 8D A3 02 134          sta FILELEN
F416      135 ;
F416      136 ;
F416      137 ; Get EPROM search range.
F416      138 ;
F416 AE D1 02 139          ldx DCBEPN
F419      140 ;
F419 20 67 F3 141          jsr GETRANGE
F41C B0 C8    142          bcs <3
F41E      143 ;
F41E      144 ;
F41E      145 ; Find the requested file and process.
F41E      146 ;
F41E AE D6 02 147          ldx DCBFADR
F421 AD D7 02 148          lda DCBFADR+1
F424 F0 C3    149          beq <4
F426      150 ;
F426 20 40 FA 151          jsr FINDFILE
F429 B0 C1    152          bcs <5
F42B      153 ;
F42B      154 ;
F42B      155 ; Get alternate run/load address.
F42B      156 ;
F42B AE D2 02 157          ldx DCBFALT
F42E AD D3 02 158          lda DCBFALT+1
F431 F0 06    159          beq >1
F433      160 ;
F433 8E B6 02 161          stx DSTVAL
F436 8D B7 02 162          sta DSTVAL+1
F439      163 ;
F439      164 ;
F439      165 ; If FILETYPE is zero, RUNLOAD will return, otherwise if
F439      166 ; the RUNFLAG is LOAD, a routine from LOADTBL will LOAD
F439      167 ; the file and return. If the RUNFLAG is RUN, a routine
F439      168 ; from RUNTBL will LOAD the file and exit EOS+ via EXITBIN
F439      169 ; with Y-reg/A-reg initialized from DSTVAL and DSTVAL+1.
F439      170 ;
F439 20 E9 F0 171 ^1      jsr RUNLOAD
F43C      172 ;
F43C 4C EF F3 173          jmp BINDONE
F43F      174 ;
F43F      175 ;
F43F      176 ; Command 3 processing, read Catalog.
F43F      177 ;
F43F      178 ; Initialize number of entries.
F43F      179 ;
F43F A9 00    180 BINCAT    lda #ZERO
F441 8D D5 02 181          sta DCBFLEN
F444      182 ;

```

```

F444      183 ;
F444      184 ; Get EPROM search range.
F444      185 ;
F444 AE D1 02 186      ldx DCBEPN
F447      187 ;
F447 20 67 F3 188      jsr GETRANGE
F44A B0 9A 189      bcs <3
F44C      190 ;
F44C      191 ;
F44C      192 ; Get buffer address.
F44C      193 ;
F44C AE D6 02 194      ldx DCBFADR
F44F AD D7 02 195      lda DCBFADR+1
F452 F0 95 196      beq <4
F454      197 ;
F454 86 CE 198      stx GENPTR
F456 85 CF 199      sta GENPTR+1
F458      200 ;
F458      201 ;
F458      202 ; Read the Catalog.
F458      203 ;
F458 AD 98 02 204      lda EPSTRT
F45B 8D 94 02 205      sta EPNMBR
F45E      206 ;
F45E 20 CC F9 207 ^1      jsr INITCAT
F461 B0 27 208      bcs >5
F463      209 ;
F463      210 ;
F463      211 ; Get a Catalog entry.
F463      212 ;
F463 20 E5 F9 213 ^2      jsr GETENTRY
F466      214 ;
F466 AD B1 02 215      lda FILETYPE
F469 F0 1F 216      beq >5
F46B      217 ;
F46B      218 ;
F46B      219 ; Copy the entry.
F46B      220 ;
F46B A0 1F 221      ldy #ENTRYLEN-1
F46D      222 ;
F46D B9 B0 02 223 ^3      lda FILEENTRY,Y
F470 91 CE 224      sta (GENPTR),Y
F472      225 ;
F472 88 226      dey
F473 10 F8 227      bpl <3
F475      228 ;
F475 18 229      clc
F476      230 ;
F476 A5 CE 231      lda GENPTR
F478 69 20 232      adc #ENTRYLEN
F47A 85 CE 233      sta GENPTR
F47C      234 ;
F47C 90 02 235      bcc >4
F47E      236 ;
F47E E6 CF 237      inc GENPTR+1
F480      238 ;
F480 EE D5 02 239 ^4      inc DCBFLEN
F483 D0 DE 240      bne <2
F485      241 ;
F485 CE D5 02 242      dec DCBFLEN
F488 D0 0B 243      bne >6

```

```

; overflow, so stop
; always taken

```

```
F48A          244  ;
F48A          245  ;
F48A          246  ; Check next EPROM.
F48A          247  ;
F48A EE 94 02  248  ^5      inc EPNMBR
F48D          249  ;
F48D AD 99 02  250          lda EPEND
F490 CD 94 02  251          cmp EPNMBR
F493 B0 C9     252          bcs <1
F495          253  ;
F495 A0 05     254  ^6      ld y #DCBFLEN-DCBCMD
F497          255  ;
F497 AD D5 02  256          lda DCBFLEN
F49A 91 EE     257          sta (CMDPTR),Y
F49C          258  ;
F49C 4C EF F3  259          jmp BINDONE
F49F          260  ;
F49F          261  ;
F49F          262          icl "ZIP.L"
```

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



```

F49F          1          ttl "EOS+ Source Code, ZIP.L"
F49F          2          ;
F49F          3          ;
F49F          4          ; ZIP.L
F49F          5          ;
F49F          6          ;
F49F          7          ; Unlock the ZipChip, test if present, get its status, set
F49F          8          ; slots 1 and 6 to normal, and disable it.
F49F          9          ;
F49F A9 80     10 DOZCOFF  lda #$80
F4A1 8D 9A 02  11          sta ZSTATUS
F4A4          12          ;
F4A4 20 77 F5  13          jsr DOZCOPEN
F4A7          14          ;
F4A7 AD 5C C0  15          lda ZCSLOTS
F4AA 49 FF     16          eor #NEGONE
F4AC 8D 5C C0  17          sta ZCSLOTS
F4AF          18          ;
F4AF CD 5C C0  19          cmp ZCSLOTS
F4B2 D0 1A     20          bne SETANNUN
F4B4          21          ;
F4B4 49 FF     22          eor #NEGONE
F4B6 8D 5C C0  23          sta ZCSLOTS
F4B9          24          ;
F4B9 CD 5C C0  25          cmp ZCSLOTS
F4BC D0 10     26          bne SETANNUN
F4BE          27          ;
F4BE 8E 9A 02  28          stx ZSTATUS          ; save state of Bit 4
F4C1          29          ;
F4C1 A9 40     30          lda #%01000000          ; paddle fast, LC cache OFF
F4C3 8D 9B 02  31          sta ZCACHE
F4C6          32          ;
F4C6 A9 C0     33          lda #%11000000          ; paddle normal, LC cache OFF
F4C8 8D 5F C0  34          sta ZCCACHE
F4CB          35          ;
F4CB 8D 5A C0  36          sta ZCCTRL          ; disables ZipChip
F4CE          37          ;
F4CE AD 58 C0  38 SETANNUN lda ANN1OFF
F4D1 AD 5A C0  39          lda ANN2OFF
F4D4 AD 5D C0  40          lda ANN3ON
F4D7 AD 5F C0  41          lda ANN4ON
F4DA          42          ;
F4DA 60        43          rts
F4DB          44          ;
F4DB          45          ;
F4DB          46          ; Unlock the ZipChip if present and if it was on before,
F4DB          47          ; turn it back on.
F4DB          48          ;
F4DB AD 9A 02  49 DOZCON  lda ZSTATUS
F4DE 30 13     50          bmi >2
F4E0          51          ;
F4E0 20 77 F5  52          jsr DOZCOPEN
F4E3          53          ;
F4E3 AD 9A 02  54          lda ZSTATUS          ; BNE to disable ZipChip
F4E6 D0 08     55          bne >1
F4E8          56          ;
F4E8 AD 9B 02  57          lda ZCACHE          ; recall paddle fast, LC OFF
F4EB 8D 5F C0  58          sta ZCCACHE          ; save setting
F4EE          59          ;
F4EE A9 A5     60          lda #ZCLOCK          ; lock ZipChip

```

```

F4F0      61 ;
F4F0 8D 5A C0 62 ^1      sta ZCCTRL
F4F3      63 ;
F4F3 4C CE F4 64 ^2      jmp SETANNUN
F4F6      65 ;
F4F6      66 ;
F4F6      67 ; Reset the ZipChip in a power up state.
F4F6      68 ;
F4F6 AD 9A 02 69 DOZCRSET lda ZSTATUS
F4F9 30 12     70      bmi >1
F4FB      71 ;
F4FB 20 77 F5 72      jsr DOZCOPEN
F4FE      73 ;
F4FE AD 82 FD 74      lda ZCDEFLT      ; slots
F501 8D 5C C0 75      sta ZCSLOTS
F504      76 ;
F504 AD 83 FD 77      lda ZCDEFLT+1    ; speed
F507 8D 5D C0 78      sta ZCSPEED
F50A      79 ;
F50A 8D 5A C0 80      sta ZCCTRL      ; lock ZipChip
F50D      81 ;
F50D 4C CE F4 82 ^1      jmp SETANNUN
F510      83 ;
F510      84 ;
F510      85 ; Read the ZipChip configuration; unable to read current
F510      86 ; ZipChip speed. Assume it is set to maximum speed.
F510      87 ;
F510 20 77 F5 88 DOZCREAD jsr DOZCOPEN
F513      89 ;
F513 A0 0B     90      ldy #ZCOPTNS-1
F515      91 ;
F515 A9 00     92      lda #ZERO
F517      93 ;
F517 99 D0 02 94 ^1      sta ZCSETBL,Y
F51A      95 ;
F51A 88       96      dey
F51B 10 FA    97      bpl <1
F51D      98 ;
F51D 8A       99      txa      ; BEQ if ZipChip enabled
F51E F0 03   100      beq >2
F520     101 ;
F520 EE D0 02 102      inc ZCSETBL      ; state
F523     103 ;
F523     104 ;
F523     105 ; Extract default language card caching.
F523     106 ;
F523 AD 9B 02 107 ^2      lda ZCACHE
F526     108 ;
F526 0A      109      asl
F527 2E D2 02 110      rol ZCSETBL+2      ; language card cache
F52A     111 ;
F52A     112 ;
F52A     113 ; Extract default paddle delay.
F52A     114 ;
F52A 0A      115      asl
F52B 2E D3 02 116      rol ZCSETBL+3
F52E     117 ;
F52E A2 07    118      ldx #7
F530     119 ;
F530 AD 5C C0 120      lda ZCSLOTS
F533     121 ;

```

```

F533 0A          122  ^3      asl
F534 3E D4 02    123      rol ZCSETBL+4,X      ; slots and speaker
F537             124      ;
F537 CA          125      dex
F538 10 F9       126      bpl <3
F53A             127      ;
F53A 8D 5A C0    128      sta ZCCTRL
F53D             129      ;
F53D 60          130      rts
F53E             131      ;
F53E             132      ;
F53E             133      ; Save the ZipChip configuration.
F53E             134      ;
F53E 20 77 F5    135 DOZCSAVE jsr DOZCOPEN
F541             136      ;
F541 AD D0 02    137      lda ZCSETBL      ; state
F544 29 01       138      and #1
F546 F0 02       139      beq >1
F548             140      ;
F548 A9 10       141      lda #ZCOFFVAL
F54A             142      ;
F54A 8D 9A 02    143 ^1      sta ZSTATUS
F54D             144      ;
F54D AE D1 02    145      ldx ZCSETBL+1      ; speed
F550             146      ;
F550 BD 84 FD    147      lda ZCSPDTBL,X
F553 8D 5D C0    148      sta ZCSPEED
F556             149      ;
F556 A9 00       150      lda #ZERO
F558             151      ;
F558 4E D3 02    152      lsr ZCSETBL+3      ; paddle
F55B 6A          153      ror
F55C             154      ;
F55C 4E D2 02    155      lsr ZCSETBL+2      ; language card
F55F 6A          156      ror
F560             157      ;
F560 8D 9B 02    158      sta ZCACHE
F563             159      ;
F563 A2 07       160      ldx #7
F565             161      ;
F565 A9 00       162      lda #ZERO
F567             163      ;
F567 5E D4 02    164 ^2      lsr ZCSETBL+4,X
F56A 2A          165      rol
F56B             166      ;
F56B CA          167      dex
F56C 10 F9       168      bpl <2
F56E             169      ;
F56E 8D 5C C0    170      sta ZCSLOTS
F571             171      ;
F571 A9 00       172      lda #ZERO
F573 8D 5A C0    173      sta ZCCTRL
F576             174      ;
F576 60          175      rts
F577             176      ;
F577             177      ;
F577 A9 5A       178 DOZCOPEN lda #ZCUNLOCK
F579             179      ;
F579 8D 5A C0    180      sta ZCCTRL
F57C 8D 5A C0    181      sta ZCCTRL
F57F 8D 5A C0    182      sta ZCCTRL

```

```

F582 8D 5A C0      183      sta ZCCTRL
F585              184      ;
F585 AD 5B C0      185      lda ZCSTATS          ; get ZipChip status
F588 29 10          186      and #ZCSTAT          ; mask out enabled bit
F58A AA            187      tax
F58B              188      ;
F58B 8D 5B C0      189      sta ZCSTATS          ; enable ZipChip
F58E              190      ;
F58E A0 D8          191      ldy #DOZCOPEN-DOZCOFF
F590              192      ;
F590 B9 9E F4      193      ^1 lda DOZCOFF-1,Y
F593              194      ;
F593 88            195      dey
F594 D0 FA          196      bne <1
F596              197      ;
F596 60            198      rts
F597              199      ;
F597              200      ;
F597 20 C2 F8      201      ZCONFIG jsr PRINT
F59A 55            202      byt HOMECMD
F59B 59            203      byt CNTRCMD
F59C DA E9 F0      204      asc "ZipChip Configuration"
F59F C3 E8 E9
F5A2 F0 A0 C3
F5A5 EF EE E6
F5A8 E9 E7 F5
F5AB F2 E1 F4
F5AE E9 EF EE
F5B1 00 63          205      hex 0063
F5B3 DA E9 F0      206      asc "ZipChip State"
F5B6 C3 E8 E9
F5B9 F0 A0 D3
F5BC F4 E1 F4
F5BF E5
F5C0 8D 8D          207      byt RETURN,RETURN
F5C2 DA E9 F0      208      asc "ZipChip Speed"
F5C5 C3 E8 E9
F5C8 F0 A0 D3
F5CB F0 E5 E5
F5CE E4
F5CF 8D 8D          209      byt RETURN,RETURN
F5D1 CC E1 EE      210      asc "Language Card Caching"
F5D4 E7 F5 E1
F5D7 E7 E5 A0
F5DA C3 E1 F2
F5DD E4 A0 C3
F5E0 E1 E3 E8
F5E3 E9 EE E7
F5E6 8D 8D          211      byt RETURN,RETURN
F5E8 A0 D0 E1      212      asc " Paddle Speed"
F5EB E4 E4 EC
F5EE E5 A0 D3
F5F1 F0 E5 E5
F5F4 E4
F5F5 8D            213      byt RETURN
F5F6 D3 F0 E5      214      asc "Speaker Speed"
F5F9 E1 EB E5
F5FC F2 A0 D3
F5FF F0 E5 E5
F602 E4
F603 8D            215      byt RETURN

```

```

F604 50          216      byt RTNCMD
F605          217      ;
F605 A9 B1      218      lda #"1"
F607 8D 97 02   219      sta TEMPVAL
F60A          220      ;
F60A 20 C2 F8   221      ^1 jsr PRINT
F60D 8D          222      byt RETURN
F60E A0 D3 EC   223      asc " Slot "
F611 EF F4 A0
F614 50          224      byt RTNCMD
F615          225      ;
F615 AD 97 02   226      lda TEMPVAL
F618 20 35 F9   227      jsr DOCOUT
F61B          228      ;
F61B 20 C2 F8   229      jsr PRINT
F61E A0 D3 F0   230      asc " Speed"
F621 E5 E5 E4
F624 50          231      byt RTNCMD
F625          232      ;
F625 EE 97 02   233      inc TEMPVAL
F628          234      ;
F628 AD 97 02   235      lda TEMPVAL
F62B C9 B8      236      cmp #"8"
F62D D0 DB      237      bne <1
F62F          238      ;
F62F 20 C2 F8   239      jsr PRINT
F632 75          240      hex 75
F633 59          241      byt CNTRCMD
F634 BC AD A0   242      asc "<- Select ->"
F637 D3 E5 EC
F63A E5 E3 F4
F63D A0 AD BE
F640 77          243      hex 77
F641 59          244      byt CNTRCMD
F642 A8 D0 A9   245      asc "(P)revious      (N)ext      (Q)uit"
F645 F2 E5 F6
F648 E9 EF F5
F64B F3 A0 A0
F64E A0 A0 A8
F651 CE A9 E5
F654 F8 F4 A0
F657 A0 A0 A0
F65A A8 D1 A9
F65D F5 E9 F4
F660 18 63      246      hex 1863
F662 50          247      byt RTNCMD
F663          248      ;
F663 20 10 F5   249      jsr DOZCREAD
F666          250      ;
F666 A2 0B      251      ldx #ZCOPTNS-1
F668 8E 9D 02   252      stx NUMSELC
F66B          253      ;
F66B 20 D1 F6   254      ^1 jsr ZCDISP
F66E          255      ;
F66E CE 9D 02   256      dec NUMSELC
F671 10 F8      257      bpl <1
F673          258      ;
F673 A2 00      259      ^2 ldx #ZERO
F675          260      ;
F675          261      ;
F675 8E 9D 02   262      ZCLOOP stx NUMSELC

```

```

F678      263 ;
F678 20 D1 F6 264      jsr ZCDISP
F67B      265 ;
F67B 20 43 F7 266 ^3      jsr GETKEY
F67E      267 ;
F67E AE 9D 02 268      ldx NUMSELC
F681      269 ;
F681 C9 88 270      cmp #LARROW
F683 F0 04 271      beq >4
F685      272 ;
F685 C9 8B 273      cmp #UARROW
F687 D0 07 274      bne >5
F689      275 ;
F689 CA 276 ^4      dex
F68A 10 E9 277      bpl ZCLOOP
F68C      278 ;
F68C A2 0B 279      ldx #ZCOPTNS-1
F68E D0 E5 280      bne ZCLOOP ; always taken
F690      281 ;
F690 C9 95 282 ^5      cmp #RARROW
F692 F0 08 283      beq >6
F694      284 ;
F694 C9 8A 285      cmp #DARROW
F696 F0 04 286      beq >6
F698      287 ;
F698 C9 8D 288      cmp #RETURN
F69A D0 07 289      bne >7
F69C      290 ;
F69C E8 291 ^6      inx
F69D      292 ;
F69D E0 0C 293      cpx #ZCOPTNS
F69F D0 D4 294      bne ZCLOOP
F6A1      295 ;
F6A1 F0 D0 296      beq <2 ; always taken
F6A3      297 ;
F6A3 C9 D1 298 ^7      cmp #"Q"
F6A5 D0 03 299      bne >8
F6A7      300 ;
F6A7 4C 3E F5 301      jmp DOZCSAVE
F6AA      302 ;
F6AA C9 D0 303 ^8      cmp #"P"
F6AC D0 05 304      bne >9
F6AE      305 ;
F6AE FE D0 02 306      inc ZCSETBL,X
F6B1 D0 07 307      bne >1 ; always taken
F6B3      308 ;
F6B3 C9 CE 309 ^9      cmp #"N"
F6B5 D0 C4 310      bne <3
F6B7      311 ;
F6B7 DE D0 02 312      dec ZCSETBL,X
F6BA      313 ;
F6BA E0 01 314 ^1      cpx #1
F6BC D0 B7 315      bne ZCLOOP
F6BE      316 ;
F6BE AD D1 02 317      lda ZCSETBL+1
F6C1 10 02 318      bpl >2
F6C3      319 ;
F6C3 A9 03 320      lda #ZCNSPEED-1
F6C5      321 ;
F6C5 C9 04 322 ^2      cmp #ZCNSPEED
F6C7 90 02 323      bcc >3

```

```

F6C9          324 ;
F6C9 A9 00    325     lda #ZERO
F6CB          326 ;
F6CB 8D D1 02 327 ^3     sta ZCSETBL+1
F6CE          328 ;
F6CE 4C 75 F6 329     jmp ZCLOOP
F6D1          330 ;
F6D1          331 ;
F6D1 20 C2 F8 332 ZCDISP   jsr PRINT
F6D4 18       333     hex 18
F6D5 A0 A0    334     asc "  "
F6D7 50       335     byt RTNCMD
F6D8          336 ;
F6D8 AE 9D 02 337     ldx NUMSELC
F6DB          338 ;
F6DB BD 6B FD 339     lda LINETBL,X
F6DE 85 25    340     sta CV
F6E0          341 ;
F6E0 20 1A F9 342     jsr DOVTAB
F6E3          343 ;
F6E3 20 C2 F8 344     jsr PRINT
F6E6 18       345     hex 18
F6E7 AD BE A0 346     asc "-> "
F6EA 50       347     byt RTNCMD
F6EB          348 ;
F6EB AC 9D 02 349     ldy NUMSELC
F6EE          350 ;
F6EE B9 D0 02 351     lda ZCSETBL,Y
F6F1          352 ;
F6F1 C0 00    353     cpy #ZERO
F6F3 D0 08    354     bne >1
F6F5          355 ;
F6F5 29 01    356     and #1
F6F7 F0 25    357     beq >4
F6F9          358 ;
F6F9 A0 03    359     ldy #ZCTEXT1-ZCTEXT
F6FB D0 21    360     bne >4 ; always taken
F6FD          361 ;
F6FD C0 01    362 ^1     cpy #1
F6FF D0 07    363     bne >2
F701          364 ;
F701 0A       365     asl
F702 0A       366     asl
F703          367 ;
F703 69 06    368     adc #ZCTEXT2-ZCTEXT
F705 A8       369     tay
F706          370 ;
F706 D0 16    371     bne >4 ; always taken
F708          372 ;
F708 C0 02    373 ^2     cpy #2
F70A D0 0A    374     bne >3
F70C          375 ;
F70C A0 16    376     ldy #ZCTEXT3-ZCTEXT
F70E          377 ;
F70E 29 01    378     and #1
F710 F0 0C    379     beq >4
F712          380 ;
F712 A0 1E    381     ldy #ZCTEXT4-ZCTEXT
F714 D0 08    382     bne >4 ; always taken
F716          383 ;
F716 A0 26    384 ^3     ldy #ZCTEXT5-ZCTEXT

```

```
F718          385 ;
F718 29 01     386      and #1
F71A F0 02     387      beq >4
F71C          388 ;
F71C A0 2C     389      ldy #ZCTEXT6-ZCTEXT
F71E          390 ;
F71E B9 3E FE  391      ^4      lda ZCTEXT,Y
F721 48        392      pha
F722          393 ;
F722 09 80     394      ora #$80
F724 20 35 F9  395      jsr DOCOUT
F727          396 ;
F727 C8        397      iny
F728          398 ;
F728 68        399      pla
F729 30 F3     400      bmi <4
F72B          401 ;
F72B 60        402      rts
F72C          403 ;
F72C          404 ;
F72C          405      icl "SUBS.L"
```

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



```
F72C          1          ttl "EOS+ Source Code, SUBS.L"
F72C          2          ;
F72C          3          ;
F72C          4          ; SUBS.L
F72C          5          ;
F72C          6          ;
F72C 38        7          EOSBELL  sec
F72D          8          ;
F72D A9 80     9          lda #$80
F72F          10         ;
F72F A0 80    11         ^1      ldy #$80
F731          12         ;
F731 88       13         ^2      dey
F732 D0 FD    14         bne <2
F734          15         ;
F734 2C 30 C0 16         bit SPKR
F737          17         ;
F737 E9 01    18         sbc #1
F739 D0 F4    19         bne <1
F73B          20         ;
F73B 60       21         rts
F73C          22         ;
F73C          23         ;
F73C A9 60    24         RDKEY   lda #$60
F73E 20 35 F9 25         jsr DOCOUT
F741          26         ;
F741 C6 24    27         dec CH
F743          28         ;
F743 AD 00 C0 29         GETKEY  lda KEY
F746          30         ;
F746 EA       31         nop
F747          32         ;
F747 10 FA    33         bpl GETKEY
F749          34         ;
F749 2C 10 C0 35         bit CLRKEY
F74C          36         ;
F74C C9 FF    37         cmp #NEGONE
F74E D0 02    38         bne >1
F750          39         ;
F750 A9 88    40         lda #LARROW
F752          41         ;
F752 C9 E0    42         ^1      cmp #LWRCASE
F754 90 02    43         bcc >2
F756          44         ;
F756 29 DF    45         and #LWRMASK
F758          46         ;
F758 60       47         ^2      rts
F759          48         ;
F759          49         ;
F759 C9 B0    50         GETHEX  cmp #"0"
F75B 90 19    51         bcc >3
F75D          52         ;
F75D C9 BA    53         cmp #"9"+1
F75F 90 08    54         bcc >1
F761          55         ;
F761 C9 C1    56         cmp #"A"
F763 90 11    57         bcc >3
F765          58         ;
F765 C9 C7    59         cmp #"F"+1
F767 B0 0D    60         bcs >3
```

```

F769          61 ;
F769 48        62 ^1 pha
F76A          63 ;
F76A C9 BA     64 cmp #"9"+1
F76C 90 02     65 bcc >2
F76E          66 ;
F76E E9 07     67 sbc #7
F770          68 ;
F770 29 0F     69 ^2 and #VALUMASK
F772 AA        70 tax
F773          71 ;
F773 68        72 pla
F774          73 ;
F774 18        74 clc
F775          75 ;
F775 60        76 rts
F776          77 ;
F776 38        78 ^3 sec
F777          79 ;
F777 60        80 rts
F778          81 ;
F778          82 ;
F778          83 GETNUM:
F778 20 3C F7  84 ^1 jsr RDKEY
F77B          85 ;
F77B C9 8D     86 cmp #RETURN
F77D F0 23     87 beq >2
F77F          88 ;
F77F C9 88     89 cmp #LARROW
F781 F0 21     90 beq >3
F783          91 ;
F783 20 59 F7  92 jsr GETHEX
F786 B0 F0     93 bcs <1
F788          94 ;
F788 20 35 F9  95 jsr DOCOUT
F78B          96 ;
F78B 06 2E     97 asl DSTPTR
F78D 26 2F     98 rol DSTPTR+1
F78F 06 2E     99 asl DSTPTR
F791 26 2F    100 rol DSTPTR+1
F793 06 2E    101 asl DSTPTR
F795 26 2F    102 rol DSTPTR+1
F797 06 2E    103 asl DSTPTR
F799 26 2F    104 rol DSTPTR+1
F79B          105 ;
F79B 8A        106 txa
F79C 05 2E    107 ora DSTPTR
F79E 85 2E    108 sta DSTPTR
F7A0          109 ;
F7A0 A9 01    110 lda #1
F7A2          111 ;
F7A2 18        112 ^2 clc
F7A3          113 ;
F7A3 60        114 rts
F7A4          115 ;
F7A4 20 30 F9  116 ^3 jsr DOSPACE
F7A7          117 ;
F7A7 C6 24    118 dec CH
F7A9 C6 24    119 dec CH
F7AB          120 ;
F7AB 38        121 sec

```

```

F7AC          122 ;
F7AC 60       123      rts
F7AD          124 ;
F7AD          125 ;
F7AD          126 GETVAL:
F7AD A2 00    127 ^1      ldx #ZERO
F7AF          128 ;
F7AF 20 3C F7 129 ^2      jsr RDKEY
F7B2          130 ;
F7B2 C9 88    131      cmp #LARROW
F7B4 D0 0C    132      bne >4
F7B6          133 ;
F7B6 20 30 F9 134 ^3      jsr DOSPACE
F7B9          135 ;
F7B9 C6 24    136      dec CH
F7BB          137 ;
F7BB CA       138      dex
F7BC 30 EF    139      bmi <1
F7BE          140 ;
F7BE C6 24    141      dec CH
F7C0 10 ED    142      bpl <2
F7C2          143 ;
F7C2 C9 8D    144 ^4      cmp #RETURN
F7C4 F0 1C    145      beq >5
F7C6          146 ;
F7C6 C9 B0    147      cmp #"0"
F7C8 90 E5    148      bcc <2
F7CA          149 ;
F7CA C9 BA    150      cmp #"9"+1
F7CC B0 E1    151      bcs <2
F7CE          152 ;
F7CE 20 35 F9 153      jsr DOCOUT
F7D1          154 ;
F7D1 29 0F    155      and #VALUMASK
F7D3 9D 9C 02 156      sta NUMIN,X
F7D6          157 ;
F7D6 E8       158      inx
F7D7          159 ;
F7D7 E0 03    160      cpx #3
F7D9 D0 D4    161      bne <2
F7DB          162 ;
F7DB 20 3C F7 163      jsr RDKEY
F7DE          164 ;
F7DE C9 88    165      cmp #LARROW
F7E0 F0 D4    166      beq <3
F7E2          167 ;
F7E2 CA       168 ^5      dex
F7E3 30 14    169      bmi >8
F7E5          170 ;
F7E5 A0 00    171      ldy #ZERO
F7E7          172 ;
F7E7 18       173      clc
F7E8          174 ;
F7E8 BD 9C 02 175      lda NUMIN,X
F7EB          176 ;
F7EB CA       177 ^6      dex
F7EC 30 0B    178      bmi >8
F7EE          179 ;
F7EE C8       180      iny
F7EF          181 ;
F7EF DE 9C 02 182 ^7      dec NUMIN,X

```

```

F7F2 30 F7      183      bmi <6
F7F4           184      ;
F7F4 79 77 FD   185      adc DECTBLL,Y
F7F7 90 F6      186      bcc <7
F7F9           187      ;
F7F9 60         188      ^8      rts
F7FA           189      ;
F7FA           190      ;
F7FA           191      ; Fall into PRTSDV.
F7FA           192      ;
F7FA           193      EDITS DV:
F7FA 20 C2 F8   194      ^1      jsr PRINT
F7FD 00 76      195      hex 0076
F7FF C5 EE F4   196      asc "Enter Slot:  "
F802 E5 F2 A0
F805 D3 EC EF
F808 F4 BA A0
F80B A0
F80C 57         197      byt EOLCMD
F80D 50         198      byt RTNCMD
F80E           199      ;
F80E 20 AD F7   200      jsr GETVAL
F811 B0 0C      201      bcs >2
F813           202      ;
F813 A8         203      tay
F814 F0 E4      204      beq <1
F816           205      ;
F816 C9 08      206      cmp #7+1
F818 B0 E0      207      bcs <1
F81A           208      ;
F81A 85 EB      209      sta MSLOT
F81C           210      ;
F81C 20 62 F8   211      jsr PRTSDV
F81F           212      ;
F81F 20 C2 F8   213      ^2      jsr PRINT
F822 00 76      214      hex 0076
F824 C5 EE F4   215      asc "Enter Drive:  "
F827 E5 F2 A0
F82A C4 F2 E9
F82D F6 E5 BA
F830 A0 A0
F832 57         216      byt EOLCMD
F833 50         217      byt RTNCMD
F834           218      ;
F834 20 AD F7   219      jsr GETVAL
F837 B0 0C      220      bcs >3
F839           221      ;
F839 A8         222      tay
F83A F0 E3      223      beq <2
F83C           224      ;
F83C C9 52      225      cmp #81+1
F83E B0 DF      226      bcs <2
F840           227      ;
F840 85 EC      228      sta DRIVE
F842           229      ;
F842 20 62 F8   230      jsr PRTSDV
F845           231      ;
F845 20 C2 F8   232      ^3      jsr PRINT
F848 00 76      233      hex 0076
F84A C5 EE F4   234      asc "Enter Volume:  "
F84D E5 F2 A0

```

```

F850 D6 EF EC
F853 F5 ED E5
F856 BA A0 A0
F859 57          235          byt EOLCMD
F85A 50          236          byt RTNCMD
F85B          237          ;
F85B 20 AD F7    238          jsr GETVAL
F85E B0 39       239          bcs RTN01
F860          240          ;
F860 85 ED       241          sta VOLUME
F862          242          ;
F862          243          ;
F862          244          ; Fall into PRTDEC.
F862          245          ;
F862 20 C2 F8    246 PRTSDV   jsr PRINT
F865 00 71       247          hex 0071          ; position cursor on line 17
F867 50          248          byt RTNCMD
F868          249          ;
F868 A2 01       250          ldx #1          ; 1 digit
F86A A0 04       251          ldy #4          ; CH
F86C A5 EB       252          lda MSLOT
F86E          253          ;
F86E 20 80 F8    254          jsr PRTDEC
F871          255          ;
F871 A2 02       256          ldx #2          ; 2 digits
F873 A0 08       257          ldy #8          ; CH
F875 A5 EC       258          lda DRIVE
F877          259          ;
F877 20 80 F8    260          jsr PRTDEC
F87A          261          ;
F87A A2 03       262          ldx #3          ; 3 digits
F87C A0 0D       263          ldy #13         ; CH
F87E A5 ED       264          lda VOLUME
F880          265          ;
F880          266          ;
F880 84 24       267 PRTDEC   sty CH
F882          268          ;
F882 A0 B0       269 ^1      ldy #"0"
F884          270          ;
F884 38          271          sec
F885          272          ;
F885 FD 76 FD    273 ^2      sbc DECTBLL-1,X
F888 90 03       274          bcc >3
F88A          275          ;
F88A C8          276          iny
F88B D0 F8       277          bne <2
F88D          278          ;
F88D 7D 76 FD    279 ^3      adc DECTBLL-1,X
F890 48          280          pha
F891          281          ;
F891 98          282          tya
F892 20 35 F9    283          jsr DOCOUT
F895          284          ;
F895 68          285          pla
F896          286          ;
F896 CA          287          dex
F897 D0 E9       288          bne <1
F899          289          ;
F899 60          290 RTN01    rts
F89A          291          ;
F89A          292          ;

```

```

F89A      293 ; Find the slot number of the next EPROM card from SLOTMAP.
F89A      294 ; If the slot number exceeds slot 7, start over at slot 1.
F89A      295 ; The value in SLOTMAP must be greater than 1 to be valid.
F89A      296 ; Return with the slot page value of 0xCs in the A-reg.
F89A      297 ;
F89A AE 91 02 298 NEXTMAP ldx EPSLOT
F89D      299 ;
F89D E8      300 ^1      inx
F89E      301 ;
F89E E0 08    302          cpx #8
F8A0 D0 02    303          bne >2
F8A2      304 ;
F8A2 A2 01    305          ldx #1
F8A4      306 ;
F8A4 AD 92 02 307 ^2      lda SLOTMAP
F8A7      308 ;
F8A7 3D 7A FD 309          and MAPMASKS,X
F8AA F0 F1    310          beq <1
F8AC      311 ;
F8AC 8A      312          txa
F8AD 09 C0    313          ora /PAGEC0
F8AF      314 ;
F8AF A0 D0    315          ldy #EPMAPEOS
F8B1      316 ;
F8B1 60      317          rts
F8B2      318 ;
F8B2      319 ;
F8B2 18      320 CLRUSER clc
F8B3      321 ;
F8B3 B0 00    322          bcs *+2
F8B5      323          dfs !-1
F8B4      324 ;
F8B4      325 ;
F8B4 38      326 SETUSER sec
F8B5      327 ;
F8B5 84 07    328          sty YREG
F8B7 85 08    329          sta AREG
F8B9      330 ;
F8B9 AC F6 BF 331          ldy MNGUSER
F8BC AD F7 BF 332          lda MNGUSER+1
F8BF      333 ;
F8BF 4C 43 F9 334          jmp JSRMEM
F8C2      335 ;
F8C2      336 ;
F8C2 68      337 PRINT   pla
F8C3 85 FC    338          sta PRNTPTR
F8C5 68      339          pla
F8C6 85 FD    340          sta PRNTPTR+1
F8C8      341 ;
F8C8 E6 FC    342 ^1      inc PRNTPTR
F8CA D0 02    343          bne >2
F8CC      344 ;
F8CC E6 FD    345          inc PRNTPTR+1
F8CE      346 ;
F8CE A0 00    347 ^2      ldy #ZERO
F8D0      348 ;
F8D0 B1 FC    349          lda (PRNTPTR),Y
F8D2 10 06    350          bpl >3
F8D4      351 ;
F8D4 20 35 F9 352          jsr DOCOUT
F8D7      353 ;

```

```

F8D7 4C C8 F8      354      jmp <1
F8DA              355      ;
F8DA C9 50         356      ^3      cmp #MAXCH
F8DC B0 04         357      bcs >4
F8DE              358      ;
F8DE 85 24         359      sta CH
F8E0              360      ;
F8E0 90 E6         361      bcc <1          ; always taken
F8E2              362      ;
F8E2 20 E8 F8      363      ^4      jsr PRINT01
F8E5              364      ;
F8E5 4C C8 F8      365      jmp <1
F8E8              366      ;
F8E8 C9 60         367      PRINT01  cmp #MINCV
F8EA 90 06         368      bcc >5
F8EC              369      ;
F8EC 29 1F         370      and #CVMASK
F8EE 85 25         371      sta CV
F8F0              372      ;
F8F0 A9 56         373      lda #TABVCMD
F8F2              374      ;
F8F2 29 0F         375      ^5      and #PRNTMASK
F8F4 AA           376      tax
F8F5 D0 0D         377      bne >6
F8F7              378      ;
F8F7 BA           379      tsx
F8F8              380      ;
F8F8 A5 FC         381      lda PRNTPTR
F8FA 9D 01 01      382      sta STACK+1,X
F8FD A5 FD         383      lda PRNTPTR+1
F8FF 9D 02 01      384      sta STACK+2,X
F902              385      ;
F902 18           386      clc
F903              387      ;
F903 60           388      rts
F904              389      ;
F904 E0 09         390      ^6      cpx #CNTRCMD&PRNTMASK
F906 D0 31         391      bne DOJSRMEM
F908              392      ;
F908 A9 9F         393      lda #SPACE-1
F90A              394      ;
F90A C8           395      ^7      iny
F90B              396      ;
F90B D1 FC         397      cmp (PRNTPTR),Y
F90D 90 FB         398      bcc <7
F90F              399      ;
F90F 98           400      tya
F910              401      ;
F910 49 FF         402      eor #NEGONE
F912 65 21         403      adc WNDWIDTH
F914              404      ;
F914 4A           405      lsr
F915              406      ;
F915 65 20         407      adc WNDLFT
F917 85 24         408      sta CH
F919              409      ;
F919 60           410      rts
F91A              411      ;
F91A              412      ;
F91A A2 06         413      DOVTAB  ldx #6
F91C D0 1B         414      bne DOJSRMEM          ; always taken

```

```

F91E          415 ;
F91E          416 ;
F91E A2 09    417 DOSHOOK   ldx #9
F920 D0 17    418          bne DOJSRMEM          ; always taken
F922          419 ;
F922          420 ;
F922 86 06    421 DOPRNTAX  stx XREG
F924          422 ;
F924 A2 0A    423          ldx #10
F926 D0 11    424          bne DOJSRMEM          ; always taken
F928          425 ;
F928          426 ;
F928 A2 0B    427 DOPRBYTE  ldx #11
F92A D0 0D    428          bne DOJSRMEM          ; always taken
F92C          429 ;
F92C          430 ;
F92C A2 0C    431 DOPRHEX   ldx #12
F92E D0 09    432          bne DOJSRMEM          ; always taken
F930          433 ;
F930          434 ;
F930 A9 A0    435 DOSPACE   lda #SPACE
F932          436 ;
F932 2C 00 00 437          bit *-*
F935          438          dfs !-2
F933          439 ;
F933 A9 8D    440 DOCROUT   lda #RETURN
F935          441 ;
F935 86 06    442 DOCOUT    stx XREG
F937          443 ;
F937 A2 00    444          ldx #ZERO
F939          445 ;
F939          446 ;
F939          447 ; Save Y-reg and A-reg; use X-reg to load routine address.
F939          448 ; Fall into JSRMEM.
F939          449 ;
F939 84 07    450 DOJSRMEM  sty YREG
F93B 85 08    451          sta AREG
F93D          452 ;
F93D BC 51 FD 453          ldy ADDRBTBLL,X
F940 BD 5E FD 454          lda ADDRBTBLH,X
F943          455 ;
F943          456 ;
F943          457 ; Y-reg and A-reg values need to be saved at YREG and AREG
F943          458 ; before loading Y-reg and A-reg with address of routine
F943          459 ; to execute outside of the EPROM card. The X-reg value
F943          460 ; is safe and does not need to be saved.
F943          461 ;
F943 8C A6 02 462 JSRMEM    sty MEMJMP
F946 8D A7 02 463          sta MEMJMP+1
F949          464 ;
F949 4C 4C 01 465          jmp EPJSR
F94C          466 ;
F94C          467 ;
F94C          468 ; Load DOS 4.5L into Main memory.
F94C          469 ;
F94C A0 04    470 LOADOSL   ldy #DOSLPRMS-CATALOG
F94E 20 7F FA 471          jsr COPYPRM0
F951          472 ;
F951 AD 82 C0 473          lda ROM2WP
F954          474 ;
F954 A0 00    475          ldy #ZERO

```



```

F956 A9 21      476      lda /PAGEC0-PAGE9F
F958           477      ;
F958 84 2C      478      sty LENPTR
F95A 85 2D      479      sta LENPTR+1
F95C           480      ;
F95C A0 00      481      ld y #PAGE9F
F95E A9 9F      482      lda /PAGE9F
F960           483      ;
F960 84 2E      484      sty DSTPTR
F962 85 2F      485      sta DSTPTR+1
F964           486      ;
F964 4C 1D 01   487      jmp EPMOVE
F967           488      ;
F967           489      ;
F967           490      ; Load DOS 4.5H into the Language card and Main memory.
F967           491      ;
F967 A0 15      492      LOADOSH ld y #DOSHPRMS-CATALOG
F969 20 7F FA   493      jsr COPYPRM0
F96C           494      ;
F96C           495      ;
F96C           496      ; Load Language Card Bank 2.
F96C           497      ;
F96C AD 81 C0   498      lda ROM2WE
F96F AD 81 C0   499      lda ROM2WE
F972           500      ;
F972 A0 00      501      ld y #ZERO
F974 A9 1A      502      lda /PAGEEA-PAGED0
F976           503      ;
F976 84 2C      504      sty LENPTR
F978 85 2D      505      sta LENPTR+1
F97A           506      ;
F97A A0 00      507      ld y #PAGED0
F97C A9 D0      508      lda /PAGED0
F97E           509      ;
F97E 20 A5 F9   510      jsr LOADOSH2
F981           511      ;
F981           512      ;
F981           513      ; Load Language Card Bank 1.
F981           514      ;
F981 AD 89 C0   515      lda ROM1WE
F984 AD 89 C0   516      lda ROM1WE
F987           517      ;
F987 A0 00      518      ld y #ZERO
F989 A9 0E      519      lda /PAGEDE-PAGED0
F98B           520      ;
F98B 84 2C      521      sty LENPTR
F98D 85 2D      522      sta LENPTR+1
F98F           523      ;
F98F A0 00      524      ld y #PAGED0
F991 A9 D0      525      lda /PAGED0
F993           526      ;
F993 20 A5 F9   527      jsr LOADOSH2
F996           528      ;
F996           529      ;
F996           530      ; Write protect Language card and load remaining memory.
F996           531      ;
F996 AD 82 C0   532      lda ROM2WP
F999           533      ;
F999 A0 00      534      ld y #ZERO
F99B A9 02      535      lda /PAGEC0-PAGEBE
F99D           536      ;

```

```

F99D 84 2C      537      sty LENPTR
F99F 85 2D      538      sta LENPTR+1
F9A1           539      ;
F9A1 A0 00      540      ldy #PAGEBE
F9A3 A9 BE      541      lda /PAGEBE
F9A5           542      ;
F9A5 84 2E      543 LOADOSH2 sty DSTPTR
F9A7 85 2F      544      sta DSTPTR+1
F9A9           545      ;
F9A9 4C 1D 01   546      jmp EPMOVE
F9AC           547      ;
F9AC           548      ;
F9AC           549      ; Load F8 ROM into the Language card.
F9AC           550      ;
F9AC A2 00      551 COPYROM ldx #ZERO
F9AE 8E 94 02   552      stx EPNMBR
F9B1           553      ;
F9B1 BD BC E0   554      ^1  lda ROMPARMS+1,X
F9B4 95 2A      555      sta SRCPTR,X
F9B6           556      ;
F9B6 E8         557      inx
F9B7           558      ;
F9B7 E0 06      559      cpx #PARMSIZE
F9B9 D0 F6      560      bne <1
F9BB           561      ;
F9BB AD 81 C0   562      lda ROM2WE
F9BE AD 81 C0   563      lda ROM2WE
F9C1           564      ;
F9C1 A2 00      565      ldx #ZERO
F9C3           566      ;
F9C3           567      .if HWCARD
F9C3           568      lda #EPOFFVAL
F9C3           569      .el
F9C3 A9 10      570      lda #QLOFFVAL
F9C5           571      .fi
F9C5           572      ;
F9C5 20 20 01   573      jsr EPMOVE2
F9C8           574      ;
F9C8 AD 82 C0   575      lda ROM2WP
F9CB           576      ;
F9CB 60         577      rts
F9CC           578      ;
F9CC           579      ;
F9CC           580      ; Initializes the pointers to begin reading an EPROM's
F9CC           581      ; catalog. The four catalog sync bytes are verified
F9CC           582      ; first.
F9CC           583      ;
F9CC A0 CA      584 INITCAT ldy #CATPARMS-CATALOG
F9CE 20 84 FA   585      jsr COPYPARM
F9D1           586      ;
F9D1 20 AC FA   587      jsr READBLK
F9D4           588      ;
F9D4 A0 03      589      ldy #SYNC.L-1
F9D6           590      ;
F9D6 B9 AC 02   591      ^1  lda SYNCBUFR,Y
F9D9 D9 70 FE   592      cmp SYNCBYTS,Y
F9DC D0 05      593      bne >2
F9DE           594      ;
F9DE 88         595      dey
F9DF 10 F5      596      bpl <1
F9E1           597      ;

```

```

F9E1 18          598          clc
F9E2          599          ;
F9E2 60          600          rts
F9E3          601          ;
F9E3 38          602          ^2      sec
F9E4          603          ;
F9E4 60          604          rts
F9E5          605          ;
F9E5          606          ;
F9E5          607          ; Gets the catalog entry currently pointed to.  Points to
F9E5          608          ; the next entry on exit.  Assumes INITCAT was previously
F9E5          609          ; called.  FILEENTRY+1 points to FILETYPE, thus ENTRYLEN is
F9E5          610          ; reduced by one byte.
F9E5          611          ;
F9E5 A5 2A       612  GETENTRY  lda  SRCPTR
F9E7 48          613          pha
F9E8 A5 2B       614          lda  SRCPTR+1
F9EA 48          615          pha
F9EB          616          ;
F9EB A0 1F       617          ldy  #ENTRYLEN-1
F9ED A9 00       618          lda  /ENTRYLEN-1
F9EF          619          ;
F9EF 84 2C       620          sty  LENPTR
F9F1 85 2D       621          sta  LENPTR+1
F9F3          622          ;
F9F3 A0 B1       623          ldy  #FILEENTRY+1
F9F5 A9 02       624          lda  /FILEENTRY+1
F9F7          625          ;
F9F7 84 2E       626          sty  DSTPTR
F9F9 85 2F       627          sta  DSTPTR+1
F9FB          628          ;
F9FB 20 1D 01    629          jsr  EPMOVE
F9FE          630          ;
F9FE 68          631          pla
F9FF 85 2B       632          sta  SRCPTR+1
FA01 68          633          pla
FA02 85 2A       634          sta  SRCPTR
FA04          635          ;
FA04 AD 94 02    636          lda  EPNMBR
FA07 8D B0 02    637          sta  FILEPNUM
FA0A          638          ;
FA0A A0 00       639          ldy  #ZERO
FA0C 8C 9E 02    640          sty  FLENGTH
FA0F          641          ;
FA0F B9 B8 02    642          ^1      lda  FILENAME,Y
FA12 48          643          pha
FA13          644          ;
FA13 09 80       645          ora  #MSBSET
FA15 99 B8 02    646          sta  FILENAME,Y
FA18          647          ;
FA18 EE 9E 02    648          inc  FLENGTH
FA1B          649          ;
FA1B 68          650          pla
FA1C 30 07       651          bmi  >2
FA1E          652          ;
FA1E C8          653          iny
FA1F          654          ;
FA1F C0 18       655          cpy  #NAME SIZE
FA21 D0 EC       656          bne  <1
FA23          657          ;
FA23 F0 0C       658          beq  >4

```

```

FA25          659 ;
FA25 A9 A0    660 ^2      lda #SPACE
FA27          661 ;
FA27 C8       662 ^3      iny
FA28          663 ;
FA28 C0 18    664          cpy #NAME SIZE
FA2A F0 05    665          beq >4
FA2C          666 ;
FA2C 99 B8 02 667          sta FILENAME,Y
FA2F          668 ;
FA2F D0 F6    669          bne <3          ; always taken
FA31          670 ;
FA31 18       671 ^4      clc
FA32          672 ;
FA32 A9 07    673          lda #PARMSIZE+1
FA34 6D 9E 02 674          adc FLENGTH
FA37          675 ;
FA37 65 2A    676          adc SRCPTR
FA39 85 2A    677          sta SRCPTR
FA3B 90 02    678          bcc >5
FA3D          679 ;
FA3D E6 2B    680          inc SRCPTR+1
FA3F          681 ;
FA3F 60       682 ^5      rts
FA40          683 ;
FA40          684 ;
FA40          685 ; Begin with EPSTRT and locate its Catalog.
FA40          686 ;
FA40 86 CE    687 FINDFILE stx GENPTR
FA42 85 CF    688          sta GENPTR+1
FA44          689 ;
FA44 AD 98 02 690          lda EPSTRT
FA47 8D 94 02 691          sta EPNMBR
FA4A          692 ;
FA4A 20 CC F9 693 ^1      jsr INITCAT
FA4D B0 23    694          bcs >4
FA4F          695 ;
FA4F          696 ;
FA4F          697 ; Get a Catalog entry. If FILETYPE is zero, at ENDCAT.
FA4F          698 ;
FA4F 20 E5 F9 699 ^2      jsr GETENTRY
FA52          700 ;
FA52 AD B1 02 701          lda FILETYPE
FA55 F0 1B    702          beq >4
FA57          703 ;
FA57          704 ;
FA57          705 ; Compare filename lengths.
FA57          706 ;
FA57 AD 9E 02 707          lda FLENGTH
FA5A CD A3 02 708          cmp FILELEN
FA5D D0 F0    709          bne <2
FA5F          710 ;
FA5F          711 ;
FA5F          712 ; Compare filenames.
FA5F          713 ;
FA5F A0 00    714          ldy #ZERO
FA61          715 ;
FA61 B1 CE    716 ^3      lda (GENPTR),Y
FA63 09 80    717          ora #MSBSET
FA65          718 ;
FA65 D9 B8 02 719          cmp FILENAME,Y

```

```

FA68 D0 E5      720      bne <2
FA6A           721      ;
FA6A C8         722      iny
FA6B           723      ;
FA6B CC A3 02   724      cpy FILELEN
FA6E D0 F1      725      bne <3
FA70           726      ;
FA70           727      ;
FA70           728      ; Found a matching filename.
FA70           729      ;
FA70 18         730      clc
FA71           731      ;
FA71 60         732      rts
FA72           733      ;
FA72           734      ;
FA72           735      ; Check next EPROM.
FA72           736      ;
FA72 EE 94 02   737      ^4      inc EPNMBR
FA75           738      ;
FA75 AD 99 02   739      lda EPEND
FA78 CD 94 02   740      cmp EPNMBR
FA7B B0 CD      741      bcs <1
FA7D           742      ;
FA7D           743      ;
FA7D           744      ; Filename not found.
FA7D           745      ;
FA7D 38         746      sec
FA7E           747      ;
FA7E 60         748      rts
FA7F           749      ;
FA7F           750      ;
FA7F           751      ; Copy the file's parameter block pointed to by Y-reg.
FA7F           752      ; Fall into SELCBANK.
FA7F           753      ;
FA7F A9 00      754      COPYPRM0 lda #ZERO
FA81 8D 94 02   755      sta EPNMBR
FA84           756      ;
FA84 A2 00      757      COPYPARM ldx #ZERO
FA86           758      ;
FA86 B9 00 E0   759      ^1      lda CATALOG,Y
FA89 9D B1 02   760      sta FILETYPE,X
FA8C           761      ;
FA8C C8         762      iny
FA8D E8         763      inx
FA8E           764      ;
FA8E E0 07      765      cpx #PARMSIZE+1
FA90 D0 F4      766      bne <1
FA92           767      ;
FA92           768      ;
FA92           769      ; Extract the EPROM bank from the EPROM offset address.
FA92           770      ;
FA92 AC B2 02   771      SELCBANK ldy SRCVAL
FA95 AD B3 02   772      lda SRCVAL+1
FA98           773      ;
FA98 4A         774      lsr
FA99 4A         775      lsr
FA9A 4A         776      lsr
FA9B 4A         777      lsr
FA9C 4A         778      lsr
FA9D           779      ;
FA9D 8D 95 02   780      sta EPBANK

```

```

FAA0          781  ;
FAA0 AD B3 02 782      lda SRCVAL+1
FAA3 29 1F    783      and #BANKMASK
FAA5          784  ;
FAA5          785      .if DEBUG
FAA5          786      ora /PAGE08
FAA5          787      .el
FAA5 09 E0    788      ora /PAGEE0
FAA7          789      .fi
FAA7          790  ;
FAA7 84 2A    791      sty SRCPTR
FAA9 85 2B    792      sta SRCPTR+1
FAAB          793  ;
FAAB 60       794      rts
FAAC          795  ;
FAAC          796  ;
FAAC          797  ; Read a block of data from an EPROM.
FAAC          798  ;
FAAC 20 92 FA 799 READBLK jsr SELCBANK
FAAF          800  ;
FAAF AC B4 02 801      ldy LENVAL
FAB2 AD B5 02 802      lda LENVAL+1
FAB5          803  ;
FAB5 84 2C    804      sty LENPTR
FAB7 85 2D    805      sta LENPTR+1
FAB9          806  ;
FAB9 AC B6 02 807      ldy DSTVAL
FABC AD B7 02 808      lda DSTVAL+1
FABF          809  ;
FABF 84 2E    810      sty DSTPTR
FAC1 85 2F    811      sta DSTPTR+1
FAC3          812  ;
FAC3 4C 1D 01 813      jmp EPMOVE
FAC6          814  ;
FAC6          815  ;
FAC6          816  ; Read two bytes of address data from memory.
FAC6          817  ;
FAC6 84 2A    818 READADR sty SRCPTR
FAC8 85 2B    819      sta SRCPTR+1
FACA          820  ;
FACA A0 02    821      ldy #2
FACC A9 00    822      lda #ZERO
FACE          823  ;
FACE 84 2C    824      sty LENPTR
FAD0 85 2D    825      sta LENPTR+1
FAD2          826  ;
FAD2 A0 AA    827      ldy #ADDRBUFR
FAD4 A9 02    828      lda /ADDRBUFR
FAD6          829  ;
FAD6 84 2E    830      sty DSTPTR
FAD8 85 2F    831      sta DSTPTR+1
FADA          832  ;
FADA A2 00    833      ldx #ZERO
FADC          834  ;
FADC          835      .if HWCARD
FADC          836      lda #EPOFFVAL
FADC          837      .el
FADC A9 10    838      lda #QLOFFVAL
FADE          839      .fi
FADE          840  ;
FADE 4C 20 01 841      jmp EPMOVE2

```

```
FAE1      842  ;  
FAE1      843  ;  
FAE1      844      icl "PAGE1.L"
```

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

```

FAE1      1          ttl "EOS+ Source Code, PAGE1.L"
FAE1      2      ;
FAE1      3      ;
FAE1      4      ; PAGE1.L
FAE1      5      ;
FAE1      6      ;
FAE1      7      ; Use the slot value in X-reg to prepare the slot*16 value
FAE1      8      ; for the Device Select index for this EPROM card. Slot 3
FAE1      9      ; ROM is already enabled.
FAE1     10      ;
FAE1  8A     11  BUILDMAP txa
FAE2     12      ;
FAE2  0A     13          asl
FAE3  0A     14          asl
FAE4  0A     15          asl
FAE5  0A     16          asl
FAE6     17      ;
FAE6  8D A4 02  18          sta SLOT16
FAE9  AA     19          tax
FAEA     20      ;
FAEA     21      ;
FAEA     22      ; Move the EPROM mapping code to the stack.
FAEA     23      ;
FAEA  A0 49     24          ldy #EPMAPLEN-1
FAEC     25      ;
FAEC  B9 12 FB  26      ^1    lda MAPCODE,Y
FAEF  99 10 01  27          sta MAPPAGE,Y
FAF2     28      ;
FAF2  88     29          dey
FAF3  10 F7    30          bpl <1
FAF5     31      ;
FAF5     32      ;
FAF5     33      ; Initialize the page-zero pointers used in slot mapping,
FAF5     34      ; then make the map of the slots.
FAF5     35      ;
FAF5  C8     36          iny
FAF6  8C 92 02  37          sty SLOTMAP
FAF9     38      ;
FAF9  A9 E1     39          lda /PAGEE1
FAFB     40      ;
FAFB  84 2E     41          sty DSTPTR
FAFD  85 2F     42          sta DSTPTR+1
FAFF     43      ;
FAFF  A9 C1     44          lda /PAGEC1
FB01     45      ;
FB01  84 2A     46          sty SRCPTR
FB03  85 2B     47          sta SRCPTR+1
FB05     48      ;
FB05  A0 B8     49          ldy #EPOFF          ; slot interface routine entry
FB07     50      ;
FB07  8C A8 02  51          sty SLOTJMP
FB0A  8D A9 02  52          sta SLOTJMP+1
FB0D     53      ;
FB0D     54          .if HWCARD
FB0D     55          lda #EPOFFVAL
FB0D     56          .el
FB0D  A9 10     57          lda #QLOFFVAL
FB0F     58          .fi
FB0F     59      ;
FB0F  4C 10 01  60          jmp MAPPAGE

```



```

FB12      61 ;
FB12      62 ;
FB12      63 ; Mapping code for all EPROM cards. The slots that contain
FB12      64 ; an EPROM card are recorded in SLOTMAP with its slot bit
FB12      65 ; set to 1.
FB12      66 ;
FB12      67 MAPCODE:
FB12      68 ;
FB12      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 3E FD     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
FB5C      162 ;
FB5C      163 ;
FB5C      164 ; Use the slot value in X-reg to prepare the slot*16 value.
FB5C      165 ;
FB5C 8A      166 MOVEEPBM txa
FB5D      167 ;
FB5D 0A      168      asl
FB5E 0A      169      asl
FB5F 0A      170      asl
FB60 0A      171      asl
FB61      172 ;
FB61 09 80 173      ora #EPSELC
FB63 AA      174      tax
FB64      175 ;
FB64      176 ;
FB64      177 ; Move the EPROM card bank management code to the stack.
FB64      178 ;
FB64 A0 79 179      ldy #EPBMLEN-1
FB66      180 ;
FB66 B9 7F FB 181 ^1      lda EPBMCODE,Y
FB69 99 10 01 182      sta EPBMPAGE,Y

```

```

FB6C      183 ;
FB6C 88    184      dey
FB6D 10 F7 185      bpl <1
FB6F      186 ;
FB6F      187 ;
FB6F      188 ; Modify the stack code for direct EPROM card access.
FB6F      189 ;
FB6F 8E 23 01 190      stx EPBMMOD1+1
FB72 8E 4F 01 191      stx EPBMMOD2+1
FB75 8E 5E 01 192      stx EPBMMOD3+1
FB78      193 ;
FB78 8E 69 01 194      stx EXECMOD1+1
FB7B 8E 74 01 195      stx EXECMOD2+1
FB7E      196 ;
FB7E 60     197      rts
FB7F      198 ;
FB7F      199 ;
FB7F      200 ; EPROM card bank management code.
FB7F      201 ;
FB7F      202 EPBMCODE:
FB7F      203 ;
FB7F      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      213      ldx EPBANK
0110      214      lda EPNMBR
0110      215      rts
0110      216      dfs 6,NEGONE
0110      217      .el
0110 AD 95 02 218      lda EPBANK
0113 4A      219      lsr
0114 AA      220      tax
0115 AD 94 02 221      lda EPNMBR
0118 90 02    222      bcc >1
011A 09 08    223      ora #EPUSR
011C 60      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           270 EPJSR    lda #EPOFFVAL
014C           271      .el
014C A9 10      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      317 ldx #EPOFFVAL
0171      318 .el
0171 A2 10 319 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
FBF9      343 ;
FBF9      344 ;
FBF9      345 ; Internal DCB structures.
FBF9      346 ;
FBF9      347 ;
FBF9      348 ; Load Lisa80 DCB
FBF9      349 ;
FBF9      350 LLDCB:
FBF9      351 ;
FBF9 02 352 LLCMD byt RUNCMD ; run
FBFA 70 353 LLEPROM byt SRCHALL ; search all EPROMs
FBFB 00 00 354 LLALTADR adr *-* ; no alternate address
FBFD FF 355 LLSTAT byt NEGONE ; return status
FBFE 0A 356 LLNAMLEN byt LLNLEN ; length of filename
FBFF D8 02 357 LLNAMADR adr LLNADR ; address of filename
FC01 CC CF C1 358 LLNAME asc "LOADLISA80"
FC04 C4 CC C9
FC07 D3 C1 B8
FC0A B0
FC0B      359 ;
000A      360 LLNLEN equ *-LLNAME
0012      361 LLDLEN equ *-LLDCB
02D8      362 LLNADR equ DCBBUFR+LLNAME-LLDCB

```

```

FC0B      363 ;
FC0B      364 ;
FC0B      365 ; RamDisk Config DCB
FC0B      366 ;
FC0B      367 RDDCB:
FC0B      368 ;
FC0B 02    369 RDCMD      byt  RUNCMD          ; run
FC0C 70    370 RDEPROM    byt  SRCHALL         ; search all EPROMs
FC0D 00 00 371 RDALTADR   adr  *-*           ; no alternate address
FC0F FF    372 RDSTAT     byt  NEGONE            ; return status
FC10 0E    373 RDNAMLEN   byt  RDNLEN         ; length of filename
FC11 D8 02 374 RDNAMADR   adr  RDNADR        ; address of filename
FC13 D2 E1 ED 375 RDNNAME   asc  "RamDisk Config"
FC16 C4 E9 F3
FC19 EB A0 C3
FC1C EF EE E6
FC1F E9 E7
FC21      376 ;
000E      377 RDNLEN     equ  *-RDNNAME
0016      378 RDDLEN     equ  *-RDDCB
02D8      379 RDNADR     equ  DCBBUFR+RDNNAME-RDDCB
FC21      380 ;
FC21      381 ;
FC21      382 ; FID DCB
FC21      383 ;
FC21      384 FDDCB:
FC21      385 ;
FC21 02    386 FDCMD      byt  RUNCMD          ; run
FC22 70    387 FDEPROM    byt  SRCHALL         ; search all EPROMs
FC23 00 00 388 FDALTADR   adr  *-*           ; no alternate address
FC25 FF    389 FDSTAT     byt  NEGONE            ; return status
FC26 03    390 FDNAMLEN   byt  FDNLEN         ; length of filename
FC27 D8 02 391 FDNAMADR   adr  FDNADR        ; address of filename
FC29 C6 C9 C4 392 FDNAME     asc  "FID"
FC2C      393 ;
0003      394 FDNLEN     equ  *-FDNAME
000B      395 FDDLLEN     equ  *-FDDCB
02D8      396 FDNADR     equ  DCBBUFR+FDNAME-FDDCB
FC2C      397 ;
FC2C      398 ;
FC2C      399 ; ADT2 DCB
FC2C      400 ;
FC2C      401 ADDCB:
FC2C      402 ;
FC2C 02    403 ADCMD      byt  RUNCMD          ; run
FC2D 70    404 ADEPROM    byt  SRCHALL         ; search all EPROMs
FC2E 00 00 405 ADALTADR   adr  *-*           ; no alternate address
FC30 FF    406 ADSTAT     byt  NEGONE            ; return status
FC31 04    407 ADNAMLEN   byt  ADNLEN         ; length of filename
FC32 D8 02 408 ADNAMADR   adr  ADNADR        ; address of filename
FC34 C1 C4 D4 409 ADNAME     asc  "ADT2"
FC37 B2
FC38      410 ;
0004      411 ADNLEN     equ  *-ADNAME
000C      412 ADDLEN     equ  *-ADDCB
02D8      413 ADNADR     equ  DCBBUFR+ADNAME-ADDCB
FC38      414 ;
FC38      415 ;
FC38      416 ; Set Clock DCB
FC38      417 ;
FC38      418 SCDCB:

```

```

FC38      419 ;
FC38 02    420 SCCMD      byt  RUNCMD          ; run
FC39 70    421 SCEPROM    byt  SRCHALL         ; search all EPROMs
FC3A 00 00 422 SCALTADR   adr  *-*           ; no alternate address
FC3C FF    423 SCSTAT     byt  NEGONE          ; return status
FC3D 09    424 SCNAMLEN   byt  SCNLEN          ; length of filename
FC3E D8 02 425 SCNAMADR   adr  SCNADR         ; address of filename
FC40 D3 E5 F4 426 SCNAME    asc  "Set Clock"
FC43 A0 C3 EC
FC46 EF E3 EB
FC49      427 ;
0009      428 SCNLEN      equ  *-SCNAME
0011      429 SCDLEN      equ  *-SCDCB
02D8      430 SCNADR      equ  DCBBUFR+SCNAME-SCDCB
FC49      431 ;
FC49      432 ;
FC49      433 ; Applesoft Formatter DCB
FC49      434 ;
FC49      435 AFDCB:
FC49      436 ;
FC49 02    437 AFCMD      byt  RUNCMD          ; run
FC4A 70    438 AFEPROM    byt  SRCHALL         ; search all EPROMs
FC4B 00 00 439 AFALTADR   adr  *-*           ; no alternate address
FC4D FF    440 AFSTAT     byt  NEGONE          ; return status
FC4E 13    441 AFNAMLEN   byt  AFNLEN          ; length of filename
FC4F D8 02 442 AFNAMADR   adr  AFNADR         ; address of filename
FC51 C1 F0 F0 443 AFNAME    asc  "Applesoft Formatter"
FC54 EC E5 F3
FC57 EF E6 F4
FC5A A0 C6 EF
FC5D F2 ED E1
FC60 F4 F4 E5
FC63 F2
FC64      444 ;
0013      445 AFNLEN      equ  *-AFNAME
001B      446 AFDLEN      equ  *-AFDCB
02D8      447 AFNADR      equ  DCBBUFR+AFNAME-AFDCB
FC64      448 ;
FC64      449 ;
FC64      450 ; Load SOURCEROR DCB
FC64      451 ;
FC64      452 LSDCB:
FC64      453 ;
FC64 02    454 LSCMD      byt  RUNCMD          ; run
FC65 70    455 LSEPROM    byt  SRCHALL         ; search all EPROMs
FC66 00 00 456 LSALTADR   adr  *-*           ; no alternate address
FC68 FF    457 LSSTAT     byt  NEGONE          ; return status
FC69 0E    458 LSNAMLEN   byt  LSNLEN          ; length of filename
FC6A D8 02 459 LSNAMADR   adr  LSNADR         ; address of filename
FC6C CC EF E1 460 LSNAME    asc  "Load Sourceror"
FC6F E4 A0 D3
FC72 EF F5 F2
FC75 E3 E5 F2
FC78 EF F2
FC7A      461 ;
000E      462 LSNLEN      equ  *-LSNAME
0016      463 LSDLEN      equ  *-LSDCB
02D8      464 LSNADR      equ  DCBBUFR+LSNAME-LSDCB
FC7A      465 ;
FC7A      466 ;
FC7A      467 ; EPROM Burner DCB

```

```
FC7A      468 ;
FC7A      469 EBD CB:
FC7A      470 ;
FC7A 02    471 EBCMD      byt  RUNCMD      ; run
FC7B 70    472 EBEPROM     byt  SRCHALL      ; search all EPROMs
FC7C 00 00 473 EBALTADR   adr  *-*        ; no alternate address
FC7E FF    474 EBSTAT      byt  NEGONE       ; return status
FC7F 0C    475 EBNAMLEN    byt  EBNLEN      ; length of filename
FC80 D8 02 476 EBNAMADR   adr  EBNADR      ; address of filename
FC82 C5 D0 D2 477 EBNAME      asc  "EPROM Burner"
FC85 CF CD A0
FC88 C2 F5 F2
FC8B EE E5 F2
FC8E      478 ;
000C      479 EBNLEN      equ  *-EBNAME
0014      480 EBDLEN      equ  *-EBDCB
02D8      481 EBNADR      equ  DCBBUFR+EBNAME-EBDCB
FC8E      482 ;
FC8E      483 ;
FC8E      484 ; VTOC Manager DCB
FC8E      485 ;
FC8E      486 VTDCB:
FC8E      487 ;
FC8E 02    488 VTCMD      byt  RUNCMD      ; run
FC8F 70    489 VTEPROM     byt  SRCHALL      ; search all EPROMs
FC90 00 00 490 VTALTADR   adr  *-*        ; no alternate address
FC92 FF    491 VTSTAT      byt  NEGONE       ; return status
FC93 0C    492 VTNAMLEN    byt  VTNLEN      ; length of filename
FC94 D8 02 493 VTNAMADR   adr  VTNADR      ; address of filename
FC96 D6 D4 CF 494 VTNAME      asc  "VTOC Manager"
FC99 C3 A0 CD
FC9C E1 EE E1
FC9F E7 E5 F2
FCA2      495 ;
000C      496 VTNLEN      equ  *-VTNAME
0014      497 VTDLEN      equ  *-VTDCB
02D8      498 VTNADR      equ  DCBBUFR+VTNAME-VTDCB
FCA2      499 ;
FCA2      500 ;
FCA2      501 ; Volume Manager DCB
FCA2      502 ;
FCA2      503 VODCB:
FCA2      504 ;
FCA2 02    505 VOCMD      byt  RUNCMD      ; run
FCA3 70    506 VOEPROM     byt  SRCHALL      ; search all EPROMs
FCA4 00 00 507 VOALTADR   adr  *-*        ; no alternate address
FCA6 FF    508 VOSTAT      byt  NEGONE       ; return status
FCA7 0E    509 VONAMLEN    byt  VONLEN      ; length of filename
FCA8 D8 02 510 VONAMADR   adr  VONADR      ; address of filename
FCAA D6 EF EC 511 VONAME      asc  "Volume Manager"
FCAD F5 ED E5
FCB0 A0 CD E1
FCB3 EE E1 E7
FCB6 E5 F2
FCB8      512 ;
000E      513 VONLEN      equ  *-VONAME
0016      514 VODLEN      equ  *-VODCB
02D8      515 VONADR      equ  DCBBUFR+VONAME-VODCB
FCB8      516 ;
FCB8      517 ;
FCB8      518 ; Volume Duplicate DCB
```



```

FCB8      519 ;
FCB8      520 VDDCB:
FCB8      521 ;
FCB8 02    522 VDCMD      byt  RUNCMD          ; run
FCB9 70    523 VDEPROM    byt  SRCHALL         ; search all EPROMs
FCBA 00 00 524 VDALTADR    adr  *-*           ; no alternate address
FCBC FF    525 VDSTAT      byt  NEGONE          ; return status
FCBD 10    526 VDNAMLEN    byt  VDNLEN         ; length of filename
FCBE D8 02 527 VDNAMADR    adr  VDNADR         ; address of filename
FCC0 D6 EF EC 528 VDNAME      asc  "Volume Duplicate"
FCC3 F5 ED E5
FCC6 A0 C4 F5
FCC9 F0 EC E9
FCCC E3 E1 F4
FCCF E5
FCD0      529 ;
0010      530 VDNLEN      equ  *-VDNAME
0018      531 VDDLLEN     equ  *-VDDCB
02D8      532 VDNADR      equ  DCBBUFR+VDNAME-VDDCB
FCD0      533 ;
FCD0      534 ;
FCD0      535 ; Disk Window DCB
FCD0      536 ;
FCD0      537 DWDCB:
FCD0      538 ;
FCD0 02    539 DWCMD      byt  RUNCMD          ; run
FCD1 70    540 DWEPROM    byt  SRCHALL         ; search all EPROMs
FCD2 00 00 541 DWALTADR    adr  *-*           ; no alternate address
FCD4 FF    542 DWSTAT      byt  NEGONE          ; return status
FCD5 0B    543 DWNAMLEN    byt  DWNLEN         ; length of filename
FCD6 D8 02 544 DWNAMADR    adr  DWNADR         ; address of filename
FCD8 C4 E9 F3 545 DWNAME      asc  "Disk Window"
FCDB EB A0 D7
FCDE E9 EE E4
FCE1 EF F7
FCE3      546 ;
000B      547 DWNLEN      equ  *-DWNAME
0013      548 DWDLEN      equ  *-DWDCB
02D8      549 DWNADR      equ  DCBBUFR+DWNAME-DWDCB
FCE3      550 ;
FCE3      551 ;
FCE3      552 ; Binary File Install DCB
FCE3      553 ;
FCE3      554 BFDCB:
FCE3      555 ;
FCE3 02    556 BFCMD      byt  RUNCMD          ; run
FCE4 70    557 BFEPPROM    byt  SRCHALL         ; search all EPROMs
FCE5 00 00 558 BFALTADR    adr  *-*           ; no alternate address
FCE7 FF    559 BFSTAT      byt  NEGONE          ; return status
FCE8 13    560 BFNAMLEN    byt  BFNLEN         ; length of filename
FCE9 D8 02 561 BFNAMADR    adr  BFNADR         ; address of filename
FCEB C2 E9 EE 562 BFNAME      asc  "Binary File Install"
FCEE E1 F2 F9
FCF1 A0 C6 E9
FCF4 EC E5 A0
FCF7 C9 EE F3
FCFA F4 E1 EC
FCFD EC
FCFE      563 ;
0013      564 BFNLEN      equ  *-BFNAME
001B      565 BFDLEN      equ  *-BFDCB

```

```

02D8          566  BFNADR      equ  DCBBUFR+BFNAME-BFDCB
FCFE          567  ;
FCFE          568  ;
FCFE          569  ; BigMac DCB
FCFE          570  ;
FCFE          571  BMDCB:
FCFE          572  ;
FCFE 02       573  BMCMD      byt  RUNCMD          ; run
FCFF 70       574  BMEPROM    byt  SRCHALL         ; search all EPROMs
FD00 00 00    575  BMALTADR   adr  *-*           ; no alternate address
FD02 FF       576  BMSTAT     byt  NEGONE          ; return status
FD03 0B       577  BMNAMLEN   byt  BMNLEN          ; length of filename
FD04 D8 02    578  BMNAMADR   adr  BMNADR          ; address of filename
FD06 CC EF E1 579  BMNAME     asc  "Load BigMac"
FD09 E4 A0 C2
FD0C E9 E7 CD
FD0F E1 E3
FD11          580  ;
000B          581  BMNLEN     equ  *-BMNAME
0013          582  BMDLEN     equ  *-BMDCB
02D8          583  BMNADR     equ  DCBBUFR+BMNAME-BMDCB
FD11          584  ;
FD11          585  ;
FD11          586  ; Scan Disk DCB
FD11          587  ;
FD11          588  SDDCB:
FD11          589  ;
FD11 02       590  SDCMD      byt  RUNCMD          ; run
FD12 70       591  SDEPROM    byt  SRCHALL         ; search all EPROMs
FD13 00 00    592  SDALTADR   adr  *-*           ; no alternate address
FD15 FF       593  SDSTAT     byt  NEGONE          ; return status
FD16 04       594  SDNAMLEN   byt  SDNLEN          ; length of filename
FD17 D8 02    595  SDNAMADR   adr  SDNADR          ; address of filename
FD19 D3 E3 E1 596  SDNAME     asc  "Scan"
FD1C EE
FD1D          597  ;
0004          598  SDNLEN     equ  *-SDNAME
000C          599  SDDLLEN    equ  *-SDDCB
02D8          600  SDNADR     equ  DCBBUFR+SDNAME-SDDCB
FD1D          601  ;
FD1D          602  ;
FD1D          603          icl  "TABLES.L"

```

```

LLOAD TABLES.L,A$4000

```

```

FD1D      1          ttl "EOS+ Source Code, TABLES.L"
FD1D      2      ;
FD1D      3      ;
FD1D      4      ; TABLES.L
FD1D      5      ;
FD1D      6      ;
FD1D      7  CMDTBL:
FD1D EB EB      8      adr AHNDLR-1
FD1F F2 EB      9      adr BHNDLR-1
FD21 08 EC     10      adr CHNDLR-1
FD23 12 EC     11      adr DHNDLR-1
FD25 1C EC     12      adr EHNDLR-1
FD27 27 EC     13      adr FHNDLR-1
FD29 2A EC     14      adr GHNDLR-1
FD2B 39 EC     15      adr HHNDLR-1
FD2D 42 EC     16      adr IHNDLR-1
FD2F 4B EC     17      adr JHNDLR-1
FD31 54 EC     18      adr KHNDLR-1
FD33 5D EC     19      adr LHNDLR-1
FD35 66 EC     20      adr MHNDLR-1
FD37 6F EC     21      adr NHNDLR-1
FD39 75 EC     22      adr OHNDLR-1
FD3B 78 EC     23      adr PHNDLR-1
FD3D A4 EC     24      adr QHNDLR-1
FD3F AD EC     25      adr RHNDLR-1
FD41 B6 EC     26      adr SHNDLR-1
FD43 BF EC     27      adr THNDLR-1
FD45 C8 EC     28      adr UHNDLR-1
FD47 D1 EC     29      adr VHNDLR-1
FD49 DA EC     30      adr WHNDLR-1
FD4B E3 EC     31      adr XHNDLR-1
FD4D EC EC     32      adr YHNDLR-1
FD4F F5 EC     33      adr ZHNDLR-1
FD51      34      ;
FD51      35      ;
FD51      36  ADDRTBLL:
FD51 ED      37      byt COUT          ; X = 0
FD52 84      38      byt SETNORM       ; X = 1
FD53 2F      39      byt INIT          ; X = 2
FD54 93      40      byt SETVID        ; X = 3
FD55 89      41      byt SETKBD        ; X = 4
FD56 58      42      byt HOME          ; X = 5
FD57 22      43      byt VTAB          ; X = 6
FD58 9C      44      byt CLREOL        ; X = 7
FD59 42      45      byt CLREOP        ; X = 8
FD5A EA      46      byt HOOKDOS       ; X = 9
FD5B 41      47      byt PRNTAX        ; X = 10
FD5C DA      48      byt PRBYTE        ; X = 11
FD5D E3      49      byt PRHEX         ; X = 12
FD5E      50      ;
FD5E      51  ADDRTBLH:
FD5E FD      52      hby COUT
FD5F FE      53      hby SETNORM
FD60 FB      54      hby INIT
FD61 FE      55      hby SETVID
FD62 FE      56      hby SETKBD
FD63 FC      57      hby HOME
FD64 FC      58      hby VTAB
FD65 FC      59      hby CLREOL
FD66 FC      60      hby CLREOP

```

```

FD67 03          61          hby HOOKDOS
FD68 F9          62          hby PRNTAX
FD69 FD          63          hby PRBYTE
FD6A FD          64          hby PRHEX
FD6B            65          ;
FD6B            66          ;
FD6B            67          LINETBL:
FD6B 03 05 07    68          hex 030507090A0C
FD6E 09 0A 0C
FD71 0D 0E 0F    69          hex 0D0E0F101112
FD74 10 11 12
FD77            70          ;
FD77            71          ;
FD77 01 0A 64    72          DECTBLL byt 1,10,100
FD7A            73          ;
FD7A            74          ;
FD7A            75          MAPMASKS:
FD7A 01 02 04    76          hex 01020408
FD7D 08
FD7E 10 20 40    77          hex 10204080
FD81 80
FD82            78          ;
FD82            79          ;
FD82            80          ZCDEFLT:
FD82            81          ;          7654321S
FD82 E6          82          byt %11100110          ; slots
FD83            83          ;
FD83 00          84          hex 00          ; speed (from ZCSPDTBL)
FD84            85          ;
FD84            86          ;
FD84 00 08 40    87          ZCSPDTBL hex 000840C8
FD87 C8
FD88            88          ;
FD88            89          ;
FD88 D4 C1 C2    90          PARMTYPE asc "TABBBRSP"
FD8B C2 C2 D2
FD8E D3 D0
FD90            91          ;
FD90            92          ;
FD90            93          LOADTBL:
FD90 1A F1        94          adr LOADTEXT-1
FD92 BE F1        95          adr EXECAS-1
FD94 FD F1        96          adr EXECBIN0-1
FD96 EE F1        97          adr EXECBIN1-1
FD98 F7 F1        98          adr EXECBIN2-1
FD9A            99          ;
FD9A           100          ;
FD9A           101          RUNTBL:
FD9A 39 ED        102          adr EXECTEXT-1
FD9C BE F1        103          adr EXECAS-1
FD9E FD F1        104          adr EXECBIN0-1
FDA0 EE F1        105          adr EXECBIN1-1
FDA2 F7 F1        106          adr EXECBIN2-1
FDA4           107          ;
FDA4           108          ;
FDA4           109          TYPETBL:
FDA4 00           110          byt TYPETEXT-TYPTTEXTS
FDA5 05           111          byt TYPEAS-TYPTTEXTS
FDA6 0F           112          byt TYPEBIN0-TYPTTEXTS
FDA7 23           113          byt TYPEBIN1-TYPTTEXTS
FDA8 35           114          byt TYPEBIN2-TYPTTEXTS

```

```

FDA9 47          115          byt TYPERESD-TYPTEXTS
FDAA 50          116          byt TYPESYS-TYPTEXTS
FDAB 57          117          byt TYPEPRI-TYPTEXTS
FDAC             118          ;
FDAC             119          ;
FDAC             120          TYPTEXTS:
FDAC             121          ;
FDAC D4 E5 F8    122          TYPETEXT asc "Text"
FDAF F4
FDB0 00          123          hex 00
FDB1             124          ;
FDB1 C1 F0 F0    125          TYPEAS  asc "Applesoft"
FDB4 EC E5 F3
FDB7 EF E6 F4
FDBA 00          126          hex 00
FDBB             127          ;
FDBB C2 E9 EE    128          TYPEBIN0 asc "Binary, main memory"
FDBE E1 F2 F9
FDC1 AC A0 ED
FDC4 E1 E9 EE
FDC7 A0 ED E5
FDCA ED EF F2
FDCD F9
FDCE 00          129          hex 00
FDCF             130          ;
FDCF C2 E9 EE    131          TYPEBIN1 asc "Binary, LC Bank 1"
FDD2 E1 F2 F9
FDD5 AC A0 CC
FDD8 C3 A0 C2
FDDB E1 EE EB
FDDE A0 B1
FDE0 00          132          hex 00
FDE1             133          ;
FDE1 C2 E9 EE    134          TYPEBIN2 asc "Binary, LC Bank 2"
FDE4 E1 F2 F9
FDE7 AC A0 CC
FDEA C3 A0 C2
FDED E1 EE EB
FDF0 A0 B2
FDF2 00          135          hex 00
FDF3             136          ;
FDF3 D2 E5 F3    137          TYPERESD asc "Reserved"
FDF6 E5 F2 F6
FDF9 E5 E4
FDFB 00          138          hex 00
FDFC             139          ;
FDFC D3 F9 F3    140          TYPESYS  asc "System"
FDFF F4 E5 ED
FE02 00          141          hex 00
FE03             142          ;
FE03 D0 F2 E9    143          TYPEPRI  asc "Primary"
FE06 ED E1 F2
FE09 F9
FE0A 00          144          hex 00
FE0B             145          ;
FE0B             146          ;
FE0B             147          ; Addresses of text strings that are issued from the EOS
FE0B             148          ; Main Selection Menu in order to support the DOS CATALOG
FE0B             149          ; and the RUN HELLO commands.
FE0B             150          ;
FE0B             151          TEXTTBLL:

```

```

FE0B 15          152          byt CATTEXT          ; X = 0
FE0C 1F          153          byt HELOTEXT          ; X = 1
FE0D 2B          154          byt SLTTEXT           ; X = 2
FE0E 2E          155          byt DRVTEXT           ; X = 3
FE0F 32          156          byt VOLTEXT           ; X = 4
FE10             157          ;
FE10             158          TEXTTBLH:
FE10 FE          159          hby CATTEXT
FE11 FE          160          hby HELOTEXT
FE12 FE          161          hby SLTTEXT
FE13 FE          162          hby DRVTEXT
FE14 FE          163          hby VOLTEXT
FE15             164          ;
FE15             165          ;
FE15 8D 84       166          CATTEXT byt RETURN,CTRLD
FE17 C3 C1 D4    167          asc "CATALOG"
FE1A C1 CC CF
FE1D C7
FE1E 00          168          byt ZERO
FE1F             169          ;
FE1F 8D 84       170          HELOTEXT byt RETURN,CTRLD
FE21 D2 D5 CE    171          asc "RUN HELLO"
FE24 A0 C8 C5
FE27 CC CC CF
FE2A 00          172          byt ZERO
FE2B             173          ;
FE2B AC D3       174          SLTTEXT asc ",S"
FE2D 00          175          byt ZERO
FE2E             176          ;
FE2E AC C4 A4    177          DRVTEXT asc ",D$"
FE31 00          178          byt ZERO
FE32             179          ;
FE32 AC D6 A4    180          VOLTEXT asc ",V$"
FE35 00          181          byt ZERO
FE36             182          ;
FE36             183          ;
FE36             184          ; Text used for discovering EPROM cards in slots.
FE36             185          ;
FE36 C5 D0 C2    186          EPTEXT asc "EPBINEOS"          ; EPBIN text
FE39 C9 CE C5
FE3C CF D3
FE3E             187          ;
FE3E             188          ;
FE3E             189          ; Text strings used in the ZipChip display routine.
FE3E             190          ;
FE3E CF EE 20    191          ZCTEXT dci "On "
FE41 CF E6 66    192          ZCTEXT1 dci "Off"
FE44 B8 AE B0    193          ZCTEXT2 dci "8.00"
FE47 30
FE48 B6 AE B0    194          dci "6.00"
FE4B 30
FE4C B4 AE B0    195          dci "4.00"
FE4F 30
FE50 B2 AE B0    196          dci "2.00"
FE53 30
FE54 C5 EE E1    197          ZCTEXT3 dci "Enabled "
FE57 E2 EC E5
FE5A E4 20
FE5C C4 E9 F3    198          ZCTEXT4 dci "Disabled"
FE5F E1 E2 EC
FE62 E5 64

```

```
FE64 C6 E1 F3    199  ZCTEXT5  dci "Fast  "
FE67 F4 A0 20
FE6A CE EF F2    200  ZCTEXT6  dci "Normal"
FE6D ED E1 6C
FE70              201  ;
FE70              202  ;
FE70 C4 B8 90    203  SYNCBYTS byt SYNCBYT0,SYNCBYT1,SYNCBYT2,SYNCBYT3
FE73 ED
FE74              204  ;
FE74              205  ;
FE74              206  ; EOS code end.
FE74              207  ;
FE74              208  ;
00FA              209  EOSEND    equ EOSVCTRS&NEGONE
FE74              210  ;
FE74              211  ;
FE74              212              .if DEBUG
FE74              213              dfs EOSEND-*&NEGONE
FE74              214              .el
FE74              215              dfs EOSVCTRS-*,NEGONE
FFFA              216              .fi
FFFA              217  ;
FFFA              218  ;
FFFA 00 E8        219  NMIRTN    adr EOS
FFFC 00 E8        220  RESET     adr EOS
FFFE 00 E8        221  IRQRTN    adr EOS
0000              222  ;
0000              223  ;
```

BSAVE SEG01,D1,A\$0800,B,L\$2000

```
0000              224              usr SEG01,D1
0000              225  ;
0000              226  ;
0000              227              icl "BANKS.L,D2"
```

LLOAD BANKS.L,D2,A\$4000

```
0000          1          ttl "EOS+ Source Code, BANKS.L"
0000          2          ;
0000          3          ;
0000          4          ; BANKS.L
0000          5          ;
0000          6          ;
CD D1
```

```
0000          7          dcm "CD D1"
0000          8          ;
0000          9          ;
0000         10          ; First half of EPROM.
0000         11          ;
0000         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 0x1EF4

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	0000
HWCARD	0000	SRCHALL	0070	CATALOG	E000	DOSLPRMS	E004	DOSHPRMS	E015
LISAPRMS	E026	LISA1PRM	E037	LISA2PRM	E046	LISA3PRM	E055	SETUPRMS	E064
RMDSKPRM	E072	FIDPARMS	E087	CLKPARMS	E091	LISTPRMS	E0A1	ROMPARMS	E0BB
CATPARMS	E0CA	HWSLOT	0007	HWSLOT16	0070	HWSLOT16	00C7	LABEL	0000
EPASEOS	C400	ASEXIT	C450	BINEXIT	C480	RTNEXIT	C4A8	EPOFF	C4B8
EPUSER1	C4C0	EPUSER2	C4C8	EPMAPEOS	C4D0	EPBINEOS	C4E0	EPEOS	C4F0
EPBINTXT	C4F8	EOS	E800	MAPEOS	E8AD	USERRTN1	E912	MAIN	E92F
MAIN2	EB46	MAIN3	EB49	SELC	EB6F	SELCERR	EBD7	AHNDLR	EBEC
BHNDLR	EBF3	BHNDLR2	EBF8	CHNDLR	EC09	USERRTN2	EC10	DHNDLR	EC13
EHNDLR	EC1D	FHNDLR	EC28	GHNDLR	EC2B	HHNDLR	EC3A	IHNDLR	EC43
JHNDLR	EC4C	KHNDLR	EC55	LHNDLR	EC5E	MHNDLR	EC67	NHNDLR	EC70
OHNDLR	EC76	PHNDLR	EC79	QHNDLR	ECA5	RHNDLR	ECAE	SHNDLR	ECB7
THNDLR	ECC0	UHNDLR	ECC9	VHNDLR	ECD2	WHNDLR	ECDB	XHNDLR	ECE4
YHNDLR	ECED	ZHNDLR	ECF6	DOEOSDCB	ED04	EXITAS	ED1B	EXITBIN	ED1E
EXITRTN	ED21	EXECTEXT	ED3A	DOEXEC	ED4B	DOEXEC2	ED59	EOSCAT	ED63
EOSCAT2	ED66	CATHDR	EE13	CATFTR	EE57	CHKCAT	EEAE	SHOWCAT	EED8
SELCFIL	EF34	SELCFIL2	EF59	CKCATCMD	EFCA	GETFILE2	F004	GETFILE	F00B
SHOWFILE	F01E	RUNLOAD	F0E9	RUNFILE	F109	LOADFILE	F112	LOADTEXT	F11B
EXECAS	F1BF	EXECBIN1	F1EF	EXECBIN2	F1F8	EXECBIN0	F1FE	ASEOS	F210
DOASFILE	F285	DOASCAT	F2AD	SAVFILE	F2D1	RTNCLC	F35E	GETEPRNG	F360
GETRANGE	F367	SAVPARM2	F38E	SAVPARM	F390	GETASVAL	F398	SETASPTR	F3A3
BINEOS	F3B0	BINEOS2	F3CC	BINDONE	F3EF	BINLOAD	F408	BINRUN	F409
BINCAT	F43F	DOZCOFF	F49F	SETANNUN	F4CE	DOZCON	F4DB	DOZCRSET	F4F6
DOZCREAD	F510	DOZCSAVE	F53E	DOZCOPEN	F577	ZCCONFIG	F597	ZCLOOP	F675
ZCDISP	F6D1	EOSBELL	F72C	RDKEY	F73C	GETKEY	F743	GETHEX	F759
GETNUM	F778	GETVAL	F7AD	EDITSDV	F7FA	PRTSDV	F862	PRTDEC	F880
RTN01	F899	NEXTMAP	F89A	CLRUSER	F8B2	SETUSER	F8B4	PRINT	F8C2
PRINT01	F8E8	DOVTAB	F91A	DOSHOCK	F91E	DOPRNTAX	F922	DOPRBYTE	F928
DOPRHEX	F92C	DOSPACE	F930	DOCROUT	F933	DOCOUT	F935	DOJSRMEM	F939
JSRMEM	F943	LOADOSL	F94C	LOADOSH	F967	LOADOSH2	F9A5	COPYROM	F9AC
INITCAT	F9CC	GETENTRY	F9E5	FINDFILE	FA40	COPYPRM0	FA7F	COPYPRM	FA84
SELCBANK	FA92	READBLK	FAAC	READADR	FAC6	BUILDMAP	FAE1	MAPCODE	FB12
MAPPAGE	0110	EPOFFJMP	0157	EPMAPLEN	004A	MOVEEPPM	FB5C	EPBMCODE	FB7F
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	FBF9
LLCMD	FBF9	LLEPROM	FBFA	LLALTADR	FBFB	LLSTAT	FBFD	LLNAMLEN	FBFE
LLNAMADR	FBFF	LLNAME	FC01	LLNLEN	000A	LLDLEN	0012	LLNADR	02D8
RDDCB	FC0B	RDCMD	FC0B	RDEPROM	FC0C	RDALTADR	FC0D	RDSTAT	FC0F
RDNAMLEN	FC10	RDNAMADR	FC11	RDNAME	FC13	RDNLEN	000E	RDDLEN	0016
RDNADR	02D8	FDDCB	FC21	FDCMD	FC21	FDEPROM	FC22	FDALTADR	FC23
FDSTAT	FC25	FDNAMLEN	FC26	FDNAMADR	FC27	FDNAME	FC29	FDNLEN	0003
FDDLLEN	000B	FDNADR	02D8	ADDCB	FC2C	ADCMD	FC2C	ADEPROM	FC2D
ADALTADR	FC2E	ADSTAT	FC30	ADNAMLEN	FC31	ADNAMADR	FC32	ADNAME	FC34
ADNLEN	0004	ADDLEN	000C	ADNADR	02D8	SCDCB	FC38	SCCMD	FC38
SCEPROM	FC39	SCALTADR	FC3A	SCSTAT	FC3C	SCNAMLEN	FC3D	SCNAMADR	FC3E
SCNAME	FC40	SCNLEN	0009	SCDLEN	0011	SCNADR	02D8	AFDCB	FC49
AFCMD	FC49	AFEPROM	FC4A	AFALTADR	FC4B	AFSTAT	FC4D	AFNAMLEN	FC4E
AFNAMADR	FC4F	AFNAME	FC51	AFNLEN	0013	AFDLEN	001B	AFNADR	02D8

LSDCB	FC64	LSCMD	FC64	LSEEPROM	FC65	LSALTADR	FC66	LSSTAT	FC68
LSNAMLEN	FC69	LSNAMADR	FC6A	LSNAME	FC6C	LSNLEN	000E	LSDLEN	0016
LSNADR	02D8	EBDCB	FC7A	EBCMD	FC7A	EBEPROM	FC7B	EBALTADR	FC7C
EBSTAT	FC7E	EBNAMLEN	FC7F	EBNAMADR	FC80	EBNAME	FC82	EBNLEN	000C
EBDLEN	0014	EBNADR	02D8	VTDCB	FC8E	VTCMD	FC8E	VTEPROM	FC8F
VTALTADR	FC90	VTSTAT	FC92	VTNAMLEN	FC93	VTNAMADR	FC94	VTNAME	FC96
VTNLEN	000C	VTDLLEN	0014	VTNADR	02D8	VODCB	FCA2	VOCMD	FCA2
VOEPROM	FCA3	VOALTADR	FCA4	VOSTAT	FCA6	VONAMLEN	FCA7	VONAMADR	FCA8
VONAME	FCAA	VONLEN	000E	VODLEN	0016	VONADR	02D8	VDDCB	FCB8
VDCMD	FCB8	VDEPROM	FCB9	VDALTADR	FCBA	VDSTAT	FCBC	VDNAMLEN	FCBD
VDNAMADR	FCBE	VDNAME	FCC0	VDNLEN	0010	VDDLEN	0018	VDNADR	02D8
DWDCB	FCD0	DWCMD	FCD0	DWEPPROM	FCD1	DWALTADR	FCD2	DWSTAT	FCD4
DWNAMLEN	FCD5	DWNAMADR	FCD6	DWNAME	FCD8	DWNLEN	000B	DWDLEN	0013
DWNADR	02D8	BFDCB	FCE3	BFCMD	FCE3	BFEPROM	FCE4	BFALTADR	FCE5
BFSTAT	FCE7	BFNAMLEN	FCE8	BFNAMADR	FCE9	BFNAME	FCEB	BFNLEN	0013
bfdlen	001B	BFNADR	02D8	BMDCB	FCFE	BMCMD	FCFE	BMEPROM	FCFF
BMALTADR	FD00	BMSTAT	FD02	BMNAMLEN	FD03	BMNAMADR	FD04	BMNAME	FD06
BMNLEN	000B	BMDLEN	0013	BMNADR	02D8	SDDCB	FD11	SDCMD	FD11
SDEPROM	FD12	SDALTADR	FD13	SDSTAT	FD15	SDNAMLEN	FD16	SDNAMADR	FD17
SDNAME	FD19	SDNLEN	0004	SDDLLEN	000C	SDNADR	02D8	CMDTBL	FD1D
ADDRTBLL	FD51	ADDRTBLLH	FD5E	LINETBL	FD6B	DECTBLL	FD77	MAPMASKS	FD7A
ZCDEFLT	FD82	ZCSPDTBL	FD84	PARMTYPE	FD88	LOADTBL	FD90	RUNTBL	FD9A
TYPETBL	FDA4	TYPTEXTS	FDAC	TYPETEXT	FDAC	TYPEAS	FDB1	TYPEBIN0	FDBB
TYPEBIN1	FDCF	TYPEBIN2	FDE1	TYPERSD	FDF3	TYPESYS	FDFC	TYPEPRI	FE03
TEXTTBL	FE0B	TEXTTBLH	FE10	CATTEXT	FE15	HELOTEXT	FE1F	SLTTEXT	FE2B
DRVTEXT	FE2E	VOLTEXT	FE32	EPTEXT	FE36	ZCTEXT	FE3E	ZCTEXT1	FE41
ZCTEXT2	FE44	ZCTEXT3	FE54	ZCTEXT4	FE5C	ZCTEXT5	FE64	ZCTEXT6	FE6A
SYNCBYS	FE70	EOSEND	00FA	NMIRTN	FFFA	RESET	FFFC	IRQRTN	FFFE
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	FC2E	ADCMD	FC2C	ADDCB	FC2C	ADDLEN	000C	ADDRBUFR	02AA
ADDRTBLLH	FD5E	ADDRTBLL	FD51	ADEPROM	FC2D	ADNADR	02D8	ADNAMADR	FC32
ADNAME	FC34	ADNAMLEN	FC31	ADNLEN	0004	ADSTAT	FC30	AFALTADR	FC4B
AFCMD	FC49	AFDCB	FC49	AFDLEN	001B	AFEPROM	FC4A	AFNADR	02D8
AFNAMADR	FC4F	AFNAME	FC51	AFNAMLEN	FC4E	AFNLEN	0013	AFSTAT	FC4D
AHNDLR	EBEC	ALTCHOFF	C00E	ANN1OFF	C058	ANN2OFF	C05A	ANN3ON	C05D
ANN4ON	C05F	APLSOFT	0002	APPLTYPE	0293	AREG	0008	ASAV	0018
ASEOS	F210	ASEXIT	C450	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	FCE5	BFCMD	FCE3
BFDCB	FCE3	bfdlen	001B	BFEPROM	FCE4	BFNADR	02D8	BFNAMADR	FCE9
BFNAME	FCEB	BFNAMLEN	FCE8	BFNLEN	0013	BFSTAT	FCE7	BHNDLR	EBF3
BHNDLR2	EBF8	BINADR	095C	BINARY0	0004	BINARY1	0008	BINARY2	0010
BINCAT	F43F	BINCMD	095E	BINDONE	F3EF	BINEOS	F3B0	BINEOS2	F3CC
BINEPN	095F	BINEXIT	C480	BINFADR	0964	BINFALT	0960	BINFLEN	0963
BINJMP	0959	BINLOAD	F408	BINRUN	F409	BINSTAT	0962	BMALTADR	FD00
BMCMD	FCFE	BMDCB	FCFE	BMDLEN	0013	BMEPROM	FCFF	BMNADR	02D8
BMNAMADR	FD04	BMNAME	FD06	BMNAMLEN	FD03	BMNLEN	000B	BMSTAT	FD02
BUILDMAP	FAE1	C3ROMOFF	C00A	C3ROMON	C00B	CATALOG	E000	CATCMD	0003
CATFTR	EE57	CATHDR	EE13	CATPARMS	E0CA	CATTEXT	FE15	CH	0024
CHKCAT	EEAE	CHKCOM	DEBE	CHNDLR	EC09	CHRADR	00B8	CHRGOT	00B7
CKCATCMD	EFCA	CLK.D	0900	CLK.L	064A	CLK.O	F22E	CLKPARMS	E091
CLREOL	FC9C	CLREOP	FC42	CLRKEY	C010	CLRROM	CFFF	CLRUSER	F8B2
CMDPTR	00EE	CMDTBL	FD1D	CNTRCMD	0059	COMMA	00AC	COPYPARM	FA84

COPYPRM0	FA7F	COPYROM	F9AC	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	0000	DECTBLL	FD77	DHNDLR	EC13
DOASCAT	F2AD	DOASFILE	F285	DOCOUT	F935	DOCROUT	F933	DOEOSDCB	ED04
DOEXEC	ED4B	DOEXEC2	ED59	DOJSRMEM	F939	DOLLAR	00A4	DOMEMJMP	0162
DOPRBYTE	F928	DOPRHEX	F92C	DOPRNTAX	F922	DOSCOLD	03D3	DOSH.D	BE00
DOSH.L	2A00	DOSH.O	4100	DOSHOK	F91E	DOSHPRMS	E015	DOSL.D	9F00
DOSL.L	2100	DOSL.O	2000	DOSLPRMS	E004	DOSPACE	F930	DOSWARM	03D0
DOVTAB	F91A	DOZCOFF	F49F	DOZCON	F4DB	DOZCOPEN	F577	DOZCREAD	F510
DOZCRSET	F4F6	DOZCSAVE	F53E	DRIVE	00EC	DRVTEXT	FE2E	DSCTMP	009D
DSTPTR	002E	DSTVAL	02B6	DWALTADR	FCD2	DWCMD	FCD0	DWDCB	FCD0
DWDLEN	0013	DWEPROM	FCD1	DWNADR	02D8	DWNAMADR	FCD6	DWNAME	FCD8
DWNAMLEN	FCD5	DWNLEN	000B	DWSTAT	FCD4	EBALTADR	FC7C	EBCMD	FC7A
EBDCB	FC7A	EBDLEN	0014	EBEPROM	FC7B	EBNADR	02D8	EBNAMADR	FC80
EBNAME	FC82	EBNAMLEN	FC7F	EBNLEN	000C	EBSTAT	FC7E	EDITSDV	F7FA
EHNDLR	EC1D	ENDCAT	0000	ENTRYLEN	0020	EOLCMD	0057	EOPCMD	0058
EOS	E800	EOSBELL	F72C	EOSCAT	ED63	EOSCAT2	ED66	EOSEND	00FA
EOSLDCB	095E	EOSVCTRS	FFFA	EPASEOS	C400	EPBANK	0295	EPBINEOS	C4E0
EPBINTXT	C4F8	EPBMCODE	FB7F	EPBMLEN	007A	EPBMMOD1	0122	EPBMMOD2	014E
EPBMMOD3	015D	EPBMPAGE	0110	EPCARD	0001	EPCONFIG	0110	EPEND	0299
EPEOS	C4F0	EPEXEC	0165	EPJSR	014C	EPMAPEOS	C4D0	EPMAPLEN	004A
EPMASK	000F	EPMOVE	011D	EPMOVE2	0120	EPNMBR	0294	EPOFF	C4B8
EPOFFJMP	0157	EPOFFVAL	0080	EPONVAL	0000	EPRETURN	015A	EPSEARCH	02A2
EPSELC	C080	EPSLOT	0291	EPSRCH	00F0	EPSTRT	0298	EPTEXT	FE36
EPUSER1	C4C0	EPUSER2	C4C8	EPUSR	0008	ERR00	0000	ERR01	0001
ERR02	0002	ERR03	0003	ERR04	0004	ERR05	0005	ESCAPE	009B
EXECAS	F1BF	EXECBIN0	F1FE	EXECBIN1	F1EF	EXECBIN2	F1F8	EXECMOD1	0168
EXECMOD2	0173	EXECPTR	00FA	EXECTEXT	ED3A	EXITAS	ED1B	EXITBIN	ED1E
EXITRTN	ED21	EXTERNAL	0002	FDALTADR	FC23	FDCMD	FC21	FDDCB	FC21
FDDLEN	000B	FDEPROM	FC22	FDNADR	02D8	FDNAMADR	FC27	FDNAME	FC29
FDNAMLEN	FC26	FDNLEN	0003	FDSTAT	FC25	FHNDLR	EC28	FID.D	0900
FID.L	1316	FID.O	DF18	FIDPARMS	E087	FILECNT	02D2	FILELEN	02A3
FILENAME	02B8	FILEND	0974	FILENTRY	02B0	FILEPNUM	02B0	FILETYPE	02B1
FILNAM	0966	FILTYPE	02D7	FINDEP	0900	FINDERR	080C	FINDFILE	FA40
FIRSTIME	02D1	FLENGTH	029E	GENPTR	00CE	GETASVAL	F398	GETENTRY	F9E5
GETEPRNG	F360	GETFILE	F00B	GETFILE2	F004	GETHEX	F759	GETKEY	F743
GETNUM	F778	GETRANGE	F367	GETVAL	F7AD	GHNDLR	EC2B	HELOTEXT	FE1F
HHNDLR	EC3A	HIRESOFF	C056	HOME	FC58	HOMECDM	0055	HOOKDOS	03EA
HOOKSLT	C010	HWCARD	0000	HWSLOT	0007	HWSLOT16	0070	HWSLOT CX	00C7
IHNDLR	EC43	INDENT	0005	INDEX	02D8	INIT	FB2F	INITCAT	F9CC
INITCMD	0052	INITDOS	BFF8	INPORT	FE8B	INPUT	0200	INTERNAL	0001
IRQRTN	FFFE	JHNDLR	EC4C	JSRMEM	F943	KBDCMD	0054	KEY	C000
KHNDLR	EC55	KSWL	0038	LABEL	0000	LARROW	0088	LCSELC	C080
LEFTOVER	1300	LENPTR	002C	LENVAL	02B4	LHNDLR	EC5E	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	FD6B	LISA1PRM	E037
LISA2PRM	E046	LISA3PRM	E055	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	E026	LISA8TRT	E000	LIST.D	8800	LIST.L	06D9
LIST.O	F878	LISTPRMS	E0A1	LLALTADR	FBFB	LLCMD	FBF9	LLDCB	FBF9
LLDLEN	0012	LLEPROM	FBFA	LLISA8.D	0900	LLISA8.L	01C0	LLISA8.O	C254
LLNADR	02D8	LLNAMADR	FBFF	LLNAME	FC01	LLNAMLEN	FBFE	LLNLEN	000A
LLSTAT	FBFD	LOADCMD	0001	LOADFILE	F112	LOADOSH	F967	LOADOSH2	F9A5
LOADOSL	F94C	LOADTBL	FD90	LOADTEXT	F11B	LSALTADR	FC66	LSCMD	FC64
LSDCB	FC64	LSDLEN	0016	LSEEPROM	FC65	LSNADR	02D8	LSNAMADR	FC6A
LSNAME	FC6C	LSNAMLEN	FC69	LSNLEN	000E	LSSTAT	FC68	LSTOPNTY	02D4

LWRCASE	00E0	LWRMASK	00DF	MAIN	E92F	MAIN2	EB46	MAIN3	EB49
MAPCODE	FB12	MAPEOS	E8AD	MAPMASKS	FD7A	MAPPAGE	0110	MAXASNUM	000C
MAXCH	0050	MEMJMP	02A6	MENUMASK	001F	MHNDLR	EC67	MINCV	0060
MNGUSER	BFF6	MONITOR	FF65	MOVEEPBM	FB5C	MSBCLR	007F	MSBSET	0080
MSLOT	00EB	NAMESIZE	0018	NEGONE	00FF	NEXTMAP	F89A	NHNDLR	EC70
NMIRTN	FFFA	NORMCMD	0051	NTRYEND	02D6	NTRYSTRT	02D5	NUMIN	029C
NUMNTRYS	02D3	NUMSCRN	02D0	NUMSELC	029D	OHNDLR	EC76	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	FD88	PHNDLR	EC79	PRBYTE	FDDA	PRHEX	FDE3	PRIMARY	0080
PRINT	F8C2	PRINT01	F8E8	PRISLOT	0290	PRNTAX	F941	PRNTMASK	000F
PRNTPTR	00FC	PROMPT	0033	PRTDEC	F880	PRTSDV	F862	PTRGET	DFE3
PWRUP0	0478	PWRUP1	0578	PWRUP2	0678	PWRUP3	0778	QHNDLR	ECA5
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	FC0D
RDCMD	FC0B	RDCXROM	C015	RDDCB	FC0B	RDDLLEN	0016	RDEPROM	FC0C
RDKEY	F73C	RDNADR	02D8	RDNAMADR	FC11	RDNAME	FC13	RDNAMLEN	FC10
RDNLLEN	000E	RDSTAT	FC0F	READADR	FAC6	READBLK	FAAC	RESERVED	0020
RESET	FFFC	RETURN	008D	RHNDLR	ECAE	RMDSKPRM	E072	ROM.D	D000
ROM.L	3000	ROM1WE	C089	ROM1WP	C08A	ROM2WE	C081	ROM2WP	C082
ROMPARMS	E0BB	RSETADR1	FA62	RSETADR2	FF59	RTN01	F899	RTNCLC	F35E
RTNCMD	0050	RTNEXIT	C4A8	RTNTYPE	0296	RUNAS	D566	RUNCMD	0002
RUNFILE	F109	RUNFLAG	029F	RUNLOAD	F0E9	RUNMODE	00FF	RUNTBL	FD9A
SAVFILE	F2D1	SAVPARM	F390	SAVPARM2	F38E	SCALTADR	FC3A	SCCMD	FC38
SCDCB	FC38	SCDLEN	0011	SCEPROM	FC39	SCNADR	02D8	SCNAMADR	FC3E
SCNAME	FC40	SCNAMLEN	FC3D	SCNLLEN	0009	SCSTAT	FC3C	SDALTADR	FD13
SDCMD	FD11	SDDCB	FD11	SDDLLEN	000C	SDEPROM	FD12	SDNADR	02D8
SDNAMADR	FD17	SDNAME	FD19	SDNAMLEN	FD16	SDNLLEN	0004	SDSTAT	FD15
SELC	EB6F	SELCBANK	FA92	SELCERR	EBD7	SELCFIL2	EF59	SELCFIL	EF34
SETANNUN	F4CE	SETASPTR	F3A3	SETKBD	FE89	SETNORM	FE84	SETUP8.D	0900
SETUP8.L	1914	SETUP8.O	A940	SETUPRMS	E064	SETUSER	F8B4	SETVID	FE93
SHNDLR	ECB7	SHOWCAT	EED8	SHOWFILE	F01E	SLOT	02A5	SLOT16	02A4
SLOT3	0003	SLOTJMP	02A8	SLOTMAP	0292	SLTTEXT	FE2B	SPACE	00A0
SPKR	C030	SRCHALL	0070	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	FE70	SYSTEM	0040	TABVCMD	0056	TEMPVAL	0297	TESTCNT	0020
TEXTFILE	0001	TEXTON	C051	TEXTTBLH	FE10	TEXTTBLL	FE0B	THNDLR	ECC0
TYPEAS	FDB1	TYPEBIN0	FDBB	TYPEBIN1	FDCF	TYPEBIN2	FDE1	TYPEPRI	FE03
TYPERSD	FDF3	TYPESYS	FDFC	TYPETBL	FDA4	TYPETEXT	FDAC	TYPTEXTS	FDAC
UARROW	008B	UHNDLR	ECC9	UHOOKSLT	C018	UNUSED	00AF	USERRTN1	E912
USERRTN2	EC10	VALUMASK	000F	VDALTADR	FCBA	VDCMD	FCB8	VDDCB	FCB8
VDDLLEN	0018	VDEPROM	FCB9	VDNADR	02D8	VDNAMADR	FCBE	VDNAME	FCC0
VDNAMLEN	FCBD	VDNLLEN	0010	VDSTAT	FCBC	VHNDLR	ECD2	VID80OFF	C00C
VIDCMD	0053	VOALTADR	FCA4	VOCMD	FCA2	VODCB	FCA2	VODLEN	0016
VOEPROM	FCA3	VOLTEXT	FE32	VOLUME	00ED	VONADR	02D8	VONAMADR	FCA8
VONAME	FCAA	VONAMLEN	FCA7	VONLEN	000E	VOSTAT	FCA6	VTAB	FC22
VTALTADR	FC90	VT CMD	FC8E	VTDCB	FC8E	VTDLLEN	0014	VTEPROM	FC8F
VTNADR	02D8	VTNAMADR	FC94	VTNAME	FC96	VTNAMLEN	FC93	VTNLLEN	000C
VTSTAT	FC92	WHNDLR	ECDB	WNDBTM	0023	WNDLFT	0020	WNDTOP	0022
WNDWDTH	0021	XHNDLR	ECE4	XMODE	04FB	XREG	0006	XSAV	0016
YHNDLR	ECED	YREG	0007	YSAV	0017	ZCACHE	029B	ZCCACHE	C05F
ZCCONFIG	F597	ZCCTRL	C05A	ZCDEFLT	FD82	ZCDELAY	C05E	ZCDISP	F6D1
ZCLOCK	00A5	ZCLOOP	F675	ZCNSPEED	0004	ZCOFFVAL	0010	ZCONVAL	0000
ZCOPTNS	000C	ZCSETBL	02D0	ZCSLOTS	C05C	ZCSPDTBL	FD84	ZCSPEED	C05D
ZCSTAT	0010	ZCSTATS	C05B	ZCTEXT	FE3E	ZCTEXT1	FE41	ZCTEXT2	FE44
ZCTEXT3	FE54	ZCTEXT4	FE5C	ZCTEXT5	FE64	ZCTEXT6	FE6A	ZCUNLOCK	005A
ZERO	0000	ZHNDLR	ECF6	ZSTATUS	029A				

Symbols numerically sorted:

ZERO	0000	ZCONVAL	0000	QLCARD	0000	LABEL	0000	HWCARD	0000
ERR00	0000	EPONVAL	0000	ENDCAT	0000	DEBUG	0000	TEXTFILE	0001
LOADCMD	0001	INTERNAL	0001	ERR01	0001	EPCARD	0001	RUNCMD	0002
EXTERNAL	0002	ERR02	0002	APLSOFT	0002	SLOT3	0003	FNDLEN	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	HOMECD	0055	TABVCMD	0056
EOLCMD	0057	EOPCMD	0058	CNTRCMD	0059	ZCUNLOCK	005A	MINCV	0060
ASPGMST	0067	ASVARS	0069	SRCHALL	0070	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	SYNCBYT2	0090	CTRLS	0093	RARROW	0095
ESCAPE	009B	DSCTMP	009D	SPACE	00A0	DOLLAR	00A4	ZCLOCK	00A5
COMMA	00AC	UNUSED	00AF	ASPEND	00AF	CHRGOT	00B7	SYNCBYT1	00B8
CHRADR	00B8	SYNCBYT0	00C4	HWSLOTX	00C7	GENPTR	00CE	ASONERR	00D8
LWRMASK	00DF	LWRCASE	00E0	MSLOT	00EB	DRIVE	00EC	VOLUME	00ED
SYNCBYT3	00ED	CMDPTR	00EE	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	SYNCBUFR	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	STARTAS	0801	FINDERR	080C
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	LISA82.L	1000	LEFTOVER	1300	FID.L	1316
SETUP8.L	1914	RAMDSK.L	1B04	PAGE20	2000	DOSL.O	2000	BANKSIZE	2000
DOSL.L	2100	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	EPASEOS	C400	RAMDSK.O	C414	ASEXIT	C450
BINEXIT	C480	RTNEXIT	C4A8	EPOFF	C4B8	EPUSER1	C4C0	EPUSER2	C4C8
EPMAPEOS	C4D0	EPBINEOS	C4E0	EPEOS	C4F0	EPBINTXT	C4F8	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	CATALOG	E000	DOSLPRMS	E004	DOSHPRMS	E015
LISAPRMS	E026	LISA1PRM	E037	LISA2PRM	E046	LISA3PRM	E055	SETUPRMS	E064
RMDSKPRM	E072	FIDPARMS	E087	CLKPARMS	E091	LISTPRMS	E0A1	ROMPARMS	E0BB
CATPARMS	E0CA	PAGEE1	E100	STRINI	E3D5	PAGEE7	E700	PAGEE8	E800
EOS	E800	MAPEOS	E8AD	USERRTN1	E912	MAIN	E92F	PAGEEA	EA00
MAIN2	EB46	MAIN3	EB49	SELC	EB6F	SELCERR	EBD7	AHNDLR	EBEC
BHNDLR	EBF3	BHNDLR2	EBF8	CHNDLR	EC09	USERRTN2	EC10	DHNDLR	EC13
EHNDLR	EC1D	FHNDLR	EC28	GHNDLR	EC2B	HHNDLR	EC3A	IHNDLR	EC43
JHNDLR	EC4C	KHNDLR	EC55	LHNDLR	EC5E	MHNDLR	EC67	NHNDLR	EC70
OHNDLR	EC76	PHNDLR	EC79	QHNDLR	ECA5	RHNDLR	ECAE	SHNDLR	ECB7
THNDLR	ECC0	UHNDLR	ECC9	VHNDLR	ECD2	WHNDLR	ECDB	XHNDLR	ECE4
YHNDLR	ECED	ZHNDLR	ECF6	DOEOSDCB	ED04	EXITAS	ED1B	EXITBIN	ED1E
EXITRTN	ED21	EXECTEXT	ED3A	DOEXEC	ED4B	DOEXEC2	ED59	EOSCAT	ED63
EOSCAT2	ED66	CATHDR	EE13	CATFTR	EE57	CHKCAT	EEAE	SHOWCAT	EED8
SELCFIL	EF34	SELCFIL2	EF59	CKCATCMD	EFCA	GETFILE2	F004	GETFILE	F00B
SHOWFILE	F01E	RUNLOAD	F0E9	RUNFILE	F109	LOADFILE	F112	LOADTEXT	F11B
EXECAS	F1BF	EXECBIN1	F1EF	EXECBIN2	F1F8	EXECBIN0	F1FE	ASEOS	F210
CLK.O	F22E	DOASFILE	F285	DOASCAT	F2AD	SAVFILE	F2D1	RTNCLC	F35E
GETEPRNG	F360	GETRANGE	F367	SAVPARM2	F38E	SAVPARM	F390	GETASVAL	F398
SETASPTR	F3A3	BINEOS	F3B0	BINEOS2	F3CC	BINDONE	F3EF	BINLOAD	F408
BINRUN	F409	BINCAT	F43F	DOZCOFF	F49F	SETANNUN	F4CE	DOZCON	F4DB
DOZCRSET	F4F6	DOZCREAD	F510	DOZCSAVE	F53E	DOZCOPEN	F577	ZCCONFIG	F597
ZCLOOP	F675	ZCDISP	F6D1	EOSBELL	F72C	RDKEY	F73C	GETKEY	F743
GETHEX	F759	GETNUM	F778	GETVAL	F7AD	EDITS DV	F7FA	PRTSDV	F862
LIST.O	F878	PRTDEC	F880	RTN01	F899	NEXTMAP	F89A	CLRUSER	F8B2
SETUSER	F8B4	PRINT	F8C2	PRINT01	F8E8	DOVTAB	F91A	DOSHOK	F91E
DOPRNTAX	F922	DOPRBYTE	F928	DOPRHEX	F92C	DOSPACE	F930	DOCROUT	F933
DOCOUT	F935	DOJSRMEM	F939	PRNTAX	F941	JSRMEM	F943	LOADOSL	F94C
LOADOSH	F967	LOADOSH2	F9A5	COPYROM	F9AC	INITCAT	F9CC	GETENTRY	F9E5
FINDFILE	FA40	RSETADR1	FA62	COPYPRM0	FA7F	COPYPARM	FA84	SELCBANK	FA92
READBLK	FAAC	READADR	FAC6	BUILDMAP	FAE1	MAPCODE	FB12	INIT	FB2F
MOVEEPBM	FB5C	EPBMCODE	FB7F	LLDCB	FBF9	LLCMD	FBF9	LLEPROM	FBFA
LLALTADR	FBFB	LLSTAT	FBFD	LLNAMLEN	FBFE	LLNAMADR	FBFF	LLNAME	FC01
RDDCB	FC0B	RDCMD	FC0B	RDEPROM	FC0C	RDALTADR	FC0D	RDSTAT	FC0F
RDNAMLEN	FC10	RDNAMADR	FC11	RDNAME	FC13	FDDCB	FC21	FDCMD	FC21
VTAB	FC22	FDEPROM	FC22	FDALTADR	FC23	FDSTAT	FC25	FDNAMLEN	FC26
FDNAMADR	FC27	FDNAME	FC29	ADDCB	FC2C	ADCMD	FC2C	ADEPROM	FC2D

ADALTADR	FC2E	ADSTAT	FC30	ADNAMLEN	FC31	ADNAMADR	FC32	ADNAME	FC34
SCDCB	FC38	SCCMD	FC38	SCEPROM	FC39	SCALTADR	FC3A	SCSTAT	FC3C
SCNAMLEN	FC3D	SCNAMADR	FC3E	SCNAME	FC40	CLREOP	FC42	AFDCB	FC49
AFCMD	FC49	AFEPROM	FC4A	AFALTADR	FC4B	AFSTAT	FC4D	AFNAMLEN	FC4E
AFNAMADR	FC4F	AFNAME	FC51	HOME	FC58	LSDCB	FC64	LSCMD	FC64
LSEEPROM	FC65	LSALTADR	FC66	LSSTAT	FC68	LSNAMLEN	FC69	LSNAMADR	FC6A
LSNAME	FC6C	EBDCB	FC7A	EBCMD	FC7A	EBEPROM	FC7B	EBALTADR	FC7C
EBSTAT	FC7E	EBNAMLEN	FC7F	EBNAMADR	FC80	EBNAME	FC82	VTDCB	FC8E
VT CMD	FC8E	VTEPROM	FC8F	VTALTADR	FC90	VTSTAT	FC92	VTNAMLEN	FC93
VTNAMADR	FC94	VTNAME	FC96	CLREOL	FC9C	VODCB	FCA2	VOCMD	FCA2
VOEPROM	FCA3	VOALTADR	FCA4	VOSTAT	FCA6	VONAMLEN	FCA7	VONAMADR	FCA8
VONAME	FCAA	VDDCB	FCB8	VDCMD	FCB8	VDEPROM	FCB9	VDALTADR	FCBA
VDSTAT	FCBC	VDNAMLEN	FCBD	VDNAMADR	FCBE	VDNAME	FCC0	DWDCB	FCD0
DWCMD	FCD0	DWEEPROM	FCD1	DWALTADR	FCD2	DWSTAT	FCD4	DWNAMLEN	FCD5
DWNAMADR	FCD6	DWNAME	FCD8	BFDCB	FCE3	BFCMD	FCE3	BFEPROM	FCE4
BFALTADR	FCE5	BFSTAT	FCE7	BFNAMLEN	FCE8	BFNAMADR	FCE9	BFNAME	FCEB
BMDCB	FCFE	BMCMD	FCFE	BMEPROM	FCFF	BMALTADR	FD00	BMSTAT	FD02
BMNAMLEN	FD03	BMNAMADR	FD04	BMNAME	FD06	SDDCB	FD11	SDCMD	FD11
SDEPROM	FD12	SDALTADR	FD13	SDSTAT	FD15	SDNAMLEN	FD16	SDNAMADR	FD17
SDNAME	FD19	CMDTBL	FD1D	ADDRTBLL	FD51	ADDRTBLH	FD5E	LINETBL	FD6B
DECTBLL	FD77	MAPMASKS	FD7A	ZCDEFLT	FD82	ZCSPDTBL	FD84	PARMTYPE	FD88
CROUT	FD8E	LOADTBL	FD90	RUNTBL	FD9A	TYPETBL	FDA4	TYPTXTS	FDAC
TYPETEXT	FDAC	TYPEAS	FDB1	TYPEBIN0	FDBB	TYPEBIN1	FDCF	PRBYTE	FDDA
TYPEBIN2	FDE1	PRHEX	FDE3	COUT	FDED	TYPERESD	FDF3	TYPESYS	FDFC
TYPEPRI	FE03	TEXTTBL	FE0B	TEXTTBLH	FE10	CATTEXT	FE15	HELOTEXT	FE1F
SLTTEXT	FE2B	DRVTEXT	FE2E	VOLTEXT	FE32	EPTEXT	FE36	ZCTEXT	FE3E
ZCTEXT1	FE41	ZCTEXT2	FE44	ZCTEXT3	FE54	ZCTEXT4	FE5C	ZCTEXT5	FE64
ZCTEXT6	FE6A	SYNCBYTS	FE70	SETNORM	FE84	SETKBD	FE89	INPORT	FE8B
SETVID	FE93	OUTPORT	FE95	RSETADR2	FF59	MONITOR	FF65	NMIRTN	FFFA
EOSVCTRS	FFFA	RESET	FFFC	IRQRTN	FFFE				