


```

0000 1: ;-----
0000 2: ;
0000 3: ;      Copyright 2005 Intergraded Logic Systems
0000 4: ;      Source Code is Copyright Stephen J. Car
0000 5: ;
0000 6: ;
0000 7: ;      Please do not share this source!
0000 8: ;-----
0000 9: ;
0000 10: ;      Source Code is Copyright 2006 Stephen J. C
0000 11: ;-----
0000 12: ;      Megadisk.S
0000 13: ;
0000 14: ;      7/24/2006      Fixed end of line error
0000 15: ;
0000 16: ;
0000 17: ;
0000 18: ;      File dissamble tool
0000 19: ;      ~~~~~
0000 20: ;-----
0000 21: ; Notes: o This source code MAY NOT be placed for download
0000 22: ;      o "mea" is a macro that loads the address of the
0000 23: ;      into the pointer specified by the second field.
0000 24: ;-----
0000 25: ; Assembler: MADMAC (tm) ST Cross Assembler (Atari Corp)
0000 26: ;-----
0000 27: ;
0000 28: ; run address $2100
0000 29: ;
0000 30: ; ... EQ ...
0000 31: ;
0000 32: ;System equates
0000 33: ;-----
0000 34: ;
0000 35: dosvec: equ $0a
0000 36: dosini: equ $0c
0000 37: appmhi: equ $0e
0000 38: lmarg: equ $52
0000 39: rowcrs: equ $54
0000 40: colcrs: equ $55
0000 41: savmsc: equ $58
0000 42: fr0: equ $d4
0000 43: cix: equ $f2
0000 44: inbuff: equ $f3
0000 45: vvblkd: equ $0224
0000 46: sdmctl: equ $022f
0000 47: sdlstl: equ $0230
0000 48: escflg: equ $02a2
0000 49: invflg: equ $02b6
0000 50: shflok: equ $02be
0000 51: color0: equ $02c4
0000 52: memtop: equ $02e5
0000 53: memlo: equ $02e7

```

```

0000      54: crsinh: equ  $02f0
0000      55: ch:   equ  $02fc
0000      56: ddevic: equ  $0300
0000      57: dunit: equ  $0301
0000      58: dcomnd: equ  $0302
0000      59: dstats: equ  $0303
0000      60: dbuflo: equ  $0304
0000      61: dbufhi: equ  $0305
0000      62: daux1:  equ  $030a
0000      63: daux2:  equ  $030b
0000      64: iccom:  equ  $0342
0000      65: icbal:  equ  $0344
0000      66: icbah:  equ  $0345
0000      67: icbll:  equ  $0348
0000      68: icblh:  equ  $0349
0000      69: icax1:  equ  $034a
0000      70: consol: equ  $d01f
0000      71: audf1:  equ  $d200
0000      72: audc1:  equ  $d201
0000      73: afp:    equ  $d800
0000      74: fasc:   equ  $d8e6
0000      75: ifp:    equ  $d9aa
0000      76: fpi:    equ  $d9d2
0000      77: dskinv: equ  $e453
0000      78: ciov:   equ  $e456
0000      79: setvbv: equ  $e45c
0000      80: xitvbv: equ  $e462
0000      81: ;-----
0000      82: ;
0000      83: ;Zero-page equates
0000      84: ;-----
0000      85: ;
0000      86: ;      .org $b0
0000      87: buffer: equ $b0  ;ds.b 2
0000      88: bufptr: equ $b2  ;ds.b 2
0000      89: buftop: equ $b4  ;ds.b 2
0000      90: bufsize: equ $b6  ;ds.b 2
0000      91: syslabel: equ $b8  ;ds.b 2
0000      92: label.adr: equ $ba  ;      ;.org *+2
0000      93: label.ptr: equ $bc  ;      ;.org *+2
0000      94: this.lbl: equ $be  ;      ;ds.b 2
0000      95: this.lptr: equ $c0  ;      ;.org *+2
0000      96: sptr:    equ $c2  ;ds.b 2
0000      97: pointr: equ $c4  ;ds.b 2
0000      98: pc:      equ $c6  ;ds.b 2
0000      99: s.org:   equ $c8  ;ds.b 2
0000     100: number: equ $ca  ;ds.b 2
0000     101: spaces: equ $cc  ;ds.b 1
0000     102: ytemp:  equ $cd  ;ds.b 1
0000     103: tempy:  equ $ce  ;ds.b 1
0000     104: ;
0000     105: ; Misc Equates
0000     106: ; -----

```

```
0000 107: ;
0000 108: esc: equ $1b
0000 109: space: equ $20
0000 110: cr: equ $9b
0000 111: cls.: equ $7d
0000 112: backsp: equ $7e
0000 113: tab: equ $7f
0000 114: em: equ $ff
0000 115: ;
0000 116: chan1: equ $10
0000 117: chan2: equ $20
0000 118: chan3: equ $30
0000 119: chan4: equ $40
0000 120: ;
0000 121: open.: equ 3
0000 122: bget.: equ 7
0000 123: putrec.: equ 9
0000 124: bput.: equ $0b
0000 125: close.: equ $0c
0000 126: ;
0000 127: xread: equ 4
0000 128: xdir: equ 7
0000 129: xwrite: equ 8
0000 130: ;
0000 131: dbsize: equ icbll+$20
0000 132: maxsec: equ 720
0000 133: dummy: equ $ffff
0000 134: ;-----
0000 135: ;
0000 136: ;
0000 137: ;Macro Definitions
0000 138: ;-----
0000 139: ;
0000 140: ;INW Word addr.
0000 141: ;
0000 142: ;
0000 143: ; .macro inw
0000 144: ; inc %1
0000 145: ; bne .Az
0000 146: ; inc %1+1
0000 147: ;.Az:
0000 148: ; .ENDM
0000 149:
0000 150: inw: .macro field
0000 151: inc %1
0000 152: bne ~a
0000 153: inc %1+1
0000 154: ~a:
0000 155: .endm
0000 156: ;-----
0000 157:
0000 158: ;
0000 159: ; .PAGE "Main Loop"
```

```

0000      160: ;      .include main
0000      161:
0000      162: ;
2100      163:      .org $2100
2100      164: ;
2100 4c3521 165: beginsave: jmp  init
2103      166:
2103      167: ;      dc.b  $22  ; version
2103      168: ;      dc.b  8    ; month
2103      169: ;      dc.b  24   ; day
2103      170: ;      dc.b  12   ; year
2103      171: ;      dc.w  31   ; dos build number Crc will
2103      172: ;      dc.w  2012
2103      173: ;      dc.w  beginsave
2103      174: ;      dc.w  win_end
2103      175: ;      dc.b  0
2103      176: ;      dc.b  0
2103      177: ;      dc.w  0
2103      178: ;
2103      179: ;      dc.b  1
2103      180: ;      dc.b  2
2103      181: ;      dc.b  1
2103      182: ;      dc.w  31   ; current dos build
2103      183: ;      dc.b  1
2103      184: ;
2103      185:
2103      186: ;
2103 707090 187: myscreen:  dc.b $70,$70,$90
2106 42     188:      dc.b $42
2107 0d21   189: dladr:  dc.w line1
2109 0001   190:      dc.b 0,1
210b ffff   191: oldscr: dc.w $ffff
210d      192: ;
210d      193: ;
210d 80ade5 194: line1:  dc.b <internal>, <+128>," Megadisk by Stephen
2135      195: ;
2135      196: ;-----
2135      197: ;
2135      198: ;Main Entry Point.
2135      199: ;Check to see if SpartaDOS
2135      200: ;is being used and point
2135      201: ;vector to RESET if so.
2135      202: ;Else skip to ENTER
2135      203: ;
2135 ad0007 204: init:   lda $0700 ;Sparta DOS?
2138 c953   205:      cmp #'S'
213a d00f   206:      bne enter
213c c952   207:      cmp #'R'   ; real.dos
213e d00b   208:      bne enter
2140 a03d   209:      ldy #$3d   ;Yes, point SD
2142 a94b   210:      lda # low enter ;RUN vector to
2144 910a   211:      sta (dosvec),y ;RESET.
2146 a921   212:      lda # high enter

```

```
2148 c8 213: iny
2149 910a 214: sta (dosvec),y
214b 215: ;
214b 216: ;Save left margin, screen
214b 217: ;colors, DOSVEC, and address
214b 218: ;of any resident VBI.
214b 219: ;
214b a552 220: enter: lda lmarg ;Save left marg
214d 8dce62 221: sta oldmar ;and screen
2150 a004 222: ldy #4 ;colors.
2152 b9c402 223: .savcolor: lda color0,y
2155 99cf62 224: sta oldcolr,y
2158 88 225: dey
2159 10f7 226: bpl .savcolor
215b a50c 227: lda dosini
215d 8dca62 228: sta dvec
2160 8d7a21 229: sta reset+1
2163 a50d 230: lda dosini+1
2165 8dcb62 231: sta dvec+1
2168 8d7b21 232: sta reset+2
216b ad2402 233: lda vvblkd ;Save Old VBI
216e 8dcc62 234: sta oldvbi ;address.
2171 ad2502 235: lda vvblkd+1
2174 8dcd62 236: sta oldvbi+1
2177 d003 237: bne e.ovrset
2179 238: ;
2179 239: ;Skip over RESET first time
2179 240: ;thru. This is where RESET
2179 241: ;button is trapped to.
2179 242: ;
2179 20ffff 243: reset: jsr dummy
217c 244: ;
217c 245: ;Trap RESET, set VBI routine
217c 246: ;to point to our VBI, Close
217c 247: ;then re-open the scr editor.
217c 248: ;
217c a979 249: e.ovrset: lda # low reset
217e 850c 250: sta dosini
2180 a921 251: lda # high reset
2182 850d 252: sta dosini+1
2184 208a25 253: start: jsr resetvbi
2187 a900 254: lda #0 ;Open Screen
2189 8d2f02 255: sta sdmctl ;editor.
218c 203d37 256: jsr opene
218f 257: ;
218f 258: ;Set up custom display list.
218f 259: ;
218f ad9361 260: lda kolor
2192 8dc602 261: sta 710
2195 ad9461 262: lda lum
2198 8dc502 263: sta 709
219b 18 264: clc
219c ad3002 265: lda sdlstl
```

```

219f 85c2 266:    sta sptr
21a1 6903 267:    adc #3
21a3 8d0b21 268:    sta oldscr
21a6 ad3102 269:    lda sdlstl+1
21a9 85c3 270:    sta sptr+1
21ab 6900 271:    adc #0
21ad 8d0c21 272:    sta oldscr+1
21b0 a01e 273:    ldy #30
21b2 a903 274:    lda # low myscreen
21b4 91c2 275:    sta (sptr),y
21b6 8d3002 276:    sta sdlstl
21b9 c8 277:    iny
21ba a921 278:    lda # high myscreen
21bc 91c2 279:    sta (sptr),y
21be 8d3102 280:    sta sdlstl+1
21c1      281: ;Close all channels except
21c1      282: ;channel 0, then initialize
21c1      283: ;variables.
21c1      284: ;
21c1 a2ff 285:    ldx #$ff
21c3 9a 286:    txs
21c4 208a3f 287:    jsr closeall
21c7      288: ;
21c7      289: ;Zero out label bitmap.
21c7      290: ;
21c7 a05e 291:    ldy # low label.map
21c9 84c2 292:    sty sptr
21cb a064 293:    ldy # high label.map
21cd 84c3 294:    sty sptr+1
21cf a201 295:    ldx #1
21d1 a000 296:    ldy #0
21d3 98 297:    tya
21d4 8de862 298:    sta binflag
21d7 91c2 299: .zero_label.map: sta (sptr),y
21d9 88 300:    dey
21da d0fb 301:    bne .zero_label.map
21dc e6c3 302:    inc sptr+1
21de ca 303:    dex
21df 10f6 304:    bpl .zero_label.map
21e1      305: ;
21e1      306: ;Init variables.
21e1      307: ;
21e1 a25a 308:    ldx #90
21e3 9dd462 309: .aloop: sta try,x
21e6 ca 310:    dex
21e7 10fa 311:    bpl .aloop
21e9      312: ;
21e9      313: ;Clear address line to spaces.
21e9      314: ;
21e9 a920 315:    lda #space
21eb a205 316:    ldx #5
21ed 9de062 317: .clr_adr: sta address,x
21f0 ca 318:    dex

```

```
21f1 10fa 319:      bpl .clr_adr
21f3      320:      ;
21f3      321:      ;Initialize line number
21f3      322:      ;to 4995.
21f3      323:      ;
21f3 a205 324:      ldx #5
21f5 bdb461 325: .lin0:  lda frstno,x
21f8 9d3863 326:      sta lnum,x
21fb ca 327:      dex
21fc 10f7 328:      bpl .lin0
21fe 8efe62 329:      stx dskon
2201      330:      ;
2201      331:      ;Set TO.LINO to jump to
2201      332:      ;GET_LINO which increments
2201      333:      ;line numbers by 5, then
2201      334:      ;set left margin and screen
2201      335:      ;colors
2201      336:      ;
2201 a92a 337:      lda # low get_lino
2203 8d3663 338:      sta to.lino
2206 a93d 339:      lda # high get_lino
2208 8d3763 340:      sta to.lino+1
220b 20f53e 341:      jsr light
220e a901 342:      lda #1
2210 8552 343:      sta lmarg
2212 85bf 344:      sta this.lbl+1
2214      345:      ;
2214      346:      ;Figure and set top of free
2214      347:      ;memory 1 page lower than
2214      348:      ;MEMTOP.
2214      349:      ;
2214 38 350:      sec
2215 ade502 351:      lda memtop
2218 8de702 352:      sta memlo
221b 850e 353:      sta appmhi
221d 8df562 354:      sta freehi
2220 ade602 355:      lda memtop+1
2223 e901 356:      sbc #1
2225 8de802 357:      sta memlo+1
2228 850f 358:      sta appmhi+1
222a 8df662 359:      sta freehi+1
222d      360:      ;
222d      361:      ;Point LABELTOP to beginning
222d      362:      ;of LABEL.BUFFER.
222d      363:      ;
222d a95e 364:      lda # low label.buffer
222f 8d1063 365:      sta labeltop
2232 a967 366:      lda # high label.buffer
2234 8d1163 367:      sta labeltop+1
2237 a95e 368:      lda # low bufadr
2239 85b0 369:      sta buffer
223b 85b4 370:      sta buftop
223d 85b2 371:      sta bufptr
```



```

223f a97d 372:    lda # high bufadr
2241 85b1 373:    sta buffer+1
2243 85b5 374:    sta buftop+1
2245 85b3 375:    sta bufptr+1
2247      376:    ;
2247      377:    ;Unmark previously used
2247      378:    ;operating system labels.
2247      379:    ;
2247 a97a 380:    .ptr_setup: lda # low page0
2249 85b8 381:    sta syslabel
224b a94c 382:    lda # high page0
224d 85b9 383:    sta syslabel+1
224f a000 384:    ldy #0
2251 20153e 385:    .offloop: jsr and7f
2254 91b8 386:    sta (syslabel),y
2256 e6b8 387:    inc syslabel
2258 d002 388:    bne .lonly
225a e6b9 389:    inc syslabel+1
225c a5b8 390:    .lonly: lda syslabel
225e c9e4 391:    cmp # low page.table
2260 d0ef 392:    bne .offloop
2262 a5b9 393:    lda syslabel+1
2264 c95d 394:    cmp # high page.table
2266 d0e9 395:    bne .offloop
2268      396:    ;
2268      397:    ;Show menu and get users wish.
2268      398:    ;
2268      399:    menu:
2268      400:    ;
2268 208a25 401:    jsr resetvbi
226b 20113f 402:    jsr newscreen
226e      403:    ; position 9,1
226e a906 404:    lda #6
2270 8555 405:    sta colcrs
2272 a901 406:    lda #1
2274 8554 407:    sta rowcrs
2276      408:
2276 208037 409:    jsr printsi
2279 577269 410:    dc.b "Written By Stephen J. Carden",-1
2296      411:
2296 a90b 412:    lda #11
2298 8552 413:    sta lmarg
229a 8555 414:    sta colcrs
229c a903 415:    lda #3
229e 8554 416:    sta rowcrs
22a0 208037 417:    jsr printsi
22a3 c3 418:    dc.b <+128>,"C"
22a4 68616e 419:    dc.b "hange Defaults",CR,CR
22b4 a3 420:    dc.b <+128>,"#"
22b5 204469 421:    dc.b " Disk Directory",CR,CR
22c6 c6 422:    dc.b <+128>,"F"
22c7 696c65 423:    dc.b "ile Disassembly",CR,CR
22d8 cd 424:    dc.b <+128>,"M"

```

```

22d9 656d6f 425:    dc.b "emory Disassembly",CR,CR
22ec d3 426:    dc.b <+128>,"S"
22ed 656374 427:    dc.b "ector Disassembly",CR,CR
2300 c1 428:    dc.b <+128>,"A"
2301 637469 429:    dc.b "ction! Code Block",CR,CR
2314 c2 430:    dc.b <+128>,"B"
2315 595445 431:    dc.b "YTE file",CR,CR
231f c4 432:    dc.b <+128>,"D"
2320 415441 433:    dc.b "ATA file",CR,CR
232a c5f3e3 434:    dc.b <+128>,"Escape"
2330 20746f 435:    dc.b " to DOS"
2337 9b9b9b 436:    dc.b CR,CR,CR,EM
233b a901 437:    lda #1
233d 8552 438:    sta lmarg
233f 8df002 439:    sta crsinh
2342 440:    ; col 13
2342 a90d 441:    lda #13
2344 8555 442:    sta colcrs
2346 443:
2346 208037 444:    jsr printsi
2349 596f75 445:    dc.b "Your Choice? ",EM
2357 20dc37 446: choice.: jsr getcap
235a 447:    ;
235a 448:    ;Save users choice and
235a 449:    ;then decode it.
235a 450:    ;
235a 8d2563 451:    sta mode
235d 20a83e 452:    jsr cheknum ;Check for #
2360 b003 453:    bcs .keychk ;which means
2362 4c714b 454:    jmp directory ;show DIR.
2365 455:    ;
2365 a009 456: .keychk: ldy #9    ;Good keypress?
2367 d99223 457: .kyloop: cmp keytable,y
236a f005 458:    beq .goodkey
236c 88 459:    dey    ;No.
236d 10f8 460:    bpl .kyloop
236f 30e6 461:    bmi choice.
2371 462:    ;
2371 98 463: .goodkey: tya    ;Yes, Multiply
2372 0a 464:    asl    ;Y by two to
2373 a8 465:    tay    ;point to right
2374 b99c23 466:    lda menutabl,y ;adr in table
2377 8d9023 467:    sta menujmp+1 ;and point
237a b99d23 468:    lda menutabl+1,y ;menujmp to
237d 8d9123 469:    sta menujmp+2 ;that address.
2380 ad9761 470:    lda line.start ;Point LINE.PTR
2383 8d2163 471:    sta line.ptr ;to proper
2386 ad9861 472:    lda line.start+1 ;output line.
2389 8d2263 473:    sta line.ptr+1
238c ad2563 474:    lda mode
238f 4cffff 475: menujmp: jmp dummy ;Go do it.
2392 476:    ;
2392 477:    ;-----

```

```

2392      478: ;Table of Menu Keys.
2392      479: ;
2392 43    480: keytable: dc.b "C"
2393 46    481:      dc.b "F"
2394 4d    482:      dc.b "M"
2395 53    483:      dc.b "S"
2396 41    484:      dc.b "A"
2397 42    485:      dc.b "B"
2398 44    486:      dc.b "D"
2399 1b    487:      dc.b $1B
239a 4b    488:      dc.b "K"
239b 4c    489:      dc.b "L" ;Hidden (color).
239c      490: ;
239c      491: ;Address table for Menu Choices.
239c      492: ;
239c ce23  493: menutabl: dc.w opt
239e d423  494:      dc.w dis
23a0 d423  495:      dc.w dis
23a2 d423  496:      dc.w dis
23a4 ff32  497:      dc.w datafile
23a6 ff32  498:      dc.w datafile
23a8 ff32  499:      dc.w datafile
23aa e824  500:      dc.w exit
23ac c023  501:      dc.w .klr
23ae b023  502:      dc.w .lm
23b0      503: ;-----
23b0      504: ;
23b0      505: ;
23b0 18    506: .lm:   clc      ;Change Character
23b1 adc502 507:      lda 709 ;Luminance.
23b4 6902  508:      adc #2
23b6 290f  509:      and #15
23b8 8dc502 510:      sta 709
23bb 8d9461 511:      sta lum
23be 1097  512:      bpl choice.
23c0      513: ;
23c0 adc602 514: .klr:   lda 710 ;Change
23c3 6910  515:      adc #16 ;Background
23c5 8dc602 516:      sta 710 ;color
23c8 8d9361 517:      sta kolor
23cb 4c5723 518:      jmp choice.
23ce      519: ;
23ce 205544 520: opt:   jsr options ;Set Options.
23d1 4c6822 521:      jmp menu
23d4      522: ;
23d4      523: ;
23d4      524: ;Must point LINE.PTR to either
23d4      525: ;PRINTLINE, if line numbers are
23d4      526: ;not to be included, or LNUM
23d4      527: ;if they are. Do it here in
23d4      528: ;case the option was changed
23d4      529: ;by using the OPTIONS menu.
23d4      530: ;

```

```
23d4 a23b 531: dis:   ldx # high vbi
23d6 a033 532:       ldy # low vbi
23d8 a907 533:       lda #7
23da 205ce4 534:       jsr setvbv
23dd ad2563 535:       lda mode
23e0 c946 536:       cmp #'F'
23e2 d010 537:       bne .m.
23e4 a9bd 538:       lda # low file_dis ;Point
23e6 8dc629 539:       sta modejump+1 ;Modejump to
23e9 a92e 540:       lda # high file_dis ;file
23eb 8dc729 541:       sta modejump+2 ;disassembly.
23ee 206840 542:       jsr get_infn ;Get Input fn.
23f1 4c4824 543:       jmp get_outfn
23f4      544: ;
23f4 c94d 545: .m.:   cmp #'M'
23f6 d04d 546:       bne .s.
23f8      547: ;
23f8      548: ;Set up for memory disassembly.
23f8      549: ;Point MODEJUMP to memory
23f8      550: ;disassembly routine.
23f8      551: ;
23f8 a974 552:       lda # low memory_dis
23fa 8dc629 553:       sta modejump+1
23fd a92c 554:       lda # high memory_dis
23ff 8dc729 555:       sta modejump+2
2402      556: ;
2402      557: ;Find out what memory to
2402      558: ;disassemble.
2402      559: ;
2402 202840 560:       jsr get_start
2405      561: ;
2405      562: ;Initialize BUFPTR and
2405      563: ;BUFTOP to zero.
2405      564: ;
2405      565: ;
2405      566: ;Check first char of input.
2405      567: ;
2405 208037 568: .end_in: jsr printsi
2408 9b7f45 569:       dc.b cr,tab,"Enter End addre"
2419 73733a 570:       dc.b "ss: ",EM
2420      571: ;
2420      572: ;Convert end address.
2420      573: ;
2420 204d38 574:       jsr getnum
2423 c91b 575:       cmp #esc
2425 d003 576:       bne .digend
2427 4c963f 577:       jmp clrstack
242a      578: ;
242a 203836 579: .digend: jsr digitize
242d b010 580:       bcs .end_err
242f a5ca 581:       lda number
2431 85b4 582:       sta buftop
2433 a5cb 583:       lda number+1
```

```
2435 85b5 584:      sta buftop+1
2437 a901 585:      lda #1
2439 8df262 586:      sta more.
243c 4c4824 587:      jmp get_outfn
243f      588: ;
243f      589: ;
243f 201c44 590: .end_err: jsr in_err
2442 4c0524 591:      jmp .end_in
2445      592: ;
2445      593: ;
2445 208a2c 594: .s.:   jsr getsectors
2448      595: ;
2448      596: ;
2448      597: ;Get output filename from user.
2448      598: ;
2448 209c3f 599: get_outfn: jsr outfn_in
244b 9008 600:      bcc .todsk
244d a900 601:      lda #0      ;Flag indicating
244f 8dfe62 602:      sta dskon    ;no disk output.
2452 4c7e24 603:      jmp a.ready
2455      604: ;
2455      605: ;Indicate disk file not open,
2455      606: ;initialize disk file count to
2455      607: ;1, set DRIVES to 1 less than
2455      608: ;number of drives being used.
2455      609: ;
2455 a001 610: .todsk: ldy #1
2457 adaf63 611:      lda outfn+1
245a cd8f63 612:      cmp infn+1
245d d001 613:      bne .knodrv
245f 88 614:      dey
2460 8cf062 615: .knodrv: sty drives
2463 ad2563 616:      lda mode
2466 c94d 617:      cmp #'M'
2468 f011 618:      beq got_info
246a adf062 619:      lda drives
246d 8de562 620:      sta both.flag
2470      621: ;
2470 20113f 622: .drvprmt: jsr newscreen
2473 20a940 623:      jsr indisk_prmt
2476 adf062 624:      lda drives
2479 f003 625:      beq a.ready
247b      626: ;
247b 201141 627: got_info: jsr outdisk_prmt
247e      628: ;
247e ad2563 629: a.ready: lda mode
2481 c946 630:      cmp #'F'
2483 d025 631:      bne .getset
2485 20053b 632:      jsr open_in ;Check for
2488 a900 633:      lda #0      ;DOS load file
248a 8ddd62 634:      sta dskopnflg ;by looking
248d 20863a 635:      jsr load_adr ;for $FFFF
2490 f006 636:      beq .setset ;file header.
```

```
2492 207641 637:      jsr not_binary
2495 4ca724 638:      jmp .opnreal
2498      639: ;
2498 20863a 640: .setset: jsr load_adr ;Get file
249b adfa62 641:      lda seg.org ;ORG address
249e 8dea62 642:      sta first ;and save it.
24a1 adfb62 643:      lda seg.org+1
24a4 8deb62 644:      sta first+1
24a7 20053b 645: .opnreal: jsr open_in
24aa      646: ;
24aa      647: ;
24aa 201d25 648: .getset: jsr loopinit
24ad ad9561 649:      lda scr ;Set screen and
24b0 8d2f02 650:      sta sdmctl ;printer to
24b3 ad9661 651:      lda prt ;default on/off.
24b6 8d1c63 652:      sta close.p.
24b9 209325 653:      jsr build_symbol_tbl
24bc 206225 654:      jsr setbuffer
24bf 201d25 655:      jsr loopinit
24c2 ad2563 656:      lda mode
24c5 8dd562 657:      sta loadflag
24c8 c946 658:      cmp #'F'
24ca d00d 659:      bne .notf
24cc adea62 660:      lda first ;Copy file
24cf 85c6 661:      sta pc ;ORG address
24d1 adeb62 662:      lda first+1 ;into program
24d4 85c7 663:      sta pc+1 ;counter.
24d6 4cdf24 664:      jmp .dodis
24d9      665: ;
24d9 20b93c 666: .notf: jsr sho_org
24dc 20923c 667:      jsr compute_address
24df 20c829 668: .dodis: jsr disassemble
24e2 205b30 669:      jsr write_equates
24e5      670: ;
24e5      671: ;Indicate to user that we are
24e5      672: ;done and see what (s)he wants
24e5      673: ;to do.
24e5      674: ;
24e5 4c8a42 675: disdone: jmp nowhat.
24e8      676: ;
24e8      677: ;Reopen scr editor, restore
24e8      678: ;all the stuff saved when the
24e8      679: ;program was entered, reset
24e8      680: ;the stack pointer, then
24e8      681: ;exit to address in DOSVEC.
24e8      682: ;
24e8      683: exit:
24e8      684: ;
24e8 203d37 685:      jsr opene
24eb adce62 686:      lda oldmar
24ee 8552 687:      sta lmarg
24f0 a004 688:      ldy #4
24f2 b9cf62 689: .fixit: lda oldcolr,y
```

```

24f5 99c402 690:    sta color0,y
24f8 88      691:    dey
24f9 10f7    692:    bpl .fixit
24fb c8      693:    iny
24fc 8cf002 694:    sty crsinh ;Turn on cursor.
24ff adca62 695:    lda dvec  ;Restore DOSINI.
2502 850c    696:    sta dosini
2504 adcb62 697:    lda dvec+1
2507 850d    698:    sta dosini+1
2509 a907    699:    lda #7    ;Restore old VBI
250b aecd62 700:    ldx oldvbi+1 ;routine
250e accc62 701:    ldy oldvbi
2511 205ce4 702:    jsr setvbv
2514 208a3f 703:    jsr closeall ;Close chans 1-7.
2517 a2ff    704:    ldx #$ff  ;Clear stack.
2519 9a      705:    txs
251a 6c0a00 706:    jmp (dosvec)
251d        707:    ;-----
251d a901    708: loopinit: lda #1
251f 8df262 709:    sta more.
2522 8df002 710:    sta crsinh
2525 20113f 711:    jsr newscreen
2528 8ede62 712:    stx cioy
252b e8      713:    inx
252c 8e9261 714:    stx mymr
252f 8655    715:    stx colcrs
2531 e8      716:    inx
2532 8654    717:    stx rowcrs
2534 20372c 718:    jsr clr_printline
2537 ad2563 719:    lda mode
253a c946    720:    cmp #'F'
253c f014    721:    beq .notmem
253e a5c8    722:    lda s.org  ;Set up for
2540 85b2    723:    sta bufptr ;memory
2542 85c6    724:    sta pc    ;disassembly
2544 a5c9    725:    lda s.org+1
2546 85b3    726:    sta bufptr+1
2548 85c7    727:    sta pc+1
254a ad2563 728:    lda mode
254d c953    729:    cmp #'S'
254f f001    730:    beq .notmem
2551 60      731:    rts
2552        732:    ;
2552 20193f 733: .notmem: jsr chk_in
2555 ad2563 734:    lda mode
2558 c953    735:    cmp #'S'
255a d003    736:    bne .notsect
255c 4c782e 737:    jmp sector_block
255f        738:    ;
255f 4c4530 739: .notsect: jmp dofile
2562        740:    ;-----
2562        741:    ;
2562 ad2563 742: setbuffer: lda mode

```

```

2565 c94d 743:      cmp #'M'
2567 d009 744:      bne .dskdis
2569 a5c8 745:      lda s.org
256b 85b2 746:      sta bufptr
256d a5c9 747:      lda s.org+1
256f 85b3 748:      sta bufptr+1
2571 60 749:      rts
2572      750: ;
2572 18 751: .dskdis: clc
2573 ad1063 752:      lda labeltop
2576 6902 753:      adc #2
2578 85b0 754:      sta buffer
257a 85b4 755:      sta buftop
257c 85b2 756:      sta bufptr
257e ad1163 757:      lda labeltop+1
2581 6900 758:      adc #0
2583 85b1 759:      sta buffer+1
2585 85b5 760:      sta buftop+1
2587 85b3 761:      sta bufptr+1
2589 60 762:      rts
258a      763: ;-----
258a      764: ;
258a a907 765: resetvbi: lda #7
258c a2e4 766:      ldx # high xitvbv
258e a062 767:      ldy # low xitvbv
2590 4c5ce4 768:      jmp setvbv
2593      769: ;-----
2593      770:
2593      771:
2593      772: ; .PAGE "Label Routines"
2593      773: ; .include labl
2593      774:
2593      775: ; ... LABL ...
2593      776: ;
2593      777: ;-----
2593      778: ;
2593      779: ; Labeling Routines.
2593      780: ;
2593      781: ;-----
2593      782: ;
2593      783: ;This routine does a fake
2593      784: ;disassembly and saves the
2593      785: ;addresses that will require
2593      786: ;labels.
2593      787: ;
2593      788: build_symbol_tbl:
2593      789: ;
2593 20c229 790:      jsr get_a_byte ;Get a byte.
2596 b03f 791:      bcs tabl_built ;End of dis.
2598 8d0563 792:      sta opcode ;Save it.
259b aa 793:      tax
259c bd1860 794:      lda op.table,x ;See what
259f c9ff 795:      cmp #$ff ;it is.

```



```

25a1 f0f0 796:      beq build_symbol_tbl ;Don't
25a3 2960 797:      and #$60  ;need a label.
25a5 f0ec 798:      beq build_symbol_tbl
25a7 20c229 799:     jsr get_a_byte ;Get label
25aa b02b 800:      bcs tabl_built ;address.
25ac 85ba 801:      sta label.adr
25ae ae0563 802:     ldx opcode  ;See whether
25b1 bd1860 803:     lda op.table,x ;it is a
25b4 c926 804:      cmp #$26  ;branch
25b6 d006 805:      bne .24.  ;instruction.
25b8 20dd25 806:     jsr set_branch ;It is.
25bb 4cd225 807:     jmp .make_lbl
25be      808: ;
25be c924 809: .24.:  cmp #$24  ;Need a label?
25c0 f0d1 810:      beq build_symbol_tbl ;No.
25c2 2940 811:      and #$40  ;A zero-page adr?
25c4 d005 812:      bne .not_page0 ;No.
25c6 20743e 813:     jsr lbladrh ;Yes.
25c9 f007 814:      beq .make_lbl
25cb      815: ;
25cb 20c229 816: .not_page0: jsr get_a_byte ;Get
25ce b007 817:      bcs tabl_built ;2nd byte of
25d0 85bb 818:      sta label.adr+1 ;address.
25d2      819: ;
25d2 20f625 820: .make_lbl: jsr save_label
25d5 90bc 821:      bcc build_symbol_tbl
25d7      822: ;
25d7 adf262 823: tabl_built: lda more.
25da d0b7 824:      bne build_symbol_tbl
25dc 60 825:      rts
25dd      826: ;-----
25dd      827: ;
25dd      828: ;Figure out the target address
25dd      829: ;of a branch instruction.
25dd      830: ;
25dd      831: set_branch:
25dd      832: ;
25dd 20743e 833:     jsr lbladrh
25e0 a5ba 834:     lda label.adr
25e2 1004 835:     bpl .zpl  ;Forward branch.
25e4 a9ff 836:     lda #$ff  ;Backward branch.
25e6 85bb 837:     sta label.adr+1
25e8 18 838: .zpl:  clc
25e9 a5ba 839:     lda label.adr
25eb 65c6 840:     adc pc
25ed 85ba 841:     sta label.adr
25ef a5bb 842:     lda label.adr+1
25f1 65c7 843:     adc pc+1
25f3 85bb 844:     sta label.adr+1
25f5 60 845:      rts
25f6      846: ;-----
25f6      847: ;
25f6      848: ;Save label address in the

```

```
25f6      849: ;label address buffer.
25f6      850: ;
25f6      851: save_label:
25f6      852: ;
25f6 20d526 853:      jsr find_label ;Used already?
25f9 b001   854:      bcs .keep_it ;No.
25fb 60    855: .sxit: rts      ;Yes.
25fc      856: ;
25fc      857: ;
25fc 38    858: .keep_it: sec      ;Room for another
25fd ad0e63 859:      lda lbuf.size ;label?
2600 e9fe   860:      sbc # low bufadr-label.buffer-2
2602 ad0f63 861:      lda lbuf.size+1
2605 e915   862:      sbc # high bufadr-label.buffer-2
2607 b0f2   863:      bcs .sxit ;NO.
2609      864: ;
2609 20ff3d 865:      jsr setl.index ;Yes.
260c adec62 866:      lda labels
260f 0ded62 867:      ora labels+1
2612 d003   868:      bne .hilp
2614 4c9d26 869:      jmp .addit
2617      870: ;
2617 a001   871: .hilp: ldy #1      ;Find where to
2619 b1bc   872:      lda (label.ptr),y ;put it.
261b c5bb   873:      cmp label.adr+1
261d f004   874:      beq .findlo
261f b024   875:      bcs .there
2621 9007   876:      bcc .lookmor
2623      877: ;
2623 88     878: .findlo: dey
2624 b1bc   879:      lda (label.ptr),y
2626 c5ba   880:      cmp label.adr
2628 b01b   881:      bcs .there
262a 18     882: .lookmor: clc
262b a5bc   883:      lda label.ptr
262d 6902   884:      adc #2
262f 85bc   885:      sta label.ptr
2631 a5bd   886:      lda label.ptr+1
2633 6900   887:      adc #0
2635 85bd   888:      sta label.ptr+1
2637 ad1163 889:      lda labeltop+1
263a c5bd   890:      cmp label.ptr+1
263c d0d9   891:      bne .hilp
263e ad1063 892:      lda labeltop
2641 c5bc   893:      cmp label.ptr
2643 d0d2   894:      bne .hilp
2645      895: ;
2645 a5bd   896: .there: lda label.ptr+1
2647 cd1163 897:      cmp labeltop+1
264a d00a   898:      bne .svit
264c a5bc   899:      lda label.ptr
264e cd1063 900:      cmp labeltop
2651 d003   901:      bne .svit
```

```
2653 4c9d26 902:      jmp .addit
2656      903:      ;
2656 a5bc 904: .svit:  lda label.ptr
2658 85ca 905:      sta number
265a a5bd 906:      lda label.ptr+1
265c 85cb 907:      sta number+1
265e 18 908:      clc
265f ad1063 909:      lda labeltop
2662 85bc 910:      sta label.ptr
2664 6902 911:      adc #2
2666 85c4 912:      sta pointr
2668 ad1163 913:      lda labeltop+1
266b 85bd 914:      sta label.ptr+1
266d 6900 915:      adc #0
266f 85c5 916:      sta pointr+1
2671 a000 917:      ldy #0
2673 adec62 918:      lda labels
2676 0ded62 919:      ora labels+1
2679 f022 920:      beq .addit
267b      921:      ;
267b a5c4 922: .bmplbl:  lda pointr ;Make room
267d d002 923:      bne .pointrlo ;for this
267f c6c5 924:      dec pointr+1 ;label adr.
2681 c6c4 925: .pointrlo: dec pointr
2683 a5bc 926:      lda label.ptr
2685 d002 927:      bne .lplo
2687 c6bd 928:      dec label.ptr+1
2689 c6bc 929: .lplo:   dec label.ptr
268b a000 930:      ldy #0
268d b1bc 931:      lda (label.ptr),y
268f 91c4 932:      sta (pointr),y
2691 a5bc 933:      lda label.ptr
2693 c5ca 934:      cmp number
2695 d0e4 935:      bne .bmplbl
2697 a5bd 936:      lda label.ptr+1
2699 c5cb 937:      cmp number+1
269b d0de 938:      bne .bmplbl
269d      939:      ;
269d a000 940: .addit:  ldy #0      ;Store this
269f a5ba 941:      lda label.adr ;label adr.
26a1 91bc 942:      sta (label.ptr),y
26a3 c8 943:      iny
26a4 a5bb 944:      lda label.adr+1
26a6 91bc 945:      sta (label.ptr),y
26a8 18 946:      clc
26a9 ad1063 947:      lda labeltop ;Add 2 to
26ac 6902 948:      adc #2      ;top of label
26ae 8d1063 949:      sta labeltop ;buffer pointer.
26b1 ad1163 950:      lda labeltop+1
26b4 6900 951:      adc #0
26b6 8d1163 952:      sta labeltop+1
26b9 18 953:      clc
26ba ad0e63 954:      lda lbuf.size ;Add 2 to
```

```

26bd 6902 955:      adc #2      ;label buffer
26bf 8d0e63 956:      sta lbuf.size ;size.
26c2 ad0f63 957:      lda lbuf.size+1
26c5 6900 958:      adc #0
26c7 8d0f63 959:      sta lbuf.size+1
26ca      960:      inw labels ;Inc Label counter.
26ca eec62 ----:      inc labels
26cd d003 ----:      bne ~a
26cf eed62 ----:      inc labels+1
26d2 ----: ~a:
26d2 4c753b 961:      jmp ir_awake
26d5      962: ;-----
26d5      963: ;
26d5      964: ;Check to see if we have
26d5      965: ;saved this label address.
26d5      966: ;
26d5      967: find_label:
26d5      968: ;
26d5 20ff3d 969:      jsr setl.index
26d8 a900 970:      lda #0      ;0 indicates
26da 8dee62 971:      sta good.label ;not found.
26dd a001 972: .find_lbl: ldy #1 ;Check adr hi
26df 20083e 973:      jsr c.ladr
26e2 d01c 974:      bne .snot_here ;Not equal.
26e4 88 975:      dey
26e5 20083e 976:      jsr c.ladr ;Check adr lo
26e8 d016 977:      bne .snot_here ;Not equal
26ea eeee62 978:      inc good.label
26ed 38 979:      sec      ;Found it!
26ee a5bc 980:      lda label.ptr ;Get label
26f0 e95e 981:      sbc # low label.buffer ;#*2.
26f2 85be 982:      sta this.lbl
26f4 a5bd 983:      lda label.ptr+1
26f6 e967 984:      sbc # high label.buffer
26f8 85bf 985:      sta this.lbl+1
26fa 46bf 986:      lsr this.lbl+1 ;Divide by 2.
26fc 66be 987:      ror this.lbl
26fe 18 988:      clc      ;Indicate found.
26ff 60 989:      rts
2700      990: ;
2700 18 991: .snot_here: clc      ;Point to next
2701 a5bc 992:      lda label.ptr ;label
2703 6902 993:      adc #2
2705 85bc 994:      sta label.ptr
2707 a5bd 995:      lda label.ptr+1
2709 6900 996:      adc #0
270b 85bd 997:      sta label.ptr+1
270d 38 998:      sec
270e a5bc 999:      lda label.ptr ;See if we
2710 ed1063 1000:      sbc labeltop ;are at the
2713 a5bd 1001:      lda label.ptr+1 ;top of
2715 ed1163 1002:      sbc labeltop+1 ;the label
2718 90c3 1003:      bcc .find_lbl ;buffer yet.

```

```

271a 60 1004:      rts
271b      1005: ;-----
271b      1006: ;
271b      1007: ;Move Label name to printline.
271b      1008: ;
271b a20b 1009: label11: ldx #11      ;Start in col 11.
271d      1010: get_label:
271d      1011: ;
271d 8a 1012:      txa      ;Save column.
271e 48 1013:      pha
271f 20a127 1014:      jsr chekfor_syslbl ;ATARI label?
2722 b008 1015:      bcs .aint_sys ;No.
2724 206628 1016:      jsr find_syslabel ;Yes, find it.
2727 68 1017:      pla      ;Restore column.
2728 aa 1018:      tax
2729 4cbb28 1019:      jmp get_syslabel
272c      1020: ;
272c 20d526 1021: .aint_sys: jsr find_label
272f 68 1022:      pla      ;Restore column.
2730 aa 1023:      tax
2731      1024: ;-----
2731      1025: ;
2731      1026: ;See if we have a good label
2731      1027: ;number. If so move label
2731      1028: ;name to printline. Else
2731      1029: ;move actual address to
2731      1030: ;printline in hex.
2731      1031: ;
2731      1032: labelize:
2731      1033: ;
2731 adee62 1034:      lda good.label
2734 d012 1035:      bne .good_lbl
2736 a924 1036:      lda #'$'      ;Add address to
2738 9d3e63 1037:      sta printline,x ;printline
273b e8 1038:      inx      ;in Hex because
273c a001 1039:      ldy #1      ;there is no
273e b9ba00 1040: .nolblp: lda label.adr,y ;label
2741 20fc35 1041:      jsr byte_to_hex ;for it.
2744 88 1042:      dey
2745 10f7 1043:      bpl .nolblp
2747 60 1044:      rts
2748      1045: ;-----
2748      1046: ;
2748      1047: ;We have a good label number
2748      1048: ;so move label name to print-
2748      1049: ;line. First check to see if
2748      1050: ;it is going in the label field
2748      1051: ;and if so unmark it so it will
2748      1052: ;not be included in equate file.
2748      1053: ;
2748      1054: .good_lbl:
2748      1055: ;
2748 e000 1056:      cpx #0      ;If in column 0

```

```

274a d010 1057:    bne .noflip ;then unmark
274c 209c29 1058:    jsr get_lbit ;it so as not
274f 3d1861 1059:    and bit.table,x ;to write
2752 f003 1060:    beq .un_it ;an equate for
2754 a207 1061:    ldx #7    ;this label.
2756 60 1062:    rts
2757      1063:    ;
2757 209229 1064: .un_it: jsr unmark_label
275a a200 1065:    ldx #0
275c 20803e 1066: .noflip: jsr point
275f f01d 1067:    beq .czp ;If ZP adr.
2761 c9a0 1068:    cmp #$a0
2763 b014 1069:    bcs .nomrk
2765 ad9f61 1070:    lda marker ;Move label marker
2768 9d3e63 1071:    sta printline,x ;first.
276b ada061 1072:    lda marker+1
276e c99b 1073:    cmp #cr
2770 f004 1074:    beq .csk
2772 e8 1075:    inx
2773 9d3e63 1076:    sta printline,x
2776 e8 1077: .csk:  inx    ;Move hi byte
2777 b1bc 1078:    lda (label.ptr),y ;of adr
2779 20fc35 1079: .nomrk: jsr byte_to_hex ;to printline.
277c d01d 1080:    bne .cnzp
277e 8cdf62 1081: .czp:  sty zp.flag ;Show that ZP labels exist.
2781 88 1082:    dey
2782 b1bc 1083:    lda (label.ptr),y
2784 c8 1084:    iny
2785 c9a0 1085:    cmp #$a0
2787 b012 1086:    bcs .cnzp
2789 ada261 1087:    lda zmarker
278c 9d3e63 1088:    sta printline,x ;page label
278f e8 1089:    inx    ;so move Zmarker
2790 ada361 1090:    lda zmarker+1 ;to printline.
2793 c99b 1091:    cmp #cr
2795 f004 1092:    beq .cnzp
2797 9d3e63 1093:    sta printline,x
279a e8 1094:    inx
279b 88 1095: .cnzp:  dey    ;Move lo byte
279c b1bc 1096:    lda (label.ptr),y ;of adr to
279e 4cfc35 1097:    jmp byte_to_hex ;printline.
27a1      1098:    ;-----
27a1      1099:    ;
27a1      1100:    ;Check to see if this requires
27a1      1101:    ;an ATARI label.
27a1      1102:    ;
27a1      1103:    chekfor_syslbl:
27a1      1104:    ;
27a1 a200 1105:    ldx #0
27a3 ec165e 1106: .tblsrch: cpx tsize ;Past top of
27a6 b045 1107:    bcs mysys. ;ATARI labels? Yes.
27a8 a000 1108:    ldy #0    ;No. Check next page.
27aa bde45d 1109: .tblmv:  lda page.table,x ;Copy

```

```

27ad 991663 1110:      sta page.loc,y ;data for this
27b0 e8 1111:      inx      ;page to page.loc.
27b1 c8 1112:      iny
27b2 c005 1113:      cpy #5
27b4 d0f4 1114:      bne .tblmv
27b6 ad1863 1115:      lda page      ;Address in this
27b9 c5bb 1116:      cmp label.adr+1 ;page.
27bb d0e6 1117:      bne .tblsrch  ;No.
27bd ad1963 1118:      lda page.top  ;Yes. Is it
27c0 c5ba 1119:      cmp label.adr      ;included here?
27c2 9029 1120:      bcc mysys.      ;No.
27c4 ad1a63 1121:      lda rw.flag      ;Two labels?
27c7 301d 1122:      bmi .found_adr ;No.
27c9 ad0563 1123:      lda opcode ;Yes, see whether
27cc 29e7 1124:      and #$e7 ;to use Read label
27ce c986 1125:      cmp #$86 ;or Write label.
27d0 f00d 1126:      beq .write
27d2 c984 1127:      cmp #$84
27d4 f009 1128:      beq .write
27d6 29e3 1129:      and #$e3
27d8 c981 1130:      cmp #$81
27da f003 1131:      beq .write
27dc a901 1132:      lda #1 ;Use Read label.
27de 2c 1133:      dc.b $2c ;Dummy BIT.
27df a900 1134:      .write: lda #0 ;Use Write label.
27e1 cd1a63 1135:      .ovr0: cmp rw.flag ;Right?
27e4 d0bd 1136:      bne .tblsrch ;Nope, Look again.
27e6 a900 1137:      .found_adr: lda #0 ;Indicate not
27e8 8de662 1138:      sta myflag ;from misc table.
27eb 18 1139:      clc
27ec 60 1140:      rts
27ed 1141: ;
27ed 1142: ;
27ed a000 1143:      mysys.: ldy #0 ;Is this adr in the
27ef a5bb 1144:      lda label.adr+1 ;misc table?
27f1 f028 1145:      beq .p0 ;Check page 0.
27f3 c903 1146:      cmp #3
27f5 f036 1147:      beq .p3 ;Check page 3.
27f7 c9d0 1148:      cmp #$d0 ;Between pages
27f9 9055 1149:      bcc .nope ;D0-DF?
27fb c9df 1150:      cmp #$df
27fd b051 1151:      bcs .nope
27ff 1152: ;
27ff b92f5e 1153:      .fploop: lda fp.adr1st+1,y ;Is it
2802 c5bb 1154:      cmp label.adr+1 ;a Floating
2804 d00d 1155:      bne .fpnxt ;point adr?
2806 b92e5e 1156:      lda fp.adr1st,y
2809 c5ba 1157:      cmp label.adr
280b d006 1158:      bne .fpnxt
280d 98 1159:      tya ;Yes, divide Y by
280e 4a 1160:      lsr ;two.
280f a8 1161:      tay
2810 4c5e28 1162:      jmp .gotim

```

```

2813      1163: ;
2813 c8    1164: .fpnxt: iny      ;Point to next
2814 c8    1165:      iny      ;FP label.
2815 c042   1166:      cpy #66  ;Past end of table?
2817 d0e6   1167:      bne .fploop ;No.
2819 f035   1168:      beq .nope  ;Yes.
281b      1169: ;
281b a917   1170: .p0:   lda # low p0.adrlst ;Set up to
281d 8d4428 1171:      sta .findit+1 ;search page
2820 a95e   1172:      lda # high p0.adrlst ;0 misc.
2822 8d4528 1173:      sta .findit+2
2825 a90d   1174:      lda #13
2827 85cd   1175:      sta ytemp
2829 a900   1176:      lda #0
282b f015   1177:      beq .savea
282d      1178: ;
282d ad9961 1179: .p3:   lda xe      ;XL/XE labels?
2830 f01e   1180:      beq .nope  ;No, skip it.
2832 a923   1181:      lda # low p3.adrlst ;Yes, set up
2834 8d4428 1182:      sta .findit+1 ;to search page
2837 a95e   1183:      lda # high p3.adrlst ;3 misc.
2839 8d4528 1184:      sta .findit+2
283c a90c   1185:      lda #12
283e 85cd   1186:      sta ytemp
2840 a903   1187:      lda #3
2842      1188: ;
2842 48     1189: .savea: pha
2843 b9ffff 1190: .findit: lda dummy,y ;Search for
2846 c5ba   1191:      cmp label.adr ;match of lo byte.
2848 f008   1192:      beq .hirit  ;Gotcha!
284a c8     1193:      iny
284b c4cd   1194:      cpy ytemp  ;Top of table?
284d d0f4   1195:      bne .findit ;No, look more.
284f 68     1196: .noway: pla      ;No match found.
2850 38     1197: .nope:  sec
2851 60     1198:      rts
2852      1199: ;
2852 68     1200: .hirit: pla      ;Double check hi byte
2853 c5bb   1201:      cmp label.adr+1 ;of adr.
2855 f007   1202:      beq .gotim  ;it's a Match.
2857 c8     1203:      iny      ;Does'nt match.
2858 c4cd   1204:      cpy ytemp
285a d0e6   1205:      bne .savea
285c f0f2   1206:      beq .nope
285e      1207: ;
285e 84cd   1208: .gotim: sty ytemp ;It's a good
2860 c8     1209:      iny      ;label.
2861 8ce662 1210:      sty myflag ;Indicate it is
2864 18     1211:      clc      ;a misc label.
2865 60     1212:      rts
2866      1213: ;-----
2866      1214: ;
2866      1215: ;Find address of ATARI

```



```
2866      1216: ;label name.
2866      1217: ;
2866      1218: find_syslabel:
2866      1219: ;
2866 ade662 1220:      lda myflag ;Misc label?
2869 d00e 1221:      bne find_mysys ;Yes.
286b 20422c 1222:      jsr reset_ptrs ;Point to label.
286e a000 1223:      ldy #0
2870 200e3e 1224:      jsr andspace ;Need a '+'?
2873 d003 1225:      bne .found_lbl ;No.
2875 20283e 1226:      jsr backloop ;Yes, point to real label.
2878      1227: ;
2878 60 1228: .found_lbl: rts
2879      1229: ;
2879      1230: ;
2879      1231: find_mysys: ;      Find Misc ATARI
2879 a5bb 1232:      lda label.adr+1 ;label.
287b c9d0 1233:      cmp #$d0
287d 900a 1234:      bcc .0adr
287f a91e 1235:      lda # low fp.labels
2881 85b8 1236:      sta syslabel
2883 a95d 1237:      lda # high fp.labels
2885 85b9 1238:      sta syslabel+1
2887 d015 1239:      bne .gotmyhi
2889      1240: ;
2889 a8 1241: .0adr:  tay      ;0 page?
288a d00a 1242:      bne .lo_adr ;No.
288c a994 1243:      lda # low p0.labels ;Yes, point
288e 85b8 1244:      sta syslabel ;to 0 page
2890 a95c 1245:      lda # high p0.labels ;labels.
2892 85b9 1246:      sta syslabel+1
2894 d008 1247:      bne .gotmyhi
2896      1248: ;
2896 a9dc 1249: .lo_adr: lda # low p3.labels ;Point
2898 85b8 1250:      sta syslabel ;to page 3
289a a95c 1251:      lda # high p3.labels ;labels.
289c 85b9 1252:      sta syslabel+1
289e      1253: ;
289e a4cd 1254: .gotmyhi: ldy ytemp
28a0 f006 1255:      beq .mysysx
28a2 201a3e 1256: .setploop: jsr bumps1.6
28a5 88 1257:      dey
28a6 d0fa 1258:      bne .setploop
28a8      1259: ;
28a8 a000 1260: .mysysx: ldy #0
28aa 8c2863 1261:      sty offset
28ad 20153e 1262:      jsr and7f
28b0 c92b 1263:      cmp #'+'
28b2 d006 1264:      bne .mysysfound
28b4 8d2863 1265:      sta offset
28b7 203e3e 1266:      jsr backup
28ba      1267: ;
28ba 60 1268: .mysysfound: rts
```

```

28bb 1269: ;-----
28bb 1270: ;
28bb 1271: ;Move system label name
28bb 1272: ;to printline.
28bb 1273: ;
28bb 1274: get_syslabel:
28bb 8a 1275: txa ;Save printline index.
28bc 48 1276: pha
28bd a95e 1277: lda # low label.buffer
28bf 85c0 1278: sta this.lptr
28c1 a967 1279: lda # high label.buffer
28c3 85c1 1280: sta this.lptr+1
28c5 204c3e 1281: jsr zthis.lbl
28c8 a001 1282: ldy #1
28ca b9ba00 1283: .mhilp: lda label.adr,y
28cd d1c0 1284: cmp (this.lptr),y
28cf f012 1285: beq .mlo
28d1 20793e 1286: jsr inths.lbl
28d4 18 1287: clc
28d5 a5c0 1288: lda this.lptr
28d7 6902 1289: adc #2
28d9 85c0 1290: sta this.lptr
28db a5c1 1291: lda this.lptr+1
28dd 6900 1292: adc #0
28df 85c1 1293: sta this.lptr+1
28e1 d0e7 1294: bne .mhilp
28e3 1295: ;
28e3 88 1296: .mlo: dey
28e4 10e4 1297: bpl .mhilp
28e6 1298: ;
28e6 209229 1299: .mrkit: jsr unmark_label
28e9 68 1300: pla
28ea aa 1301: tax
28eb ade662 1302: .skflp: lda myflag
28ee f003 1303: beq .nmsys
28f0 4c5f29 1304: jmp get_mysys
28f3 a900 1305: .nmsys: lda #0
28f5 8d2863 1306: sta offset
28f8 20422c 1307: jsr reset_ptrs
28fb a001 1308: ldy #1
28fd e000 1309: cpx #0
28ff d003 1310: bne .oprnd
2901 20533e 1311: jsr ora80
2904 88 1312: .oprnd: dey
2905 200e3e 1313: jsr andspace
2908 d003 1314: bne .whatsx
290a 20283e 1315: jsr backloop
290d e000 1316: .whatsx: cpx #0
290f d005 1317: bne .movem
2911 ad2863 1318: lda offset
2914 d047 1319: bne .moved
2916 a000 1320: .movem: ldy #0
2918 20533e 1321: jsr ora80

```

```
291b 297f 1322:    and #$7f
291d 9d3e63 1323:    sta printline,x
2920      1324:    ;
2920 e8      1325: .movemlp: inx
2921 c8      1326:    iny
2922 200e3e 1327:    jsr andspace
2925 f008 1328:    beq .chkoffset
2927 9d3e63 1329:    sta printline,x
292a c005 1330:    cpy #5
292c 90f2 1331:    bcc .movemlp
292e e8      1332:    inx
292f ad2863 1333: .chkoffset: lda offset
2932 f029 1334:    beq .moved
2934 e007 1335:    cpx #7
2936 9025 1336:    bcc .moved
2938 a92b 1337:    lda #'+'
293a 9d3e63 1338:    sta printline,x
293d e8      1339:    inx
293e 8a      1340:    txa
293f 48      1341:    pha
2940 ad2863 1342:    lda offset
2943 203137 1343:    jsr decimal
2946 68      1344:    pla
2947 aa      1345:    tax
2948 a000 1346:    ldy #0
294a b1f3 1347: .cdecpl: lda (inbuff),y
294c 3009 1348:    bmi .lstdc
294e 9d3e63 1349:    sta printline,x
2951 e8      1350:    inx
2952 c8      1351:    iny
2953 c002 1352:    cpy #2
2955 d0f3 1353:    bne .cdecpl
2957 297f 1354: .lstdc: and #$7f
2959 9d3e63 1355:    sta printline,x
295c e8      1356:    inx
295d 18      1357: .moved: clc
295e 60      1358:    rts
295f      1359:    ;
295f      1360:    ;
295f      1361: get_mysys:
295f      1362:    ;
295f a000 1363:    ldy #0
2961 20533e 1364:    jsr ora80
2964 c8      1365:    iny
2965 e006 1366:    cpx #6
2967 b003 1367:    bcs .moprnd
2969 20533e 1368:    jsr ora80
296c      1369:    ;
296c 88      1370: .moprnd: dey
296d 200e3e 1371: .movmyslp: jsr andspace
2970 f009 1372:    beq .chkmoff
2972 9d3e63 1373:    sta printline,x
2975 e8      1374:    inx
```

```

2976 c8 1375:    iny
2977 c006 1376:    cpy #6
2979 d0f2 1377:    bne .movmyslp
297b e007 1378:    .chkmoff: cpx #7
297d 9011 1379:    bcc .gotmsys
297f ad2863 1380:    lda offset
2982 c92b 1381:    cmp #'+'
2984 d00a 1382:    bne .gotmsys
2986 9d3e63 1383:    sta printline,x
2989 a931 1384:    lda #'l'
298b e8 1385:    inx
298c 9d3e63 1386:    sta printline,x
298f e8 1387:    inx
2990 18 1388:    .gotmsys: clc
2991 60 1389:    rts
2992 1390: ;-----
2992 1391: ;
2992 1392: unmark_label: ;    Unmark label
2992 1393: ;
2992 209c29 1394:    jsr get_lbit ;so we do not
2995 1d1861 1395:    ora bit.table,x ;write an
2998 91c0 1396:    sta (this.lptr),y ;equate
299a 18 1397:    clc    ;for this label.
299b 60 1398:    rts
299c 1399: ;-----
299c 1400: ;
299c 1401: ;Labels are marked in a bitmap
299c 1402: ;similar to the bitmap sector
299c 1403: ;of a DOS disk. This routine
299c 1404: ;finds the bit in the bitmap
299c 1405: ;Corresponding to the current
299c 1406: ;label.
299c 1407: ;
299c 1408: get_lbit:
299c 1409: ;
299c a5be 1410:    lda this.lbl
299e 2907 1411:    and #7
29a0 aa 1412:    tax
29a1 a5be 1413:    lda this.lbl
29a3 85c0 1414:    sta this.lptr
29a5 a5bf 1415:    lda this.lbl+1
29a7 85c1 1416:    sta this.lptr+1
29a9 a003 1417:    ldy #3
29ab 46c1 1418:    .divide: lsr this.lptr+1
29ad 66c0 1419:    ror this.lptr
29af 88 1420:    dey
29b0 d0f9 1421:    bne .divide
29b2 18 1422:    clc
29b3 a95e 1423:    lda # low label.map
29b5 65c0 1424:    adc this.lptr
29b7 85c0 1425:    sta this.lptr
29b9 a964 1426:    lda # high label.map
29bb 65c1 1427:    adc this.lptr+1

```

```
29bd 85c1 1428:      sta this.lptr+1
29bf b1c0 1429:      lda (this.lptr),y
29c1 60 1430:      rts
29c2 1431: ;-----
29c2 1432:
29c2 1433:
29c2 1434:
29c2 1435: ; .PAGE "Disassembly Routines"
29c2 1436: ; .include diss
29c2 1437: ; ... DISS ...
29c2 206a2c 1438: get_a_byte: jsr chkbuf
29c5 4cffff 1439: modejump: jmp dummy
29c8 1440: ;
29c8 1441: ;-----
29c8 1442: ;
29c8 1443: ;
29c8 1444: disassemble:
29c8 1445: ;
29c8 20242c 1446:      jsr prep
29cb 1447: ;
29cb 1448: ;Compute current address
29cb 1449: ;and store it in LABEL.ADR.
29cb 1450: ;
29cb 20923c 1451:      jsr compute_address
29ce 20973d 1452:      jsr setprogc
29d1 1453: ;
29d1 1454: ;See if a label goes here.
29d1 1455: ;
29d1 20a127 1456:      jsr chekfor_syslbl
29d4 b00a 1457:      bcs .chklbl
29d6 206628 1458:      jsr find_syslabel
29d9 a200 1459:      ldx #0
29db 20bb28 1460:      jsr get_syslabel
29de 900a 1461:      bcc .ovrlbl
29e0 1462: ;
29e0 20d526 1463: .chklbl: jsr find_label
29e3 b005 1464:      bcs .ovrlbl
29e5 1465: ;
29e5 1466: ;One does so put it here.
29e5 1467: ;
29e5 a200 1468:      ldx #0
29e7 203127 1469:      jsr labelize
29ea 1470: ;
29ea 1471: ;Go disassemble what we got.
29ea 1472: ;
29ea 20c229 1473: .ovrlbl: jsr get_a_byte
29ed 9003 1474:      bcc .get_opratr
29ef 4ca62a 1475:      jmp .go_to_byte
29f2 1476: ;
29f2 aa 1477: .get_opratr: tax
29f3 bd1860 1478:      lda op.table,x
29f6 c9ff 1479:      cmp #$ff
29f8 d003 1480:      bne .notyet
```

```
29fa 4c7d2a 1481:    jmp .get_oprand
29fd      1482:    ;
29fd ac2d63 1483: .notyet: ldy byts.this.op
2a00 f01a  1484:    beq .dis.2
2a02 8e1563 1485:    stx xtemp
2a05 20cd2a 1486:    jsr byte
2a08 a5c6  1487:    lda pc
2a0a d002  1488:    bne .A1
2a0c c6c7  1489:    dec pc+1
2a0e c6c6  1490: .A1:   dec pc
2a10 20923c 1491:    jsr compute_address
2a13 20c63e 1492:    jsr inw.pcountr
2a16      1493:    ;
2a16 20242c 1494: .dis.1: jsr prep
2a19 ae1563 1495:    ldx xtemp
2a1c      1496:    ;
2a1c 8e0563 1497: .dis.2: stx opcode
2a1f ee0d63 1498:    inc bytes.done
2a22 bd1860 1499:    lda op.table,x
2a25 2960   1500:    and #$60
2a27 4a     1501:    lsr
2a28 4a     1502:    lsr
2a29 4a     1503:    lsr
2a2a 4a     1504:    lsr
2a2b 4a     1505:    lsr
2a2c 8d1463 1506:    sta bytes.to.do ;will be 0,1,2, or 3.
2a2f 2902   1507:    and #2
2a31 8df462 1508:    sta op.temp
2a34 ad1463 1509:    lda bytes.to.do
2a37      1510:    ;
2a37 f014   1511: .dis.3: beq .dis.4
2a39 20c229 1512:    jsr get_a_byte
2a3c b062   1513:    bcs .sav_done
2a3e ae0d63 1514:    ldx bytes.done
2a41 9d0563 1515:    sta opcode,x
2a44 ee0d63 1516:    inc bytes.done
2a47 ce1463 1517:    dec bytes.to.do
2a4a 4c372a 1518:    jmp .dis.3
2a4d      1519:    ;
2a4d adf462 1520: .dis.4: lda op.temp
2a50 f00d   1521:    beq .monicize
2a52 ad0763 1522:    lda operand+1
2a55 d008   1523:    bne .monicize
2a57 a903   1524:    lda #3
2a59 8d2d63 1525:    sta byts.this.op
2a5c 4c8f2a 1526:    jmp .do_a_byte
2a5f      1527:    ;
2a5f 20132b 1528: .monicize: jsr get_mnemonic
2a62 20ee38 1529:    jsr write_line
2a65 ae0563 1530:    ldx opcode
2a68 bd185f 1531:    lda mnem.pointr,x
2a6b c953   1532:    cmp #$53
2a6d f008   1533:    beq .brkit
```

```

2a6f c980 1534:      cmp #$80
2a71 f004 1535:      beq .brkit
2a73 c97d 1536:      cmp #$7d
2a75 d003 1537:      bne .disj
2a77      1538: ;
2a77 201d3c 1539: .brkit: jsr onespace
2a7a 4cc829 1540: .disj:  jmp disassemble
2a7d      1541: ;
2a7d 8a    1542: .get_oprand: txa
2a7e ae2d63 1543:      ldx byts.this.op
2a81 9d0563 1544:      sta opcode,x
2a84 ee2d63 1545:      inc byts.this.op
2a87 20973d 1546:      jsr setprogc
2a8a 20d526 1547:      jsr find_label
2a8d b006 1548:      bcs .dis.8
2a8f      1549: ;
2a8f 20cd2a 1550: .do_a_byte: jsr byte
2a92 4cc829 1551:      jmp disassemble
2a95      1552: ;
2a95 ad2d63 1553: .dis.8: lda byts.this.op
2a98 cd9a61 1554:      cmp bytes
2a9b b0f2 1555:      bcs .do_a_byte
2a9d 4cea29 1556:      jmp .ovrlbl
2aa0      1557: ;
2aa0 ae0d63 1558: .sav_done: ldx bytes.done
2aa3 8e2d63 1559:      stx byts.this.op
2aa6      1560: ;
2aa6 ad2d63 1561: .go_to_byte: lda byts.this.op
2aa9 f003 1562:      beq .mortodo.
2aab 20cd2a 1563:      jsr byte
2aae      1564: ;
2aae adf262 1565: .mortodo.: lda more.
2ab1 f011 1566:      beq .dis_exit
2ab3 ad2663 1567:      lda lastorg
2ab6 c5c6 1568:      cmp pc
2ab8 d007 1569:      bne .do_org
2aba ad2763 1570:      lda lastorg+1
2abd c5c7 1571:      cmp pc+1
2abf f003 1572:      beq .dis_exit
2ac1 20b93c 1573: .do_org: jsr sho_org
2ac4      1574: ;
2ac4 adf262 1575: .dis_exit: lda more.
2ac7 f003 1576:      beq .dis_done
2ac9 4cc829 1577:      jmp disassemble
2acc      1578: ;
2acc 60    1579: .dis_done: rts
2acd      1580: ;-----
2acd      1581: ;
2acd      1582: byte:
2acd      1583: ;
2acd a97c 1584:      lda # low byte_m
2acf 8d453c 1585:      sta from+1
2ad2 a962 1586:      lda # high byte_m

```

```

2ad4 8d463c 1587:      sta from+2
2ad7 a207 1588:      ldx #7
2ad9 20423c 1589:      jsr move
2adc a000 1590:      ldy #0
2ade      1591: ;
2ade a924 1592: .storbyte: lda #'$'
2ae0 9d3e63 1593:      sta printline,x
2ae3 e8 1594:      inx
2ae4 b90563 1595:      lda opcode,y
2ae7 20fc35 1596:      jsr byte_to_hex
2aea a92c 1597:      lda #','
2aec 9d3e63 1598:      sta printline,x
2aef e8 1599:      inx
2af0 c8 1600:      iny
2af1 ce2d63 1601:      dec byts.this.op
2af4 d0e8 1602:      bne .storbyte
2af6 ca 1603:      dex
2af7 a99b 1604:      lda #cr
2af9 9d3e63 1605:      sta printline,x
2afc 20ee38 1606:      jsr write_line
2aff ad4c63 1607:      lda printline+14
2b02 c936 1608:      cmp #'6'
2b04 d00a 1609:      bne .jprp
2b06 ad4d63 1610:      lda printline+15
2b09 c943 1611:      cmp #'C'
2b0b d003 1612:      bne .jprp
2b0d 201d3c 1613:      jsr onespace
2b10 4c242c 1614: .jprp: jmp prep
2b13      1615: ;-----
2b13      1616: ;
2b13      1617: ;Move mnemonic to printline
2b13      1618: ;
2b13      1619: get_mnemonic:
2b13      1620: ;
2b13 ae0563 1621:      ldx opcode
2b16 bd185f 1622:      lda mnem.pointr,x
2b19 aa 1623:      tax
2b1a a002 1624:      ldy #2
2b1c bd705e 1625: .mov_instr: lda mnemonic.tbl,x
2b1f 994563 1626:      sta printline+7,y
2b22 ca 1627:      dex
2b23 88 1628:      dey
2b24 10f6 1629:      bpl .mov_instr
2b26 ae0563 1630:      ldx opcode ;Add addressing
2b29 bd1860 1631:      lda op.table,x ;mode.
2b2c 291f 1632:      and #$1f
2b2e aa 1633:      tax
2b2f bd2061 1634:      lda mode.tbl,x
2b32 8d3d2b 1635:      sta .arg_jump+1
2b35 e8 1636:      inx
2b36 bd2061 1637:      lda mode.tbl,x
2b39 8d3e2b 1638:      sta .arg_jump+2
2b3c 4cffff 1639: .arg_jump: jmp dummy

```



```
2b3f      1640: ;-----
2b3f      1641: ;
2b3f      1642: ;Routines to add the correct
2b3f      1643: ;addressing mode to the
2b3f      1644: ;output line.
2b3f      1645: ;
2b3f a20a  1646: reg:   ldx #10
2b41 4c122c 1647:      jmp Acr
2b44      1648: ;
2b44 a941  1649: acc:   lda #'A'
2b46 a20b  1650:      ldx #11
2b48 4c0e2c 1651:      jmp Achr
2b4b      1652: ;
2b4b a923  1653: imm:   lda #'#'
2b4d 8d4963 1654:      sta printline+11
2b50 a20c  1655:      ldx #12
2b52 ad0663 1656:      lda operand
2b55 c90a  1657:      cmp #10
2b57 b005  1658:      bcs .immh
2b59 0930  1659:      ora #'0'
2b5b 4c0e2c 1660:      jmp Achr
2b5e a924  1661: .immh: lda #'$'
2b60 9d3e63 1662:      sta printline,x
2b63 e8     1663:      inx
2b64 ad0663 1664:      lda operand
2b67 20fc35 1665:      jsr byte_to_hex
2b6a 4c122c 1666:      jmp Acr
2b6d      1667: ;
2b6d ad0663 1668: branch: lda operand
2b70 85ba  1669:      sta label.adr
2b72 20dd25 1670:      jsr set_branch
2b75 201b27 1671:      jsr label11
2b78 4c122c 1672:      jmp Acr
2b7b      1673: ;
2b7b 201e2c 1674: adress: jsr Aplbl
2b7e 4c122c 1675:      jmp Acr
2b81      1676: ;
2b81 201e2c 1677: x_indx: jsr Aplbl
2b84 a958  1678:      lda #'X'
2b86 4c062c 1679:      jmp Aopcr
2b89      1680: ;
2b89 201e2c 1681: y_indx: jsr Aplbl
2b8c a959  1682:      lda #'Y'
2b8e 4c062c 1683:      jmp Aopcr
2b91      1684: ;
2b91 a900  1685: zpage: lda #0
2b93 85bb  1686:      sta label.adr+1
2b95 ad0663 1687:      lda operand
2b98 85ba  1688:      sta label.adr
2b9a 201b27 1689:      jsr label11
2b9d 4c122c 1690:      jmp Acr
2ba0      1691: ;
2ba0 a900  1692: x_indx1: lda #0
```

```

2ba2 85bb 1693:    sta label.adr+1
2ba4 ad0663 1694:    lda operand
2ba7 85ba 1695:    sta label.adr
2ba9 201b27 1696:    jsr label11
2bac a958 1697:    lda #'X'
2bae 4c062c 1698:    jmp Aopcr
2bb1      1699:    ;
2bb1 a900 1700: yindx1: lda #0
2bb3 85bb 1701:    sta label.adr+1
2bb5 ad0663 1702:    lda operand
2bb8 85ba 1703:    sta label.adr
2bba 201b27 1704:    jsr label11
2bbd a959 1705:    lda #'Y'
2bbf 4c062c 1706:    jmp Aopcr
2bc2      1707:    ;
2bc2 a928 1708: ind_x:  lda #'('
2bc4 8d4963 1709:    sta printline+11
2bc7 208c3d 1710:    jsr optolbl
2bca 20182c 1711:    jsr Ainsrtlabl
2bcd a929 1712:    lda #')'
2bcf 4c0e2c 1713:    jmp Achr
2bd2      1714:    ;
2bd2 a928 1715: ind_x1: lda #'('
2bd4 8d4963 1716:    sta printline+11
2bd7 ad0663 1717:    lda operand
2bda 85ba 1718:    sta label.adr
2bdc a900 1719:    lda #0
2bde 85bb 1720:    sta label.adr+1
2be0 20182c 1721:    jsr Ainsrtlabl
2be3 a958 1722:    lda #'X'
2be5 20062c 1723:    jsr Aopcr
2be8 a929 1724:    lda #')'
2bea 4c0e2c 1725:    jmp Achr
2bed      1726:    ;
2bed a928 1727: ind_y:  lda #'('
2bef 8d4963 1728:    sta printline+11
2bf2 ad0663 1729:    lda operand
2bf5 85ba 1730:    sta label.adr
2bf7 a900 1731:    lda #0
2bf9 85bb 1732:    sta label.adr+1
2bfb 20182c 1733:    jsr Ainsrtlabl
2bfe a929 1734:    lda #')'
2c00 9d3e63 1735:    sta printline,x
2c03 e8 1736:    inx
2c04 a959 1737:    lda #'Y'
2c06      1738:    ;-----
2c06      1739:    ;
2c06 48 1740: Aopcr: pha
2c07 a92c 1741:    lda #','
2c09 9d3e63 1742:    sta printline,x
2c0c e8 1743:    inx
2c0d 68 1744:    pla
2c0e 9d3e63 1745: Achr:  sta printline,x

```

```

2c11 e8 1746:    inx
2c12 a99b 1747: Acr:  lda #cr
2c14 9d3e63 1748:    sta printline,x
2c17 60 1749:    rts
2c18      1750: ;-----
2c18      1751: ;
2c18 a20c 1752: Ainsrtlabl: ldx #12
2c1a 201d27 1753:    jsr get_label
2c1d 60 1754:    rts
2c1e      1755: ;
2c1e      1756: ;
2c1e 208c3d 1757: Aplbl:  jsr optolbl
2c21 4c1b27 1758:    jmp label11
2c24      1759: ;-----
2c24      1760: ;
2c24      1761: prep:
2c24      1762: ;
2c24 a900 1763:    lda #0
2c26 a207 1764:    ldx #7
2c28 9d0563 1765: .clrlp1: sta opcode,x
2c2b ca 1766:    dex
2c2c 10fa 1767:    bpl .clrlp1
2c2e 8d2d63 1768:    sta byts.this.op
2c31 8d0d63 1769:    sta bytes.done
2c34 8df462 1770:    sta op.temp
2c37      1771: ;
2c37      1772: clr_printline:
2c37      1773: ;
2c37 a920 1774:    lda #space
2c39 a227 1775:    ldx #39
2c3b 9d3e63 1776: .clrloop: sta printline,x
2c3e ca 1777:    dex
2c3f 10fa 1778:    bpl .clrloop
2c41 60 1779:    rts
2c42      1780: ;-----
2c42      1781: ;
2c42      1782: reset_ptrs:
2c42      1783: ;
2c42 a900 1784:    lda #0
2c44 85b8 1785:    sta syslabel
2c46 85b9 1786:    sta syslabel+1
2c48 a006 1787:    ldy #6
2c4a 18 1788: .adlupe: clc
2c4b a5b8 1789:    lda syslabel
2c4d 65ba 1790:    adc label.adr
2c4f 85b8 1791:    sta syslabel
2c51 a5b9 1792:    lda syslabel+1
2c53 6900 1793:    adc #0
2c55 85b9 1794:    sta syslabel+1
2c57 88 1795:    dey
2c58 d0f0 1796:    bne .adlupe
2c5a 18 1797:    clc
2c5b a5b8 1798:    lda syslabel

```

```
2c5d 6d1663 1799:    adc page.loc
2c60 85b8 1800:    sta syslabel
2c62 a5b9 1801:    lda syslabel+1
2c64 6d1763 1802:    adc page.loc+1
2c67 85b9 1803:    sta syslabel+1
2c69 60 1804:    rts
2c6a 1805: ;-----
2c6a 1806: ;
2c6a 1807: chkbuf:
2c6a 1808: ;
2c6a 38 1809:    sec
2c6b a5b4 1810:    lda buftop
2c6d e5b2 1811:    sbc bufptr
2c6f a5b5 1812:    lda buftop+1
2c71 e5b3 1813:    sbc bufptr+1
2c73 60 1814:    rts
2c74 1815: ;
2c74 1816: ;-----
2c74 1817: ;
2c74 b003 1818: memory_dis: bcs fetch_byte
2c76 4c7e3d 1819:    jmp no_more
2c79 1820: ;
2c79 1821: fetch_byte:
2c79 1822: ;
2c79 a000 1823:    ldy #0    ;Get next byte
2c7b b1b2 1824:    lda (bufptr),y ;to be
2c7d aa 1825:    tax    ;disassembled
2c7e 1826:    inw bufptr ;from buffer.
2c7e e6b2 ----:    inc  bufptr
2c80 d002 ----:    bne  ~a
2c82 e6b3 ----:    inc  bufptr+1
2c84 ----: ~a:
2c84 20c63e 1827:    jsr inw.pcount
2c87 8a 1828:    txa
2c88 18 1829:    clc
2c89 60 1830:    rts
2c8a 1831: ;-----
2c8a 1832:
2c8a 1833:
2c8a 1834: ; .PAGE "Sector Disassembly Routines"
2c8a 1835: ; .include sect
2c8a 1836:
2c8a 1837: ; ... SECT ...
2c8a 1838: ;
2c8a 1839: ; Routines used for
2c8a 1840: ; Disassembling Sectors.
2c8a 1841: ;
2c8a 1842: ;Point MODEJUMP to sector
2c8a 1843: ;disassembly routine.
2c8a 1844: ;
2c8a 1845: getsectors:
2c8a 1846: ;
2c8a a9e5 1847:    lda # low sector_dis
```

```
2c8c 8dc629 1848:      sta modejump+1
2c8f a92d  1849:      lda # high_sector_dis
2c91 8dc729 1850:      sta modejump+2
2c94 a944  1851:      lda #'D'
2c96 8d8e63 1852:      sta infn
2c99 adca61 1853:      lda dn+1
2c9c 8d8f63 1854:      sta infn+1
2c9f a900  1855:      lda #0
2ca1 8d1d63 1856:      sta sector
2ca4 8d1e63 1857:      sta sector+1
2ca7 8d2963 1858:      sta currnt.sectr
2caa 8d2a63 1859:      sta currnt.sectr+1
2cad      1860: ;
2cad      1861: ;Find out which sectors
2cad      1862: ;to disassemble.
2cad      1863: ;
2cad 208e2e 1864:      jsr zero_sector.map
2cb0 20113f 1865:      jsr newscreen
2cb3 a906  1866:      lda #6
2cb5 8552  1867:      sta lmarg
2cb7 8555  1868:      sta colcrs
2cb9      1869: ;
2cb9 208037 1870: a.get_first: jsr printsi
2cbc      1871: ;
2cbc 202044 1872:      dc.b " Disassemble W"
2ccb 686963 1873:      dc.b "hich Sectors"
2cd7 3f9b9b 1874:      dc.b "?",CR,CR,SPACE,EM
2cdc      1875: ;
2cdc      1876: ;
2cdc a920  1877: a.sector_in: lda #32
2cde 204f38 1878:      jsr getline
2ce1 c91b  1879:      cmp #esc
2ce3 f007  1880:      beq a.orsec
2ce5 ad6663 1881:      lda inline
2ce8 c99b  1882:      cmp #cr
2cea d00e  1883:      bne a.do_first
2cec ad1d63 1884: a.orsec:  lda sector
2cef 0d1e63 1885:      ora sector+1
2cf2 d003  1886:      bne a.stjmp
2cf4 4c6822 1887:      jmp menu
2cf7      1888: ;
2cf7 4c2840 1889: a.stjmp:  jmp get_start
2cfa      1890: ;
2cfa 203836 1891: a.do_first: jsr digitize
2cfd 9003  1892:      bcc a.first_ok
2cff 4cdf2d 1893:      jmp a.sectr_err
2d02      1894: ;
2d02 20c12d 1895: a.first_ok: jsr Achek_sec
2d05 9003  1896:      bcc a.fok
2d07 4cdf2d 1897:      jmp a.sectr_err
2d0a      1898: ;
2d0a a5ca  1899: a.fok:   lda number
2d0c 8d1d63 1900:      sta sector
```

```
2d0f a5cb 1901:    lda number+1
2d11 8d1e63 1902:    sta sector+1
2d14 20cb2d 1903:    jsr Asetbit
2d17      1904:    ;
2d17 aef362 1905: a.get_last: ldx xsave
2d1a bd6663 1906:    lda inline,x
2d1d c99b 1907:    cmp #cr
2d1f f055 1908:    beq a.moresecs.
2d21      1909:    ;
2d21 48 1910:    pha
2d22 a0ff 1911:    ldy #$ff
2d24      1912:    ;
2d24 c8 1913: a.lshloop: iny
2d25 e8 1914:    inx
2d26 bd6663 1915:    lda inline,x
2d29 996663 1916:    sta inline,y
2d2c c99b 1917:    cmp #cr
2d2e d0f4 1918:    bne a.lshloop
2d30 203836 1919:    jsr digitize
2d33 9004 1920:    bcc a.numok
2d35 68 1921:    pla
2d36 4cdf2d 1922:    jmp a.sectr_err
2d39      1923:    ;
2d39 20c12d 1924: a.numok: jsr Achek_sec
2d3c 9004 1925:    bcc a.num1
2d3e 68 1926:    pla
2d3f 4cdf2d 1927:    jmp a.sectr_err
2d42      1928:    ;
2d42 68 1929: a.num1: pla
2d43 c92c 1930:    cmp #','
2d45 d003 1931:    bne a.dash.
2d47 4c0a2d 1932:    jmp a.fok
2d4a      1933:    ;
2d4a c92d 1934: a.dash.: cmp #'-'
2d4c f003 1935:    beq a.dodash
2d4e 4cdf2d 1936:    jmp a.sectr_err
2d51      1937:    ;
2d51      1938:    ;
2d51 20b52d 1939: a.dodash: jsr Achekdiff
2d54 b003 1940:    bcs a.mark_loop
2d56 4cdf2d 1941:    jmp a.sectr_err
2d59      1942:    ;
2d59      1943: a.mark_loop: inw sector
2d59 ee1d63 ----:    inc sector
2d5c d003 ----:    bne ~a
2d5e ee1e63 ----:    inc sector+1
2d61 ----: ~a:
2d61 20b52d 1944:    jsr Achekdiff
2d64 b003 1945:    bcs a.diffok
2d66 4c172d 1946:    jmp a.get_last
2d69      1947:    ;
2d69 20c12d 1948: a.diffok: jsr Achek_sec
2d6c 9003 1949:    bcc a.sec_ok
```

```

2d6e 4cdf2d 1950:      jmp a.sectr_err
2d71      1951: ;
2d71 20cb2d 1952: a.sec_ok: jsr Asetbit
2d74 90e3  1953:      bcc a.mark_loop
2d76      1954: ;
2d76 a900  1955: a.moresecs.: lda #0
2d78 8df002 1956:      sta crsinh
2d7b 208037 1957:      jsr printsi
2d7e 9b9b20 1958:      dc.b cr,cr," Any More "
2d8a 536563 1959:      dc.b "Sectors? ",EM
2d94      1960: ;
2d94 20dc37 1961:      jsr getcap
2d97 c959  1962:      cmp #'Y'
2d99 f003  1963:      beq a.moreprmt
2d9b 4c2840 1964:      jmp get_start
2d9e 208037 1965: a.moreprmt: jsr printsi
2da1      1966: ;
2da1 9b9b57 1967:      dc.b cr,cr,"Which Ones"
2dad 3f209b 1968:      dc.b "? ",CR,CR,EM
2db2      1969: ;
2db2 4cdc2c 1970:      jmp a.sector_in
2db5      1971: ;
2db5      1972: ;
2db5 38    1973: Achekdiff: sec
2db6 a5ca  1974:      lda number
2db8 ed1d63 1975:      sbc sector
2dbb a5cb  1976:      lda number+1
2dbd ed1e63 1977:      sbc sector+1
2dc0 60    1978:      rts
2dc1      1979: ;
2dc1      1980: ;
2dc1 38    1981: Achek_sec: sec
2dc2 a5ca  1982:      lda number
2dc4 e9d1  1983:      sbc # low 721
2dc6 a5cb  1984:      lda number+1
2dc8 e902  1985:      sbc # high 721
2dca 60    1986:      rts
2dcb      1987: ;
2dcb ae1d63 1988: Asetbit: ldx sector
2dce ac1e63 1989:      ldy sector+1
2dd1 20992e 1990:      jsr get_sbit
2dd4 bdce63 1991:      lda sector.map,x
2dd7 191861 1992:      ora bit.table,y
2dda 9dce63 1993:      sta sector.map,x
2ddd 18    1994:      clc
2dde 60    1995:      rts
2ddf      1996: ;
2ddf 201c44 1997: a.sectr_err: jsr in_err
2de2 4cdc2c 1998:      jmp a.sector_in
2de5      1999: ;-----
2de5      2000: ;
2de5      2001: sector_dis:
2de5      2002: ;

```

```
2de5 b008 2003:      bcs a.nxt_sectr_byt
2de7 20f22d 2004:      jsr sectr_setup ;*
2dea 9003 2005:      bcc a.nxt_sectr_byt
2dec 4c7e3d 2006:      jmp no_more
2def      2007: ;
2def 4c792c 2008: a.nxt_sectr_byt: jmp fetch_byte
2df2      2009: ;
2df2 207f3d 2010: sectr_setup: jsr zmore ;*
2df5 20a73d 2011:      jsr setndx
2df8      2012: ;
2df8 20b93d 2013: a.sectr_chkbuf: jsr setbsize
2dfb 38 2014:      sec
2dfc a5b6 2015:      lda bufsize
2dfe e980 2016:      sbc #$80
2e00 a5b7 2017:      lda bufsize+1
2e02 e900 2018:      sbc #0
2e04 906f 2019:      bcc a.havsectors
2e06      2020: ;
2e06      2021: prep__nxtsectr: inw currnt.sectr
2e06 ee2963 ----:      inc currnt.sectr
2e09 d003 ----:      bne ~a
2e0b ee2a63 ----:      inc currnt.sectr+1
2e0e ----: ~a:
2e0e 38 2022:      sec
2e0f a9d0 2023:      lda # low maxsec ;SECTOR 720?
2e11 ed2963 2024:      sbc currnt.sectr
2e14 a902 2025:      lda # high maxsec
2e16 ed2a63 2026:      sbc currnt.sectr+1
2e19 905a 2027:      bcc a.havsectors
2e1b ae2963 2028:      ldx currnt.sectr
2e1e ac2a63 2029:      ldy currnt.sectr+1
2e21 20992e 2030:      jsr get_sbit
2e24 bdce63 2031:      lda sector.map,x
2e27 391861 2032:      and bit.table,y
2e2a f0da 2033:      beq prep__nxtsectr
2e2c 20193f 2034:      jsr chk_in
2e2f      2035: ;
2e2f ad2963 2036: a.read_nxtsectr: lda currnt.sectr
2e32 8d0a03 2037:      sta daux1
2e35 ad2a63 2038:      lda currnt.sectr+1
2e38 8d0b03 2039:      sta daux2
2e3b ad8f63 2040:      lda infn+1
2e3e 8d0003 2041:      sta ddevic
2e41 290f 2042:      and #$0f
2e43 8d0103 2043:      sta dunit
2e46 a952 2044:      lda #$52 ;READ SECTOR
2e48 8d0203 2045:      sta dcomnd
2e4b 18 2046:      clc
2e4c a5b4 2047:      lda buftop
2e4e 6901 2048:      adc #1
2e50 8d0403 2049:      sta dbuflo
2e53 a5b5 2050:      lda buftop+1
2e55 6900 2051:      adc #0
```



```
2e57 8d0503 2052:      sta dbufhi
2e5a 2053e4 2053:      jsr dskinv
2e5d ac0303 2054:      ldy dstats
2e60 1003   2055:      bpl a.goodread
2e62 4cca43 2056:      jmp io_err
2e65      2057: ;
2e65 18     2058: a.goodread: clc
2e66 a5b4   2059:      lda buftop
2e68 6980   2060:      adc #$80
2e6a 85b4   2061:      sta buftop
2e6c a5b5   2062:      lda buftop+1
2e6e 6900   2063:      adc #0
2e70 85b5   2064:      sta buftop+1
2e72 4cf82d 2065:      jmp a.sectr_chkbuf
2e75      2066: ;
2e75 4cc93d 2067: a.havsectors: jmp docarry
2e78      2068: ;-----
2e78      2069: ;
2e78 a900   2070: sector_block: lda #0
2e7a 8d2963 2071:      sta currnt.sectr
2e7d 8d2a63 2072:      sta currnt.sectr+1
2e80 a5c8   2073:      lda s.org
2e82 85c6   2074:      sta pc
2e84 a5c9   2075:      lda s.org+1
2e86 85c7   2076:      sta pc+1
2e88 20a73d 2077: set_blkptrs: jsr setndx
2e8b 4c853d 2078:      jmp say_more
2e8e      2079: ;-----
2e8e      2080: ;
2e8e      2081: zero_sector.map:
2e8e      2082: ;
2e8e a900   2083:      lda #0
2e90 a25a   2084:      ldx #90
2e92 9dce63 2085: a.zero_sector.map: sta sector.map,x
2e95 ca     2086:      dex
2e96 10fa   2087:      bpl a.zero_sector.map
2e98 60     2088:      rts
2e99      2089: ;-----
2e99      2090: ;
2e99      2091: get_sbit:
2e99      2092: ;
2e99 a900   2093:      lda #0
2e9b 8d2c63 2094:      sta xhi
2e9e 8e2b63 2095:      stx xlo
2ea1 98     2096:      tya
2ea2 a203   2097:      ldx #3
2ea4 4a     2098: a.by8:   lsr
2ea5 6e2b63 2099:      ror xlo
2ea8 6e2c63 2100:      ror xhi
2eab ca     2101:      dex
2eac d0f6   2102:      bne a.by8
2eae a205   2103:      ldx #5
2eb0 4e2c63 2104: a.by32:  lsr xhi
```

```

2eb3 ca 2105: dex
2eb4 d0fa 2106: bne a.by32
2eb6 ae2b63 2107: ldx xlo
2eb9 ac2c63 2108: ldy xhi
2ebc 60 2109: rts
2ebd 2110: ;-----
2ebd 2111:
2ebd 2112: ; .PAGE "File Disassembly Routines"
2ebd 2113: ; .include file
2ebd 2114:
2ebd 2115: ; ... FILE ...
2ebd 2116: ;
2ebd 2117: ; File Disassembly Routines.
2ebd 2118: ;
2ebd 2119: ;-----
2ebd 2120: ;
2ebd 2121: file_dis:
2ebd 2122: ;
2ebd b01d 2123: bcs file_disloop
2ebf adf262 2124: lda more.
2ec2 d003 2125: bne b.chk88
2ec4 38 2126: b.nevrmind: sec
2ec5 b012 2127: bcs b.nmj
2ec7 2128: ;
2ec7 adde62 2129: b.chk88: lda cioy
2eca c988 2130: cmp #$88
2ecc f0f6 2131: beq b.nevrmind
2ece 20a73d 2132: jsr setndx
2ed1 20b93d 2133: jsr setbsize
2ed4 20342f 2134: jsr loadfile ;Yes, load buffer.
2ed7 9003 2135: bcc file_disloop
2ed9 4c7e3d 2136: b.nmj: jmp no_more ;EOF reached.
2edc 2137: ;
2edc 20a03d 2138: file_disloop: jsr orleft ;left.to.do>0?
2edf d040 2139: bne b.fdcont ;Yes.
2ee1 207f3d 2140: jsr zmore ;more=0.
2ee4 2141: ;
2ee4 2142: ;Copy file segment address
2ee4 2143: ;bytes to seg.adr.
2ee4 2144: ;
2ee4 a003 2145: ldy #3
2ee6 b1b2 2146: b.pcloop: lda (bufptr),y
2ee8 990163 2147: sta seg.adr,y
2eeb 88 2148: dey
2eec 10f8 2149: bpl b.pcloop
2eee 2150: ;
2eee 18 2151: clc ;Bump bufptr past
2eef a5b2 2152: lda bufptr ;address bytes.
2ef1 6904 2153: adc #4
2ef3 85b2 2154: sta bufptr
2ef5 a5b3 2155: lda bufptr+1
2ef7 6900 2156: adc #0
2ef9 85b3 2157: sta bufptr+1

```

```
2efb      2158: ;
2efb      2159: ;Check seg.adr to see if PC
2efb      2160: ;needs to be reset.
2efb      2161: ;
2efb ad0163 2162:      lda seg.adr
2efe c5c6  2163:      cmp pc
2f00 d007  2164:      bne b.reset_pc
2f02 ad0263 2165:      lda seg.adr+1
2f05 c5c7  2166:      cmp pc+1
2f07 d00e  2167:      bne b.chkload
2f09      2168: ;
2f09      2169: ;pc=seg.adr;more=1
2f09      2170: ;Will just show new .ORG on
2f09      2171: ;next output line
2f09      2172: ;
2f09 ad0163 2173: b.reset_pc: lda seg.adr
2f0c 85c6  2174:      sta pc
2f0e ad0263 2175:      lda seg.adr+1
2f11 85c7  2176:      sta pc+1
2f13 38    2177: b.ld1:   sec
2f14 4c863d 2178:      jmp imore ;more=1
2f17      2179: ;
2f17      2180: ;Get next byte to be dis-
2f17      2181: ;assembled - will be in X reg -
2f17      2182: ;then decrement left.to.do.
2f17      2183: ;
2f17 add562 2184: b.chkload: lda loadflag
2f1a d005  2185:      bne b.fdcont
2f1c eed562 2186:      inc loadflag
2f1f d0f2  2187:      bne b.ld1
2f21      2188: ;
2f21 20792c 2189: b.fdcont: jsr fetch_byte
2f24 ad0363 2190:      lda left.to.do
2f27 d003  2191:      bne b.A2
2f29 ce0463 2192:      dec left.to.do+1
2f2c ce0363 2193: b.A2:   dec left.to.do
2f2f 20853d 2194:      jsr say_more ;more=1.
2f32 8a    2195:      txa
2f33 60    2196:      rts
2f34      2197: ;
2f34      2198: ;-----
2f34      2199: ;File loading routines. =
2f34      2200: ;-----
2f34      2201: ;
2f34 20193f 2202: loadfile: jsr chk_in ;Check for right disk.
2f37      2203: ;
2f37 20c32f 2204: loadseg: jsr check88
2f3a      2205: ;
2f3a ad1f63 2206: b.clib:  lda left.in.buf ;Check for
2f3d 0d2063 2207:      ora left.in.buf+1 ;empty buffer.
2f40 f009  2208:      beq b.bufis_mt
2f42 200530 2209:      jsr chek_buf ;It's not.
2f45 20c53a 2210:      jsr fload
```

```
2f48 4cab2f 2211:      jmp next
2f4b      2212:      ;
2f4b ade862 2213: b.bufis_mt: lda binflag
2f4e f003  2214:      beq nextseg
2f50 4ccb2f 2215:      jmp nbload
2f53      2216:      ;
2f53 20c32f 2217: nextseg: jsr check88
2f56 38    2218:      sec      ;Room in buffer
2f57 a5b6  2219:      lda bufsize ;for segment
2f59 e905  2220:      sbc #5    ;addresses?
2f5b a5b7  2221:      lda bufsize+1
2f5d e900  2222:      sbc #0
2f5f b003  2223:      bcs b.jladr ;Yes.
2f61 4cc93d 2224:      jmp docarry ;No.
2f64      2225:      ;
2f64 20863a 2226: b.jladr: jsr load_adr
2f67 f0fb  2227:      beq b.jladr
2f69 20b23a 2228:      jsr load_end
2f6c      2229:      ;
2f6c      2230: ;seg.size=seg.end-seg.org+1
2f6c      2231: ;left.in.buf=seg.size.
2f6c      2232:      ;
2f6c 38    2233:      sec
2f6d adfc62 2234:      lda seg.end
2f70 edfa62 2235:      sbc seg.org
2f73 8dfc62 2236:      sta seg.size
2f76 adfd62 2237:      lda seg.end+1
2f79 edfb62 2238:      sbc seg.org+1
2f7c 8dfd62 2239:      sta seg.size+1
2f7f 18    2240:      clc
2f80 adfc62 2241:      lda seg.size
2f83 6901  2242:      adc #1
2f85 8dfc62 2243:      sta seg.size
2f88 8d1f63 2244:      sta left.in.buf
2f8b adfd62 2245:      lda seg.size+1
2f8e 6900  2246:      adc #0
2f90 8dfd62 2247:      sta seg.size+1
2f93 8d2063 2248:      sta left.in.buf+1
2f96      2249:      ;
2f96      2250: ;Copy start and end addresses
2f96      2251: ;into file buffer.
2f96      2252:      ;
2f96 a004  2253:      ldy #4
2f98 88    2254: b.sav_fhed: dey
2f99 b9fa62 2255:      lda seg.org,y
2f9c c8    2256:      iny
2f9d 91b4  2257:      sta (buftop),y
2f9f 88    2258:      dey
2fa0 d0f6  2259:      bne b.sav_fhed
2fa2 20f42f 2260:      jsr bump4
2fa5 200530 2261:      jsr chek_buf
2fa8 20c53a 2262:      jsr fload
2fab      2263:      ;
```

```

2fab 202230 2264: next:   jsr bumptop
2fae 4c532f 2265:        jmp nextseg ;Get next segment.
2fb1      2266: ;
2fb1 68    2267: pla_eof: pla
2fb2 68    2268:        pla
2fb3 207f3d 2269: b.file_eof: jsr zmore ;more=0
2fb6 4cc93d 2270: buf_full: jmp docarry
2fb9      2271: ;
2fb9 8cde62 2272: b.f_err: sty cioy
2fbc c088   2273:        cpy #$88
2fbe f0f3   2274:        beq b.file_eof
2fc0 4cca43 2275:        jmp io_err
2fc3      2276: ;-----
2fc3      2277: ;
2fc3 adde62 2278: check88: lda cioy
2fc6 c988   2279:        cmp #$88
2fc8 f0e7   2280:        beq pla_eof
2fca 60     2281:        rts
2fcb      2282: ;-----
2fcb      2283: ;
2fcb 20f42f 2284: nbload: jsr bump4
2fce 200530 2285:        jsr chek_buf
2fd1 205b3a 2286:        jsr bload
2fd4 18     2287:        clc
2fd5 a5c8   2288:        lda s.org
2fd7 8d5e7d 2289:        sta bufadr
2fda 7d4803 2290:        adc icbll,x
2fdd 8d607d 2291:        sta bufadr+2
2fe0 a5c9   2292:        lda s.org+1
2fe2 8d5f7d 2293:        sta bufadr+1
2fe5 7d4903 2294:        adc icblh,x
2fe8 8d617d 2295:        sta bufadr+3
2feb 202230 2296:        jsr bumptop
2fee c088   2297:        cpy #$88
2ff0 f0c1   2298:        beq b.file_eof
2ff2 d0c2   2299:        bne buf_full
2ff4      2300: ;-----
2ff4      2301: ;
2ff4      2302: ;Bump buftop 4 bytes.
2ff4      2303: ;
2ff4 18     2304: bump4:  clc
2ff5 a5b4   2305:        lda buftop
2ff7 6904   2306:        adc #4
2ff9 85b4   2307:        sta buftop
2ffb a5b5   2308:        lda buftop+1
2ffd 6900   2309:        adc #0
2fff 85b5   2310:        sta buftop+1
3001 60     2311:        rts
3002      2312: ;-----
3002      2313: ;
3002      2314: ;bufsize=freehi-buftop to
3002      2315: ;Check for room in buffer.
3002 fd4803 2316:        sbc icbll,x

```

```

3005      2317: ;
3005 38   2318: chek_buf: sec
3006 adf562 2319:      lda freehi
3009 e5b4  2320:      sbc buftop
300b 85b6  2321:      sta bufsize
300d adf662 2322:      lda freehi+1
3010 e5b5  2323:      sbc buftop+1
3012 85b7  2324:      sta bufsize+1
3014 9007  2325:      bcc b.bjmp
3016 d004  2326:      bne b.rume
3018 a5b6  2327:      lda bufsize
301a f001  2328:      beq b.bjmp
301c 60    2329: b.rume:  rts
301d      2330: ;
301d 68    2331: b.bjmp:  pla
301e 68    2332:      pla
301f 4cc93d 2333:      jmp docarry
3022      2334: ;-----
3022      2335: ;
3022      2336: ;buftop==+icbll,x.
3022      2337: ;left.in.buf==+icbll,x.
3022      2338: ;
3022 18    2339: bumptop: clc
3023 a5b4  2340:      lda buftop
3025 7d4803 2341:      adc icbll,x
3028 85b4  2342:      sta buftop
302a a5b5  2343:      lda buftop+1
302c 7d4903 2344:      adc icblh,x
302f 85b5  2345:      sta buftop+1
3031 38    2346:      sec
3032 ad1f63 2347:      lda left.in.buf
3035 fd4803 2348:      sbc icbll,x
3038 8d1f63 2349:      sta left.in.buf
303b ad2063 2350:      lda left.in.buf+1
303e fd4903 2351:      sbc icblh,x
3041 8d2063 2352:      sta left.in.buf+1
3044 60    2353:      rts
3045      2354: ;-----
3045      2355: ;
3045      2356: ;Open input file.
3045      2357: ;
3045 20053b 2358: dofile: jsr open_in
3048      2359: ;
3048      2360: ;left.in.buf=0
3048      2361: ;seg.adr=left.to.do=0
3048      2362: ;
3048 a900  2363: b.zl.in.buf: lda #0
304a 8d1f63 2364:      sta left.in.buf
304d 8d2063 2365:      sta left.in.buf+1
3050      2366: ;
3050 a203  2367:      ldx #3
3052 9d0163 2368: b.zseg.adr: sta seg.adr,x
3055 ca    2369:      dex

```

```

3056 10fa 2370:      bpl b.zseg.adr
3058 4c882e 2371:      jmp set_blkptrs
305b      2372: ;-----
305b      2373:
305b      2374:
305b      2375:
305b      2376: ;      .PAGE "Write Equate file"
305b      2377: ;      .include equs
305b      2378:
305b      2379: ;      ... EQUUS ...
305b      2380: ;
305b      2381: ;-----
305b      2382: ; Routines to write Equate
305b      2383: ; and .ASM files.
305b      2384: ;
305b      2385: ;-----
305b      2386: ;
305b      2387: write_equates:
305b      2388: ;
305b ad9c61 2389:      lda asmflg
305e d003 2390:      bne c.do_equates
3060 20e532 2391:      jsr endline
3063      2392: ;
3063 20143c 2393: c.do_equates: jsr dash
3066 20543f 2394:      jsr blokclose
3069 addc62 2395:      lda totfiles
306c a941 2396:      lda # low inc_lino
306e 8d3663 2397:      sta to.lino
3071 a93d 2398:      lda # high inc_lino
3073 8d3763 2399:      sta to.lino+1
3076 addc62 2400:      lda totfiles
3079 8ddb62 2401:      sta mainfiles
307c a900 2402:      lda #0
307e 8ddc62 2403:      sta totfiles
3081 a204 2404:      idx #4
3083 bdba61 2405: c.lolp:  lda secndno,x
3086 9d3863 2406:      sta lnum,x
3089 ca 2407:      dex
308a 10f7 2408:      bpl c.lolp
308c      2409: ;
308c 20cd3e 2410:      jsr set_extender
308f a945 2411:      lda #'E'
3091 99ae63 2412:      sta outfn,y
3094 a951 2413:      lda #'Q'
3096 99af63 2414:      sta outfn+1,y
3099 20333a 2415:      jsr opn_disk
309c a900 2416:      lda #0
309e 8dff62 2417:      sta files.size
30a1 8d0063 2418:      sta files.size+1
30a4 a907 2419:      lda # low sequ_m
30a6 a062 2420:      ldy # high sequ_m
30a8 a200 2421:      idx #0
30aa 200e3c 2422:      jsr misc_line

```

```
30ad 20143c 2423:    jsr dash
30b0 a200 2424:    ldx #0
30b2 20d332 2425: c.nextpage: jsr setpage
30b5 ad1863 2426:    lda page
30b8 c9e4 2427:    cmp #$e4
30ba d003 2428:    bne c.showthis
30bc 209432 2429:    jsr sho_fp
30bf a900 2430: c.showthis: lda #0
30c1 85ba 2431:    sta label.adr
30c3 20422c 2432: c.pageloop: jsr reset_ptr
30c6 209e3e 2433:    jsr getfirst
30c9 1015 2434:    bpl c.nx_label
30cb 20a33e 2435:    jsr getsecond
30ce 3010 2436:    bmi c.nx_label
30d0 20de3d 2437:    jsr to_pline
30d3 ad1863 2438:    lda page
30d6 f003 2439:    beq c.lopg
30d8 20fc35 2440:    jsr byte_to_hex
30db a5ba 2441: c.lopg:  lda label.adr
30dd 20f43d 2442:    jsr do_pline
30e0 e6ba 2443: c.nx_label: inc label.adr
30e2 f009 2444:    beq c._sysdone.
30e4 a5ba 2445:    lda label.adr
30e6 cd1963 2446:    cmp page.top
30e9 90d8 2447:    bcc c.pageloop
30eb f0d6 2448:    beq c.pageloop
30ed      2449: ;
30ed ad1863 2450: c._sysdone.: lda page
30f0 d005 2451:    bne c.chkp3
30f2 203532 2452:    jsr sho_p0
30f5 f007 2453:    beq c._pagedone
30f7 c903 2454: c.chkp3: cmp #3
30f9 d003 2455:    bne c._pagedone
30fb 207b32 2456:    jsr sho_p3
30fe      2457: ;
30fe aed662 2458: c._pagedone: ldx pagex
3101 ec165e 2459:    cpx tsize
3104 b003 2460:    bcs c.chkforzeq
3106 4cb230 2461:    jmp c.nextpage
3109      2462: ;
3109 a901 2463: c.chkforzeq: lda #1
310b 8dee62 2464:    sta good.label
310e addf62 2465:    lda zp.flag
3111 d003 2466:    bne c.do_zqs
3113 4c6731 2467:    jmp c.xeqs
3116      2468: ;
3116 20143c 2469: c.do_zqs: jsr dash
3119 a9ef 2470:    lda # low zeq_m
311b a061 2471:    ldy # high zeq_m
311d a200 2472:    ldx #0
311f 200e3c 2473:    jsr misc_line
3122 20143c 2474:    jsr dash
3125 205a3e 2475:    jsr get_labelcount
```



```

3128 a200 2476:    ldx #0
312a 86be 2477:    stx this.lbl
312c 86bf 2478:    stx this.lbl+1
312e 86ba 2479:    stx label.adr
3130 86bb 2480:    stx label.adr+1
3132 f003 2481:    beq c.ovritl
3134 20793e 2482: c.nxtzq: jsr inths.lbl
3137 e6ba 2483: c.ovritl: inc label.adr
3139 38 2484:      sec
313a ad1263 2485:    lda label.count
313d e5be 2486:    sbc this.lbl
313f ad1363 2487:    lda label.count+1
3142 e5bf 2488:    sbc this.lbl+1
3144 9021 2489:    bcc c.xeqs
3146 20803e 2490:    jsr point
3149 d01c 2491:    bne c.xeqs
314b 20f232 2492:    jsr andl
314e d0e4 2493:    bne c.nxtzq
3150 a200 2494:    ldx #0
3152 203127 2495:    jsr labelize
3155 a920 2496:    lda #space
3157 9d3e63 2497:    sta printline,x
315a 20dc3e 2498:    jsr eqlize
315d a000 2499:    ldy #0
315f b1bc 2500:    lda (label.ptr),y
3161 20f43d 2501:    jsr do_pline
3164 4c3431 2502:    jmp c.nxtzq
3167      2503: ;
3167 20143c 2504: c.xeqs: jsr dash
316a a91f 2505:    lda # low xref_m
316c a062 2506:    ldy # high xref_m
316e a200 2507:    ldx #0
3170 200e3c 2508:    jsr misc_line
3173 20143c 2509:    jsr dash
3176 205a3e 2510:    jsr get_labelcount
3179 204c3e 2511:    jsr zthis.lbl
317c f006 2512:    beq c.noxin
317e 20793e 2513: c.nxt_lbl: jsr inths.lbl
3181 20673e 2514:    jsr delcount
3184 ad1363 2515: c.noxin: lda label.count+1
3187 3023 2516:    bmi c.ext_eqsdone
3189 20f232 2517:    jsr andl
318c d0f0 2518:    bne c.nxt_lbl
318e a200 2519:    ldx #0
3190 203127 2520:    jsr labelize
3193 a920 2521:    lda #space
3195 9d3e63 2522:    sta printline,x
3198 20dc3e 2523:    jsr eqlize
319b 20803e 2524:    jsr point
319e f0de 2525:    beq c.nxt_lbl
31a0 20fc35 2526:    jsr byte_to_hex
31a3 88 2527:    dey
31a4 b1bc 2528:    lda (label.ptr),y

```

```
31a6 20f43d 2529:    jsr do_pline
31a9 4c7e31 2530:    jmp c.nxt_lbl
31ac      2531:    ;
31ac 20143c 2532: c.ext_eqsdone: jsr dash
31af ad9c61 2533:    lda asmflg
31b2 d00c  2534:    bne c.closeqs
31b4 202f3c 2535:    jsr twospcs
31b7 20cd3e 2536:    jsr set_extender
31ba 20533c 2537:    jsr get_link
31bd 20ee38 2538: c.cfncr: jsr write_line
31c0      2539:    ;
31c0 20543f 2540: c.closeqs: jsr blokclose
31c3 adfe62 2541:    lda dskon
31c6 d003  2542:    bne c.chkifle
31c8 4ce524 2543:    jmp disdone
31cb ad9c61 2544: c.chkifle: lda asmflg
31ce d003  2545:    bne ifile
31d0 4ce524 2546:    jmp disdone
31d3      2547:    ;
31d3 ace962 2548: ifile:  ldy outdot
31d6 a941  2549:    lda #'A'
31d8 99ae63 2550:    sta outfn,y
31db a953  2551:    lda #'S'
31dd 99af63 2552:    sta outfn+1,y
31e0 a94d  2553:    lda #'M'
31e2 99b063 2554:    sta outfn+2,y
31e5 a204  2555:    ldx #4
31e7 a930  2556:    lda #'O'
31e9 9d3863 2557: c.lulp:  sta lnum,x
31ec ca    2558:    dex
31ed 10fa  2559:    bpl c.lulp
31ef a939  2560:    lda #'9'
31f1 8d3c63 2561:    sta lnum+4
31f4 202a3f 2562:    jsr chk_out
31f7 cedc62 2563:    dec totfiles
31fa 20cd3e 2564:    jsr set_extender
31fd a945  2565:    lda #'E'
31ff 99ae63 2566:    sta outfn,y
3202 a951  2567:    lda #'Q'
3204 99af63 2568:    sta outfn+1,y
3207 20533c 2569: c.eifloop: jsr get_link
320a 20ee38 2570:    jsr write_line
320d 20583d 2571:    jsr nxt_outfn
3210 a900  2572:    lda #0
3212 8dd462 2573:    sta try
3215 cedc62 2574:    dec totfiles
3218 d0ed  2575:    bne c.eifloop
321a addb62 2576:    lda mainfiles
321d f00d  2577:    beq c.ifdun
321f 8ddc62 2578:    sta totfiles
3222 20cd3e 2579:    jsr set_extender
3225 a900  2580:    lda #0
3227 8ddb62 2581:    sta mainfiles
```

```
322a f0db 2582:      beq c.eifloop
322c      2583:      ;
322c 20e532 2584: c.ifdun: jsr endline
322f 20143c 2585:      jsr dash
3232 4c543f 2586:      jmp blokclose
3235      2587: ;-----
3235      2588: ;
3235      2589: sho_p0:
3235 a994 2590:      lda # low p0.labels
3237 85b8 2591:      sta syslabel
3239 a95c 2592:      lda # high p0.labels
323b 85b9 2593:      sta syslabel+1
323d a917 2594:      lda # low p0.adrlst
323f 8d6632 2595:      sta c.show+1
3242 a95e 2596:      lda # high p0.adrlst
3244 8d6732 2597:      sta c.show+2
3247 a90c 2598:      lda #12
3249 8d6f32 2599:      sta c.yy+1
324c a000 2600: c.sho0: ldy #0
324e 84cd 2601:      sty ytemp
3250 209e3e 2602: c.sho0_loop: jsr getfirst
3253 1016 2603:      bpl c.nxt0
3255 20a33e 2604:      jsr getsecond
3258 3011 2605:      bmi c.nxt0
325a 20de3d 2606:      jsr to_pline
325d ad1863 2607:      lda page
3260 f003 2608:      beq c.show
3262 20fc35 2609:      jsr byte_to_hex
3265 b9175e 2610: c.show: lda p0.adrlst,y
3268 20f43d 2611:      jsr do_pline
326b a4cd 2612: c.nxt0: ldy ytemp
326d c8 2613:      iny
326e c00c 2614: c.yy:  cpy #12
3270 b008 2615:      bcs c.p0shode
3272 84cd 2616:      sty ytemp
3274 201a3e 2617:      jsr bumps1.6
3277 4c5032 2618:      jmp c.sho0_loop
327a      2619:      ;
327a 60 2620: c.p0shode: rts
327b      2621:      ;
327b      2622: sho_p3:
327b      2623:      ;
327b a9dc 2624:      lda # low p3.labels
327d 85b8 2625:      sta syslabel
327f a95c 2626:      lda # high p3.labels
3281 85b9 2627:      sta syslabel+1
3283 a923 2628:      lda # low p3.adrlst
3285 8d6632 2629:      sta c.show+1
3288 a95e 2630:      lda # high p3.adrlst
328a 8d6732 2631:      sta c.show+2
328d a90b 2632:      lda #11
328f 8d6f32 2633:      sta c.yy+1
3292 d0b8 2634:      bne c.sho0
```

```

3294      2635: ;-----
3294      2636: ;
3294      2637: sho_fp:
3294      2638: ;
3294 a91e 2639:      lda # low fp.labels
3296 85b8 2640:      sta syslabel
3298 a95d 2641:      lda # high fp.labels
329a 85b9 2642:      sta syslabel+1
329c a000 2643:      ldy #0
329e 84cd 2644:      sty ytemp
32a0 209e3e 2645: c.sfploop: jsr getfirst
32a3 1014 2646:      bpl c.nfp
32a5 20a33e 2647:      jsr getsecond
32a8 300f 2648:      bmi c.nfp
32aa 20de3d 2649:      jsr to_pline
32ad b92f5e 2650:      lda fp.adrlst+1,y
32b0 20fc35 2651:      jsr byte_to_hex
32b3 b92e5e 2652:      lda fp.adrlst,y
32b6 20f43d 2653:      jsr do_pline
32b9 a4cd 2654: c.nfp:  ldy ytemp
32bb c8 2655:      iny
32bc c8 2656:      iny
32bd c042 2657:      cpy #66
32bf f011 2658:      beq c.fpshode
32c1 84cd 2659:      sty ytemp
32c3 18 2660:      clc
32c4 a5b8 2661:      lda syslabel
32c6 6906 2662:      adc #6
32c8 85b8 2663:      sta syslabel
32ca a5b9 2664:      lda syslabel+1
32cc 6900 2665:      adc #0
32ce 85b9 2666:      sta syslabel+1
32d0 d0ce 2667:      bne c.sfploop
32d2      2668: ;
32d2 60 2669: c.fpshode: rts
32d3      2670: ;-----
32d3      2671: ;
32d3 a000 2672: setpage: ldy #0
32d5 bde45d 2673: c.nploop: lda page.table,x
32d8 991663 2674:      sta page.loc,y
32db e8 2675:      inx
32dc c8 2676:      iny
32dd c005 2677:      cpy #5
32df d0f4 2678:      bne c.nploop
32e1 8ed662 2679:      stx pagex
32e4 60 2680:      rts
32e5      2681: ;-----
32e5      2682: endline:
32e5      2683: ;
32e5 201d3c 2684:      jsr onespace
32e8 a992 2685:      lda # low end_m
32ea a062 2686:      ldy # high end_m
32ec a207 2687:      ldx #7

```

```

32ee 200e3c 2688:      jsr misc_line
32f1 60      2689:      rts
32f2      2690: ;-----
32f2      2691: ;
32f2 209c29 2692: andl:  jsr get_lbit
32f5 3d1861 2693:      and bit.table,x
32f8 60      2694:      rts
32f9      2695: ;-----
32f9      2696:
32f9      2697:
32f9      2698: ;      .PAGE "Data File Routines"
32f9      2699: ;      .include byte
32f9      2700:
32f9      2701: ;      ... BYTE ...
32f9      2702: ;
32f9      2703: ;Routines to create data files
32f9      2704: ;for Action!, Assembler, and
32f9      2705: ;BASIC.
32f9      2706: ;
32f9      2707: ;Entry - A reg = 'A', 'B', or 'D
32f9      2708: ;ANDING results in 0=ACTION,
32f9      2709: ;2=Assembler, & 4=BASIC. Use
32f9      2710: ;result as an index.
32f9      2711: ;
32f9 d633fc 2712: abd_tbl: dc.w action,assembler
32fd 0634    2713:      dc.w basic
32ff      2714: ;
32ff 2906    2715: datafile: and #6
3301 8d2563 2716:      sta mode
3304 aa      2717:      tax
3305 bdf932 2718:      lda abd_tbl,x ;Set JMP
3308 8dd433 2719:      sta langjmp+1 ;to called
330b bdfa32 2720:      lda abd_tbl+1,x ;routine.
330e 8dd533 2721:      sta langjmp+2
3311 ad9c61 2722:      lda asmflg ;Save these.
3314 8d1963 2723:      sta oldifile
3317 ad9b61 2724:      lda lino.flag
331a 8d1863 2725:      sta oldlino
331d a000    2726:      ldy #0 ;Zero byte count.
331f 8c1263 2727:      sty totaldc.bs
3322 8c1363 2728:      sty totaldc.bs+1
3325 c8      2729:      iny ;Turn off ASM file
3326 8c9c61 2730:      sty asmflg ;flag.
3329 c8      2731: d.fbylp: iny ;Find end of
332a b97c62 2732:      lda byte_m,y ;BYTE mnem.
332d c99b    2733:      cmp #cr
332f d0f8    2734:      bne d.fbylp
3331 98      2735:      tya
3332 6903    2736:      adc #3
3334 8dfa35 2737:      sta indent+1
3337 a95e    2738:      lda # low label.buffer
3339 8d1063 2739:      sta labeltop
333c a967    2740:      lda # high label.buffer

```

```

333e 8d1163 2741:    sta labeltop+1
3341 206225 2742:    jsr setbuffer
3344 206840 2743:    jsr get_infn ;Get input fname.
3347 205b37 2744:    jsr :2crs
334a 209c3f 2745: d.out:  jsr outfn_in ;Output fname.
334d a000 2746:    ldy #0    ;Indicate file not
334f 8ddd62 2747:    sta dskopnflg ;open
3352 ad6663 2748:    lda inline ;CR means screen
3355 c99b 2749:    cmp #cr   ;or printer output
3357 f015 2750:    beq d.ovrout ;only.
3359      2751: ;
3359 ad6763 2752: d.dout: lda inline+1 ;Check to
335c c93a 2753:    cmp #'.' ;see if user used
335e f00d 2754:    beq d.coln ;a Dn: in output
3360 ad6863 2755:    lda inline+2 ;filename and
3363 c93a 2756:    cmp #'.' ;set it to default
3365 f006 2757:    beq d.coln ;output drive if
3367 adcc61 2758:    lda d2    ;not.
336a 8daf63 2759:    sta outfn+1
336d c8 2760: d.coln: iny
336e 8cfe62 2761: d.ovrout: sty dskon ;Write to disk.
3371 a000 2762:    ldy #0
3373 8cf002 2763:    sty crsinh
3376 ad8f63 2764:    lda infn+1
3379 cdaf63 2765:    cmp outfn+1
337c f001 2766:    beq d.same
337e c8 2767:    iny
337f 8cf062 2768: d.same: sty drives
3382 8ce562 2769:    sty both.flag
3385      2770: ;
3385      2771: ;All for skipping over 6
3385      2772: ;file header bytes.
3385      2773: ;
3385 208037 2774:    jsr printsi
3388      2775: ;
3388 9b9b7f 2776:    dc.b cr,cr,tab,"Skip over"
3394 206669 2777:    dc.b " first 6 bytes? ",EM
33a5      2778: ;
33a5 20dc37 2779: d.yn:  jsr getcap
33a8 a001 2780:    ldy #1
33aa c959 2781:    cmp #'Y'
33ac f005 2782:    beq d.itsy
33ae c94e 2783:    cmp #'N'
33b0 d0f3 2784:    bne d.yn
33b2 88 2785:    dey
33b3 8cd862 2786: d.itsy: sty all.
33b6 20113f 2787:    jsr newscreen
33b9 20a940 2788:    jsr indisk_prmpt
33bc adf062 2789:    lda drives
33bf f003 2790:    beq d.open_dat
33c1 201141 2791:    jsr outdisk_prmpt
33c4 20053b 2792: d.open_dat: jsr open_in
33c7 201d35 2793:    jsr set_buffer

```

```
33ca 20de34 2794:      jsr load_dat
33cd ad9561 2795:      lda scr
33d0 8d2f02 2796:      sta sdmctl
33d3 4cffff 2797: langjmp: jmp dummy
33d6      2798: ;-----
33d6      2799: ;
33d6      2800: action:
33d6      2801: ;
33d6 a901 2802:      lda #1      ;No line numbers!
33d8 8d9b61 2803:      sta lino.flag
33db a93e 2804:      lda # low printline
33dd 8d2163 2805:      sta line.ptr
33e0 a963 2806:      lda # high printline
33e2 8d2263 2807:      sta line.ptr+1
33e5 a9a8 2808:      lda # low code_m ;define an
33e7 a062 2809:      ldy # high code_m ;Action!
33e9 203834 2810:      jsr nameit ;PROC.
33ec a95b 2811:      lda #['
33ee 8d4063 2812:      sta printline+2
33f1 a99b 2813:      lda #cr
33f3 8d4163 2814:      sta printline+3
33f6 20ee38 2815:      jsr write_line
33f9 4cbd34 2816:      jmp doloop
33fc      2817: ;-----
33fc      2818: ;
33fc      2819: assembler:
33fc      2820: ;
33fc a9ba 2821:      lda # low mega_m ;Add a label.
33fe a062 2822:      ldy # high mega_m
3400 203834 2823:      jsr nameit
3403 4cbd34 2824:      jmp doloop
3406      2825: ;
3406      2826: ;-----
3406      2827: ;
3406      2828: basic:
3406      2829: ;
3406 204334 2830: d.bloop: jsr loopset
3409 a200 2831:      idx #0
340b a99d 2832:      lda # low data_m
340d 8d453c 2833:      sta from+1
3410 a962 2834:      lda # high data_m
3412 8d463c 2835:      sta from+2
3415 20423c 2836:      jsr move
3418 a920 2837:      lda #space
341a 9d3e63 2838:      sta printline,x
341d 8ef362 2839: d.bloop2: stx xsave
3420 205b34 2840:      jsr readbyt
3423 20c135 2841:      jsr shodecml
3426 206a34 2842:      jsr repoint
3429 9006 2843:      bcc d.bold
342b 20db34 2844:      jsr xload_dat
342e aef362 2845:      idx xsave
3431 208d34 2846: d.bold: jsr oneless
```

```

3434 90e7 2847:      bcc d.bloop2
3436 b0ce 2848:      bcs d.bloop
3438      2849: ;-----
3438      2850: ;
3438 a200 2851: nameit: ldx #0
343a 200e3c 2852:      jsr misc_line
343d 201d3c 2853:      jsr onespace
3440 4c372c 2854:      jmp clr_printline
3443      2855: ;
3443      2856: ;
3443 ad9a61 2857: loopset: lda bytes
3446 8d0d63 2858:      sta bytemax
3449 20372c 2859:      jsr clr_printline
344c ad2563 2860:      lda mode
344f 2902 2861:      and #2
3451 f003 2862:      beq d.lpsx
3453 20db35 2863:      jsr setbyte
3456 20f035 2864: d.lpsx:  jsr getx
3459 aa 2865:      tax
345a 60 2866:      rts
345b      2867: ;
345b      2868: ;
345b a000 2869: readbyt: ldy #0
345d b1b2 2870:      lda (bufptr),y
345f ac2563 2871:      ldy mode
3462 c004 2872:      cpy #4
3464 f003 2873:      beq d.rdbx
3466 4cfc35 2874:      jmp byte_to_hex
3469      2875: ;
3469 60 2876: d.rdbx:  rts
346a      2877: ;
346a      2878: ;
346a      2879: repoint: inw totaldc.bs
346a ee1263 ----:      inc  totaldc.bs
346d d003 ----:      bne  ~a
346f ee1363 ----:      inc  totaldc.bs+1
3472 ----: ~a:
3472 2880:      inw  bufptr
3472 e6b2 ----:      inc  bufptr
3474 d002 ----:      bne  ~a
3476 e6b3 ----:      inc  bufptr+1
3478 ----: ~a:
3478 ad0e63 2881:      lda bytecount
347b d003 2882:      bne d.aloA
347d ce0f63 2883:      dec bytecount+1
3480 ce0e63 2884: d.aloA:  dec bytecount
3483 ad0e63 2885:      lda bytecount
3486 0d0f63 2886:      ora bytecount+1
3489 d017 2887:      bne d.clrx
348b 38 2888:      sec
348c 60 2889:      rts
348d      2890: ;
348d      2891: ;

```



```

348d ce0d63 2892: oneless: dec bytemax
3490 d005 2893:      bne comma
3492 20ac34 2894:      jsr linefull
3495 38 2895:      sec
3496 60 2896:      rts
3497      2897: ;
3497 ad2563 2898: comma: lda mode
349a f006 2899:      beq d.clrx
349c a92c 2900:      lda #','
349e 9d3e63 2901:      sta printline,x
34a1 e8 2902:      inx
34a2 18 2903: d.clrx: clc
34a3 60 2904:      rts
34a4      2905: ;
34a4      2906: ;
34a4 b93e63 2907: chekline: lda printline,y
34a7 c920 2908:      cmp #space
34a9 d001 2909:      bne linefull
34ab 60 2910:      rts
34ac      2911: ;
34ac a99b 2912: linefull: lda #cr
34ae 9d3e63 2913:      sta printline,x
34b1 4cee38 2914:      jmp write_line
34b4      2915: ;
34b4      2916: ;
34b4 20423c 2917: closedat: jsr move
34b7 20ee38 2918:      jsr write_line
34ba 4c543f 2919:      jmp blokclose
34bd      2920: ;
34bd      2921: ;
34bd 204334 2922: doloop: jsr loopset
34c0      2923: ;
34c0 a924 2924: d.loop2: lda #'$'
34c2 9d3e63 2925:      sta printline,x
34c5 e8 2926:      inx
34c6 205b34 2927:      jsr readbyt
34c9 206a34 2928:      jsr repoint
34cc 9006 2929:      bcc d.nold
34ce 20db34 2930:      jsr xload_dat
34d1 aef362 2931:      ldx xsave
34d4 208d34 2932: d.nold: jsr oneless
34d7 90e7 2933:      bcc d.loop2
34d9 b0e2 2934:      bcs doloop
34db      2935: ;-----
34db      2936: ;
34db 8ef362 2937: xload_dat: stx xsave
34de adde62 2938: load_dat: lda ciop
34e1 c988 2939:      cmp #$88
34e3 d005 2940:      bne d.skip6
34e5 68 2941:      pla
34e6 68 2942:      pla
34e7 4c3a35 2943:      jmp datdun
34ea      2944: ;

```

```

34ea add862 2945: d.skip6: lda all.
34ed f00e 2946:      beq fillbuffer
34ef a906 2947:      lda #6
34f1 85b6 2948:      sta bufsize
34f3 a900 2949:      lda #0
34f5 8dd862 2950:      sta all.
34f8 85b7 2951:      sta bufsize+1
34fa 205b3a 2952:      jsr bload
34fd      2953: ;
34fd 201d35 2954: fillbuffer: jsr set_buffer
3500 205b3a 2955:      jsr bload
3503 18 2956:      clc
3504 bd4803 2957:      lda icbll,x
3507 8d0e63 2958:      sta bytecount
350a 65b4 2959:      adc buftop
350c 85b4 2960:      sta buftop
350e bd4903 2961:      lda icblh,x
3511 8d0f63 2962:      sta bytecount+1
3514 65b5 2963:      adc buftop+1
3516 85b5 2964:      sta buftop+1
3518 202a3f 2965:      jsr chk_out
351b 18 2966:      clc
351c 60 2967:      rts
351d      2968: ;
351d      2969: ;
351d a5b0 2970: set_buffer: lda buffer
351f 85b2 2971:      sta bufptr
3521 85b4 2972:      sta buftop
3523 a5b1 2973:      lda buffer+1
3525 85b5 2974:      sta buftop+1
3527 85b3 2975:      sta bufptr+1
3529 38 2976:      sec
352a adf562 2977:      lda freehi
352d e5b0 2978:      sbc buffer
352f 85b6 2979:      sta bufsize
3531 adf662 2980:      lda freehi+1
3534 e5b1 2981:      sbc buffer+1
3536 85b7 2982:      sta bufsize+1
3538 18 2983:      clc
3539 60 2984:      rts
353a      2985: ;-----
353a      2986: ;
353a 20f035 2987: datdun: jsr getx
353d a8 2988:      tay
353e 20a434 2989: d.kpy: jsr chekline
3541 20372c 2990:      jsr clr_printline
3544 ad2563 2991:      lda mode
3547 d010 2992:      bne d.asmdun
3549      2993: ;
3549 a95d 2994:      lda #']'
354b 8d4063 2995:      sta printline+2
354e a99b 2996:      lda #cr
3550 8d4163 2997:      sta printline+3

```

```
3553 20ee38 2998:      jsr write_line
3556 4c5d35 2999:      jmp a_end
3559      3000: ;
3559      3001: ;
3559      3002: d.asmdun:
3559      3003: ;
3559 2902 3004:      and #2
355b f00e 3005:      beq d.basdun
355d 20372c 3006: a_end:  jsr clr_printline
3560 a93b 3007:      lda #','
3562 8d3e63 3008:      sta printline
3565 209e35 3009:      jsr showcount
3568 4c8335 3010:      jmp bytexit
356b      3011: ;
356b      3012: ;
356b      3013: d.basdun:
356b      3014: ;
356b a9a3 3015:      lda # low rem_m
356d a062 3016:      ldy # high rem_m
356f a200 3017:      ldx #0
3571 200e3c 3018:      jsr misc_line
3574 a200 3019:      ldx #0
3576 20423c 3020:      jsr move
3579 a920 3021:      lda #space
357b 9d3e63 3022:      sta printline,x
357e e8 3023:      inc
357f e8 3024:      inc
3580 209e35 3025:      jsr showcount
3583      3026: ;
3583 ad1863 3027: bytexit: lda oldlino
3586 8d9b61 3028:      sta lino.flag
3589 ad1963 3029:      lda oldifile
358c 8d9c61 3030:      sta asmflg
358f ad9761 3031:      lda line.start
3592 8d2163 3032:      sta line.ptr
3595 ad9861 3033:      lda line.start+1
3598 8d2263 3034:      sta line.ptr+1
359b 4c6542 3035:      jmp bydone
359e      3036: ;-----
359e      3037: ;
359e a205 3038: showcount: ldx #5
35a0 8ef362 3039:      stx xsave
35a3 ad1263 3040:      lda totaldc.bs
35a6 85d4 3041:      sta fr0
35a8 ad1363 3042:      lda totaldc.bs+1
35ab 85d5 3043:      sta fr0+1
35ad 203737 3044:      jsr Adecimal
35b0 20c435 3045:      jsr recover
35b3 a95c 3046:      lda # low size_m
35b5 8d453c 3047:      sta from+1
35b8 a962 3048:      lda # high size_m
35ba 8d463c 3049:      sta from+2
35bd 20b434 3050:      jsr closedat
```

```

35c0 60 3051: rts
35c1 3052: ;
35c1 3053: ;
35c1 203137 3054: shodecml: jsr decimal
35c4 a0ff 3055: recover: ldy #$ff
35c6 aef362 3056: ldx xsave
35c9 c8 3057: d.mvfp: iny
35ca b1f3 3058: lda (inbuff),y
35cc 9d3e63 3059: sta printline,x
35cf 3003 3060: bmi d.fplast
35d1 e8 3061: inx
35d2 d0f5 3062: bne d.mvfp
35d4 297f 3063: d.fplast: and #$7f
35d6 9d3e63 3064: sta printline,x
35d9 e8 3065: inx
35da 60 3066: rts
35db 3067: ;
35db 3068: ;
35db a97c 3069: setbyte: lda # low byte_m
35dd 8d453c 3070: sta from+1
35e0 a962 3071: lda # high byte_m
35e2 8d463c 3072: sta from+2
35e5 a204 3073: ldx #4
35e7 20423c 3074: jsr move
35ea a920 3075: lda #space
35ec 9d3e63 3076: sta printline,x
35ef 60 3077: rts
35f0 3078: ;-----
35f0 ad2563 3079: getx: lda mode
35f3 4a 3080: lsr
35f4 a8 3081: tay
35f5 b9f935 3082: lda indent,y
35f8 60 3083: rts
35f9 3084: ;
35f9 3085: ;
35f9 040705 3086: indent: dc.b 4,7,5
35fc 3087: ;-----
35fc 3088:
35fc 3089:
35fc 3090: ;-----
35fc 3091: ;BYTE_TO_HEX converts the byte
35fc 3092: ;in the A register to hex and
35fc 3093: ;stores it in PRINTLINE. The
35fc 3094: ;X reg is used for the index
35fc 3095: ;index into PRINTLINE.
35fc 3096: ;-----
35fc 3097: ;
35fc 3098: byte_to_hex:
35fc 3099: ;
35fc 48 3100: pha ;Save the A reg.
35fd 4a 3101: lsr ;Get the hi
35fe 4a 3102: lsr ;nibble.
35ff 4a 3103: lsr

```

```

3600 4a 3104:    lsr
3601 201336 3105:    jsr hex_nibble ;Convert it.
3604 9d3e63 3106:    sta printline,x ;Save it
3607 e8 3107:    inx
3608 68 3108:    pla      ;Restore A reg.
3609 290f 3109:    and #$0f  ;Get lo nibble.
360b 201336 3110:    jsr hex_nibble ;Convert it.
360e 9d3e63 3111:    sta printline,x ;Save it.
3611 e8 3112:    inx
3612 60 3113:    rts
3613      3114: ;
3613 84ce 3115: hex_nibble: sty tempy
3615 a8 3116:    tay
3616 b98261 3117:    lda hextbl,y
3619 a4ce 3118:    ldy tempy
361b 60 3119: got_nibble: rts
361c      3120: ;
361c      3121: ;-----
361c      3122: ;SHOW_HEX converts the byte
361c      3123: ;in the A register to hex and
361c      3124: ;prints it to the screen
361c      3125: ;preceded by a '$.
361c      3126: ;-----
361c      3127: ;
361c      3128: show_hex:
361c 48 3129:    pha
361d a924 3130:    lda #'$'
361f 206437 3131:    jsr putchar
3622 68 3132:    pla
3623      3133: ;
3623 48 3134: just_sho: pha
3624 4a 3135:    lsr
3625 4a 3136:    lsr
3626 4a 3137:    lsr
3627 4a 3138:    lsr
3628 201336 3139:    jsr hex_nibble
362b 206437 3140:    jsr putchar
362e 68 3141:    pla
362f 290f 3142:    and #$0f
3631 201336 3143:    jsr hex_nibble
3634 206437 3144:    jsr putchar
3637 60 3145:    rts
3638      3146: ;
3638      3147: ;-----
3638      3148: ;DIGITIZE converts a number
3638      3149: ;stored in inline to binary.
3638      3150: ;The number can be either
3638      3151: ;decimal or hex. The result
3638      3152: ;will be stored in NUMBER.
3638      3153: ;-----
3638      3154: ;
3638      3155: digitize:
3638      3156: ;

```

```
3638 a944 3157:    lda #'D'
363a 8de762 3158:    sta hex.flag
363d a966 3159:    lda # low inline
363f 85c2 3160:    sta sptr
3641 a963 3161:    lda # high inline
3643 85c3 3162:    sta sptr+1
3645 a000 3163:    ldy #0
3647 a200 3164:    ldx #0
3649 86ca 3165:    stx number
364b 86cb 3166:    stx number+1
364d 8ef362 3167:    stx xsave
3650      3168: ;
3650 b1c2 3169: d.find_dig: lda (sptr),y
3652 c920 3170:    cmp #space
3654 f004 3171:    beq d.inc_ptr
3656 c930 3172:    cmp #'0'
3658 d005 3173:    bne d.hex.
365a      3174: ;
365a 20bf3e 3175: d.inc_ptr: jsr inw.sptr
365d d0f1 3176:    bne d.find_dig
365f      3177: ;
365f c924 3178: d.hex.:  cmp #'$'
3661 d00a 3179:    bne d.next_dig
3663 a948 3180:    lda #'H'
3665 8de762 3181:    sta hex.flag
3668 20bf3e 3182:    jsr inw.sptr
366b d0e3 3183:    bne d.find_dig
366d      3184: ;
366d b1c2 3185: d.next_dig: lda (sptr),y
366f c99b 3186:    cmp #cr
3671 f030 3187:    beq d.x_0.
3673 c92c 3188:    cmp #','
3675 902a 3189:    bcc d.uh_oh
3677 f02a 3190:    beq d.x_0.
3679 c92d 3191:    cmp #'-'
367b f026 3192:    beq d.x_0.
367d c930 3193:    cmp #'0'
367f 9020 3194:    bcc d.uh_oh
3681 c93a 3195:    cmp #':'
3683 9016 3196:    bcc d.got_dig
3685 c941 3197:    cmp #'A'
3687 9018 3198:    bcc d.uh_oh
3689 a948 3199:    lda #'H'
368b 8de762 3200:    sta hex.flag
368e b1c2 3201:    lda (sptr),y
3690 c947 3202:    cmp #'G'
3692 9007 3203:    bcc d.got_dig
3694 c948 3204:    cmp #'H'
3696 d009 3205:    bne d.uh_oh
3698 4ca336 3206:    jmp d.x_0.
369b      3207: ;
369b e8 3208: d.got_dig: inx
369c 20bf3e 3209:    jsr inw.sptr
```

```
369f d0cc 3210:      bne d.next_dig
36a1      3211: ;
36a1 38 3212: d.uh_oh: sec
36a2 60 3213:      rts
36a3      3214: ;
36a3 8a 3215: d.x_0.: txa
36a4 8ef362 3216:      stx xsave
36a7 d002 3217:      bne d.dec_or_hex.
36a9 18 3218:      clc
36aa 60 3219:      rts
36ab      3220: ;
36ab ade762 3221: d.dec_or_hex.: lda hex.flag
36ae c948 3222:      cmp #'H'
36b0 d058 3223:      bne d.dcmptest
36b2      3224: ;
36b2 e005 3225: d.its_hex: cpx #5
36b4 9002 3226:      bcc d.do_hex
36b6 38 3227:      sec
36b7 60 3228:      rts
36b8      3229: ;
36b8 20f736 3230: d.do_hex: jsr dew_sptr
36bb b1c2 3231:      lda (sptr),y
36bd 200037 3232:      jsr sbc_ascii
36c0 85ca 3233:      sta number
36c2 ca 3234:      dex
36c3 f030 3235:      beq d.good_hexit
36c5 20f736 3236:      jsr dew_sptr
36c8 b1c2 3237:      lda (sptr),y
36ca 200037 3238:      jsr sbc_ascii
36cd 0a 3239:      asl
36ce 0a 3240:      asl
36cf 0a 3241:      asl
36d0 0a 3242:      asl
36d1 05ca 3243:      ora number
36d3 85ca 3244:      sta number
36d5 ca 3245:      dex
36d6 f01d 3246:      beq d.good_hexit
36d8 20f736 3247:      jsr dew_sptr
36db b1c2 3248:      lda (sptr),y
36dd 200037 3249:      jsr sbc_ascii
36e0 85cb 3250:      sta number+1
36e2 ca 3251:      dex
36e3 f010 3252:      beq d.good_hexit
36e5 20f736 3253:      jsr dew_sptr
36e8 b1c2 3254:      lda (sptr),y
36ea 200037 3255:      jsr sbc_ascii
36ed 0a 3256:      asl
36ee 0a 3257:      asl
36ef 0a 3258:      asl
36f0 0a 3259:      asl
36f1 05cb 3260:      ora number+1
36f3 85cb 3261:      sta number+1
36f5      3262: ;
```

```

36f5 18 3263: d.good_hexit: clc
36f6 60 3264:      rts
36f7      3265: ;
36f7 a5c2 3266: dew_sptr: lda sptr
36f9 d002 3267:      bne d.A3
36fb c6c3 3268:      dec sptr+1
36fd c6c2 3269: d.A3:   dec sptr
36ff 60 3270:      rts
3700      3271: ;
3700 38 3272: sbc_ascii: sec
3701 e930 3273:      sbc #'0'
3703 c90a 3274:      cmp #10
3705 9002 3275:      bcc d.ascii_dun
3707 e907 3276:      sbc #7
3709 60 3277: d.ascii_dun: rts
370a      3278: ;
370a e006 3279: d.dcm1test: cpx #6
370c 9002 3280:      bcc ascii_to_bin
370e 38 3281:      sec
370f 60 3282:      rts
3710      3283: ;
3710 a966 3284: ascii_to_bin: lda # low inline
3712 85f3 3285:      sta inbuff
3714 a963 3286:      lda # high inline
3716 85f4 3287:      sta inbuff+1
3718 a900 3288:      lda #0
371a 85f2 3289:      sta cix
371c 2000d8 3290:      jsr afp
371f 9001 3291:      bcc d.success
3721 60 3292:      rts
3722      3293: ;
3722 20d2d9 3294: d.success: jsr fpi
3725 a5d4 3295:      lda fr0
3727 85ca 3296:      sta number
3729 a5d5 3297:      lda fr0+1
372b 85cb 3298:      sta number+1
372d b001 3299:      bcs d.noclc
372f 18 3300:      clc
3730 60 3301: d.noclc: rts
3731      3302: ;
3731      3303: ;-----
3731      3304: ;DECIMAL converts the number
3731      3305: ;in the A reg to Decimal.
3731      3306: ;-----
3731      3307: ;
3731      3308: decimal:
3731      3309: ;
3731 85d4 3310:      sta fr0
3733 a900 3311:      lda #0
3735 85d5 3312:      sta fr0+1
3737 20aad9 3313: Adecimal: jsr ifp
373a 4ce6d8 3314:      jmp fasc
373d      3315: ;-----

```



```

373d    3316: ;
373d    3317: opene:
373d    3318: ;
373d    3319: ;      close 0
373d a200 3320:      ldx #0
373f 20823f 3321:      jsr close.r
3742    3322:
3742    3323:
3742 a90c 3324:      lda #12
3744 8d4a03 3325:      sta icaxl
3747 a9c6 3326:      lda # low e
3749 8d4403 3327:      sta icbal
374c a961 3328:      lda # high e
374e 8d4503 3329:      sta icbah
3751 4c723f 3330:      jmp opnly
3754    3331: ;
3754    3332: ;-----
3754    3333: ;Routines to print characters
3754    3334: ;to the screen....
3754    3335: ;
3754    3336: ;CLS clears screen.
3754    3337: ;PUTH prints a '$'.
3754    3338: ;PUTCR prints a carriage return.
3754    3339: ;PUTSPACE prints a space
3754    3340: ;PUTCHAR prints char in A reg.
3754    3341: ;-----
3754    3342: ;
3754 a97d 3343: cls:   lda #cls.
3756 d00c 3344:      bne putchar
3758 205e37 3345: :3crs: jsr putcr
375b 205e37 3346: :2crs: jsr putcr
375e a99b 3347: putcr: lda #cr
3760 d002 3348:      bne putchar
3762 a920 3349: putspace: lda #space
3764    3350: ;
3764 48    3351: putchar: pha      ;Save A and Y
3765 aa    3352:      tax      ;Registers.
3766 98    3353:      tya
3767 48    3354:      pha
3768 8a    3355:      txa
3769 a200 3356:      ldx #0
376b 8ea202 3357:      stx escflg
376e 8e4803 3358:      stx icbll
3771 8e4903 3359:      stx icblh
3774 a00b 3360:      ldy #bput.
3776 8c4203 3361:      sty iccom
3779 2056e4 3362:      jsr ciov
377c 68    3363:      pla      ;Restore Regs.
377d a8    3364:      tay
377e 68    3365:      pla
377f 60    3366:      rts
3780    3367: ;
3780    3368: ;-----

```

```

3780      3369: ;
3780      3370: ;Prints string from memory
3780      3371: ;then jumps over it. Mark
3780      3372: ;end of String with a $FF.
3780      3373: ;-----
3780      3374: ;
3780 68    3375: printsi: pla      ;Get String adr
3781 85c2  3376:      sta sptr  ;from stack.
3783 18    3377:      clc
3784 6901  3378:      adc #1
3786 8d4403 3379:      sta icbal
3789 68    3380:      pla
378a 85c3  3381:      sta sptr+1
378c 6900  3382:      adc #0
378e 8d4503 3383:      sta icbah
3791 a000  3384:      ldy #0    ;Set length
3793 8c4803 3385:      sty icbll ;to zero.
3796 8c4903 3386:      sty icblh
3799 b1c2  3387: d.srch: lda (sptr),y ;Search for
379b c9ff  3388:      cmp #em   ;end of string.
379d f00d  3389:      beq d.eom
379f 20bf3e 3390:      jsr inw.sptr
37a2 ee4803 3391:      inc icbll
37a5 d0f2  3392:      bne d.srch
37a7 ee4903 3393:      inc icblh
37aa d0ed  3394:      bne d.srch
37ac      3395: ;
37ac a5c3  3396: d.eom: lda sptr+1 ;Put new return
37ae 48    3397:      pha      ;adr on stack.
37af a5c2  3398:      lda sptr
37b1 48    3399:      pha
37b2 a200  3400:      ldx #0
37b4 4c653f 3401:      jmp bput
37b7      3402: ;
37b7      3403: ;-----
37b7      3404: ;   Beep routine
37b7      3405: ;-----
37b7      3406: ;
37b7 a912  3407: noise: lda #18
37b9 8d00d2 3408:      sta audf1
37bc a9af  3409:      lda #$af
37be 8d01d2 3410:      sta audc1
37c1 a019  3411:      ldy #25
37c3 20d537 3412:      jsr Await
37c6 8d01d2 3413: d.decay: sta audc1
37c9 a008  3414:      ldy #8
37cb 20d537 3415:      jsr Await
37ce e901  3416:      sbc #1
37d0 c99f  3417:      cmp #$9f
37d2 d0f2  3418:      bne d.decay
37d4 60    3419:      rts
37d5      3420: ;
37d5 ca    3421: Await: dex

```

```

37d6 d0fd 3422:    bne Await
37d8 88 3423:    dey
37d9 d0fa 3424:    bne Await
37db 60 3425:    rts
37dc 3426: ;
37dc 3427: ;-----
37dc 3428: ;GETKEY gets a keypress from
37dc 3429: ;keyboard using channel 4.
37dc 3430: ;Does not echo to screen.
37dc 3431: ;Entry point at GK skips
37dc 3432: ;noise prompt.
37dc 3433: ;-----
37dc 3434: ;
37dc a940 3435: getcap: lda #$40
37de 8dbe02 3436:    sta shflok
37e1 a900 3437:    lda #0
37e3 8db602 3438:    sta invflg
37e6 3439: ;
37e6 20b737 3440: getkey: jsr noise
37e9 a9ff 3441: gk:    lda #$ff
37eb 8dfc02 3442: gk1:   sta ch
37ee 3443: ;    close 4
37ee a240 3444:    ldx #$40
37f0 20823f 3445:    jsr close.r
37f3 3446:
37f3 a9c0 3447:    lda # low k
37f5 9d4403 3448:    sta icbal,x
37f8 a961 3449:    lda # high k
37fa 9d4503 3450:    sta icbah,x
37fd a904 3451:    lda #xread
37ff 9d4a03 3452:    sta icax1,x
3802 20723f 3453:    jsr opnly
3805 a9ff 3454:    lda #em
3807 a000 3455:    ldy #0
3809 3456: ;
3809 844d 3457: d.ch:  sty 77
380b cdfc02 3458:    cmp ch
380e f0f9 3459:    beq d.ch
3810 98 3460:    tya
3811 9d4803 3461:    sta icbll,x
3814 9d4903 3462:    sta icblh,x
3817 207a3f 3463:    jsr bget
381a c99b 3464:    cmp #cr
381c f010 3465:    beq d.kok
381e c91b 3466:    cmp #esc
3820 f00c 3467:    beq d.kok
3822 c97e 3468:    cmp #backsp
3824 f008 3469:    beq d.kok
3826 c97b 3470:    cmp #$7b
3828 b0bc 3471:    bcs getkey
382a c920 3472:    cmp #space
382c 90b8 3473:    bcc getkey
382e 3474: ;

```

```

382e 60 3475: d.kok: rts
382f 3476: ;-----
382f 3477: ;
382f 20e637 3478: getreturn: jsr getkey
3832 c99b 3479: cmp #cr
3834 d0f9 3480: bne getreturn
3836 60 3481: rts
3837 3482: ;
3837 3483: ;-----
3837 3484: ;
3837 20dc37 3485: checkkey: jsr getcap
383a c91b 3486: cmp #esc
383c d009 3487: bne d.isitcr
383e 208037 3488: jsr printsi
3841 1c9c9c 3489: dc.b $1c,$9c,$9c,em
3845 38 3490: sec
3846 60 3491: rts
3847 3492: ;
3847 c99b 3493: d.isitcr: cmp #cr
3849 d0ec 3494: bne checkkey
384b 18 3495: clc
384c 60 3496: rts
384d 3497: ;-----
384d 3498: ;
384d 3499: ;GETLINE gets a line of input
384d 3500: ;from keyboard and echos to
384d 3501: ;screen. Enter with maximum
384d 3502: ;number of characters to be
384d 3503: ;allowed in the A register.
384d 3504: ;Input is terminated by a
384d 3505: ;carriage return.
384d 3506: ;-----
384d 3507: ;
384d 3508: ;
384d a905 3509: getnum: lda #5
384f 8dda62 3510: getline: sta maxlen
3852 a900 3511: lda #0 ;Turn on
3854 8df002 3512: sta crsinh ;the cursor.
3857 208037 3513: jsr printsi
385a 2020ff 3514: dc.b " ",EM
385d acda62 3515: ldy maxlen
3860 3516: ;
3860 a99b 3517: d.clr_buf: lda #cr ;Clear input
3862 20a838 3518: jsr Aaddchar ;buffer
3865 88 3519: dey
3866 10f8 3520: bpl d.clr_buf
3868 20b737 3521: jsr noise ;Beep
386b a000 3522: ldy #0
386d 3523: ;
386d 84cd 3524: d.nxt_letr: sty ytemp ;Save index.
386f 20e937 3525: jsr gk ;Get a character.
3872 a4cd 3526: ldy ytemp ;Restore index.
3874 c91b 3527: cmp #esc

```

```

3876 d004 3528:      bne d.bsp.
3878 eef002 3529:      inc crsinh
387b 60 3530:      rts
387c      3531: ;
387c c97e 3532: d.bsp.: cmp #backsp
387e f012 3533:      beq d.bak_spce
3880 c99b 3534:      cmp #cr
3882 f01d 3535:      beq d.gotline
3884 ccda62 3536:      cpy maxlen
3887 f0e4 3537:      beq d.nxt_letr
3889 206437 3538:      jsr putchar
388c 20a838 3539:      jsr Aaddchar
388f c8 3540:      iny
3890 d0db 3541:      bne d.nxt_letr
3892      3542: ;
3892 c000 3543: d.bak_spce: cpy #0 ;Handle a
3894 f0d7 3544:      beq d.nxt_letr ;backspace.
3896 206437 3545:      jsr putchar
3899 88 3546:      dey
389a a99b 3547:      lda #cr
389c 20a838 3548:      jsr Aaddchar
389f 30cc 3549:      bmi d.nxt_letr
38a1      3550: ;
38a1 88 3551: d.gotline: dey
38a2 8df002 3552:      sta crsinh
38a5 4c6237 3553:      jmp putspace
38a8      3554: ;
38a8 996663 3555: Aaddchar: sta inline,y ;Add input.
38ab 60 3556:      rts
38ac      3557: ;-----
38ac      3558: ;
38ac ade062 3559: sho_address: lda address
38af c920 3560:      cmp #space
38b1 d001 3561:      bne shopc
38b3 60 3562:      rts
38b4      3563: ;
38b4 a905 3564: shopc:  lda #5
38b6 9d4803 3565:      sta icbll,x
38b9 a900 3566:      lda #0
38bb 9d4903 3567:      sta icblh,x
38be a9e0 3568:      lda # low address
38c0 9d4403 3569:      sta icbal,x
38c3 a962 3570:      lda # high address
38c5 9d4503 3571:      sta icbah,x
38c8 4c653f 3572:      jmp bput
38cb      3573: ;
38cb      3574: ;-----
38cb      3575: ;
38cb 6c3663 3576: to_lino: jmp (to.lino)
38ce      3577: ;
38ce      3578: ;-----
38ce      3579: ;PRINT_LINE prints a line of
38ce      3580: ;up to 40 characters to

```

```
38ce 3581: ;whichever channel the X reg
38ce 3582: ;is pointing to. Address of
38ce 3583: ;string must be in LINE.PTR.
38ce 3584: ;Location 77 is poked with a
38ce 3585: ;zero here to keep the screen
38ce 3586: ;from going to attract mode.
38ce 3587: ;-----
38ce 3588: ;
38ce 3589: print_line:
38ce 3590: ;
38ce ad2163 3591:     lda line.ptr
38d1 9d4403 3592:     sta icbal,x
38d4 ad2263 3593:     lda line.ptr+1
38d7 9d4503 3594:     sta icbah,x
38da a909 3595:     lda #putrec.
38dc 9d4203 3596:     sta iccom,x
38df a93c 3597:     lda #60
38e1 9d4803 3598:     sta icbll,x
38e4 a900 3599:     lda #0
38e6 9d4903 3600:     sta icblh,x
38e9 854d 3601:     sta 77
38eb 4c56e4 3602:     jmp ciov
38ee 3603: ;
38ee 3604: ;
38ee 3605: ;-----
38ee 3606: ;WRITE_LINE writes a line of
38ee 3607: ;output to whichever devices
38ee 3608: ;are active. Devices looked
38ee 3609: ;at are Disk, Screen, and
38ee 3610: ;Printer. The line is written
38ee 3611: ;using PRINT_LINE above.
38ee 3612: ;-----
38ee 3613: ;
38ee 3614: ;
38ee adfc02 3615: write_line: lda ch
38f1 c9ef 3616:     cmp #239 ;SHFT^Q
38f3 d008 3617:     bne d.write.1
38f5 a922 3618:     lda #$22
38f7 8d2f02 3619:     sta sdmctl
38fa 4c8421 3620:     jmp start
38fd 3621: ;
38fd 3622: ;
38fd ad9b61 3623: d.write.1: lda lino.flag ;See whether to
3900 f003 3624:     beq d.which. ;write a line #.
3902 20cb38 3625:     jsr to_lino
3905 3626: ;
3905 adfe62 3627: d.which.: lda dskon
3908 f003 3628:     beq d.screen
390a 202a3f 3629:     jsr chk_out
390d 3630: ;
390d ad2f02 3631: d.screen: lda sdmctl ;Screen on?
3910 f008 3632:     beq d.disk ;No.
3912 a200 3633:     idx #0
```

```
3914 20ac38 3634:      jsr sho_address ;Print PC.
3917 20ce38 3635:      jsr print_line ;Print line.
391a adfe62 3636: d.disk: lda dskon
391d f035 3637:      beq d.pntr
391f a000 3638:      ldy #0
3921 ae2363 3639:      ldx linesize
3924 ad2163 3640:      lda line.ptr
3927 85c2 3641:      sta sptr
3929 ad2263 3642:      lda line.ptr+1
392c 85c3 3643:      sta sptr+1
392e b1c2 3644: d.lenloop: lda (sptr),y
3930 9d5e66 3645:      sta drvbuf,x
3933 c99b 3646:      cmp #cr
3935 f004 3647:      beq d.gotlen
3937 e8 3648:      inx
3938 c8 3649:      iny
3939 d0f3 3650:      bne d.lenloop
393b      3651: ;
393b c8 3652: d.gotlen: iny
393c 98 3653:      tya
393d 18 3654:      clc
393e 6d2363 3655:      adc linesize
3941 8d2363 3656:      sta linesize
3944 ad2463 3657:      lda linesize+1
3947 6900 3658:      adc #0
3949 8d2463 3659:      sta linesize+1
394c ad2363 3660:      lda linesize
394f 1003 3661:      bpl d.pntr
3951 20a039 3662:      jsr putblock
3954      3663: ;
3954 a230 3664: d.pntr: ldx #chan3
3956 ad1c63 3665:      lda close.p. ;Option toggled?
3959 f032 3666:      beq d.close_pntr ;Yes.
395b ad1b63 3667:      lda is.p.opn. ;Printer open?
395e d024 3668:      bne d.pntr_is_opn ;Yes
3960 a9c3 3669:      lda # low p
3962 9d4403 3670:      sta icbal,x
3965 a961 3671:      lda # high p
3967 9d4503 3672:      sta icbah,x
396a 206d3f 3673:      jsr ciopen
396d 100d 3674:      bpl d.popnok
396f a9fd 3675:      lda #$fd ;UHOH! Ring the
3971 206437 3676:      jsr putchar ;lousy Bell!
3974 a200 3677:      ldx #0 ;Then close the
3976 8e1c63 3678:      stx close.p. ;cottonpicker!
3979 4c9539 3679:      jmp d.closep
397c      3680: ;
397c a9ff 3681: d.popnok: lda #$ff ;Tell us it's
397e 8d1b63 3682:      sta is.p.opn. ;open
3981 4c9f39 3683:      jmp d.pdone
3984      3684: ;
3984 20b438 3685: d.pntr_is_opn: jsr shope
3987 20ce38 3686:      jsr print_line
```

```
398a 4c9f39 3687:      jmp d.pdone
398d      3688:      ;
398d ad1b63 3689: d.close_prntr: lda is.p.opn.
3990 f00d 3690:      beq d.pdone
3992 20ce38 3691:      jsr print_line
3995      3692:      ;
3995      3693: d.closep:
3995      3694:      ; close 3
3995 a230 3695:      ldx #$30
3997 20823f 3696:      jsr close.r
399a      3697:
399a a900 3698:      lda #0
399c 8d1b63 3699:      sta is.p.opn.
399f 60 3700: d.pdone: rts
39a0      3701: ;-----
39a0      3702: ;
39a0 ad2363 3703: putblock: lda linesize
39a3 0d2463 3704:      ora linesize+1
39a6 f04c 3705:      beq d.blockput
39a8      3706: ;
39a8 202a3f 3707: d.putblk: jsr chk_out
39ab a220 3708: d.retry: ldx #chan2
39ad a95e 3709:      lda # low drvbuf
39af 9d4403 3710:      sta icbal,x
39b2 a966 3711:      lda # high drvbuf
39b4 9d4503 3712:      sta icbah,x
39b7 ad2363 3713:      lda linesize
39ba 9d4803 3714:      sta icbll,x
39bd ad2463 3715:      lda linesize+1
39c0 9d4903 3716:      sta icblh,x
39c3 20653f 3717:      jsr bput
39c6 302d 3718:      bmi d.bady
39c8 a900 3719:      lda #0
39ca 8d2363 3720:      sta linesize
39cd 8d2463 3721:      sta linesize+1
39d0 18 3722:      clc
39d1 adff62 3723:      lda files.size ;Add lines
39d4 6d6803 3724:      adc dbsize ;length to
39d7 8dff62 3725:      sta files.size ;Filesize.
39da ad0063 3726:      lda files.size+1
39dd 6d6903 3727:      adc dbsize+1
39e0 8d0063 3728:      sta files.size+1
39e3      3729:      ;
39e3 adff62 3730:      lda files.size ;is room
39e6 ed9d61 3731:      sbc max.fsize ;left in
39e9 ad0063 3732:      lda files.size+1 ;this file.
39ec ed9e61 3733:      sbc max.fsize+1
39ef 9003 3734:      bcc d.blockput ;There is.
39f1 20013a 3735:      jsr opn_nxt_file ;isn't.
39f4      3736:      ;
39f4 60 3737: d.blockput: rts
39f5      3738:      ;
39f5 c0a2 3739: d.bady: cpy #$a2
```



```
39f7 d005 3740:      bne d.killit
39f9 20ae41 3741:      jsr newdisk
39fc 10ad 3742:      bpl d.retry
39fe      3743: ;
39fe 4cca43 3744: d.killit: jmp io_err
3a01      3745: ;-----
3a01      3746: ;
3a01      3747: opn_nxt_file:
3a01      3748: ;
3a01 20583d 3749:      jsr nxt_outfn
3a04 ad9c61 3750:      lda asmflg
3a07 d011 3751:      bne d.closeit
3a09      3752: ;
3a09 a900 3753: d.dotry: lda #0
3a0b 8d0063 3754:      sta files.size+1
3a0e 20ee38 3755:      jsr write_line
3a11 202f3c 3756:      jsr twospcs
3a14 20533c 3757:      jsr get_link
3a17 20ee38 3758:      jsr write_line
3a1a      3759: ;
3a1a 20a039 3760: d.closeit: jsr putblock
3a1d 205a3f 3761: d.noput:  jsr closedisk
3a20 202a3f 3762: d.opn_it: jsr chk_out
3a23 3001 3763:      bmi d.opnerr
3a25 60 3764:      rts
3a26      3765: ;
3a26 c0a2 3766: d.opnerr: cpy #$a2
3a28 f003 3767:      beq d.fixa2
3a2a 4cca43 3768:      jmp io_err
3a2d      3769: ;
3a2d 20ae41 3770: d.fixa2:  jsr newdisk
3a30 4c093a 3771:      jmp d.dotry
3a33      3772: ;-----
3a33      3773: ;
3a33 adfe62 3774: opn_disk: lda dskon
3a36 f022 3775:      beq d.opnoop
3a38 a220 3776:      ldx #chan2
3a3a a9ae 3777:      lda # low outfn
3a3c 9d4403 3778:      sta icbal,x
3a3f a963 3779:      lda # high outfn
3a41 9d4503 3780:      sta icbah,x
3a44 206d3f 3781:      jsr ciopen
3a47 3011 3782:      bmi d.opnoop
3a49 eedd62 3783:      inc dskopnflg
3a4c eedc62 3784:      inc totfiles
3a4f a900 3785:      lda #0
3a51 8dd462 3786:      sta try
3a54 8dff62 3787:      sta files.size
3a57 8d0063 3788:      sta files.size+1
3a5a 60 3789: d.opnoop: rts
3a5b      3790: ;
3a5b      3791: ;-----
3a5b      3792: ;
```

```
3a5b      3793: ;Load non-binary file.
3a5b      3794: ;
3a5b a210 3795: bload:  ldx #chan1
3a5d      3796: ; CLC
3a5d a5b4 3797:      lda buftop
3a5f      3798: ; ADC #1
3a5f 9d4403 3799:      sta icbal,x
3a62 a5b5 3800:      lda buftop+1
3a64      3801: ; ADC #0
3a64 9d4503 3802:      sta icbah,x
3a67 a5b6 3803:      lda bufsize
3a69 9d4803 3804:      sta icbll,x
3a6c a5b7 3805:      lda bufsize+1
3a6e 9d4903 3806:      sta icblh,x
3a71 a907 3807:      lda #bget.
3a73 9d4203 3808:      sta iccom,x
3a76 2056e4 3809:      jsr ciov
3a79 8cde62 3810:      sty cioy
3a7c 1007 3811:      bpl d.goodbl
3a7e c088 3812:      cpy #$88
3a80 f003 3813:      beq d.goodbl
3a82 4cca43 3814:      jmp io_err
3a85      3815: ;
3a85 60 3816: d.goodbl: rts
3a86      3817: ;-----
3a86      3818: ;
3a86      3819: ;Load .ORG address from
3a86      3820: ;input file into seg.org.
3a86      3821: ;
3a86 a210 3822: load_adr: ldx #chan1
3a88 a9fa 3823:      lda # low seg.org
3a8a 9d4403 3824:      sta icbal,x
3a8d a962 3825:      lda # high seg.org
3a8f 9d4503 3826:      sta icbah,x
3a92 a907 3827:      lda #bget.
3a94 9d4203 3828:      sta iccom,x
3a97 a902 3829:      lda #2
3a99 9d4803 3830:      sta icbll,x
3a9c a900 3831:      lda #0
3a9e 9d4903 3832:      sta icblh,x
3aa1 2056e4 3833: d.get2:  jsr ciov
3aa4 1003 3834:      bpl d.2ok
3aa6 4cb92f 3835:      jmp b.f_err
3aa9      3836: ;
3aa9 adfa62 3837: d.2ok:  lda seg.org ;$FFFF adr
3aac 2dfb62 3838:      and seg.org+1 ;header?
3aaf c9ff 3839:      cmp #$ff
3ab1 60 3840:      rts
3ab2      3841: ;
3ab2 a9fc 3842: load_end: lda # low seg.end ;Get end
3ab4 9d4403 3843:      sta icbal,x ;address of segment.
3ab7 a962 3844:      lda # high seg.end
3ab9 9d4503 3845:      sta icbah,x
```

```
3abc 2056e4 3846:      jsr ciov
3abf 1003 3847:      bpl d.goodadr
3ac1 4cb92f 3848:      jmp b.f_err
3ac4      3849:      ;
3ac4 60 3850: d.goodadr: rts
3ac5      3851: ;-----
3ac5      3852: ;
3ac5      3853: ;Load as much of file segment
3ac5      3854: ;as will fit in buffer.
3ac5      3855: ;
3ac5 a210 3856: fload: ldx #chan1
3ac7 ad1f63 3857:      lda left.in.buf
3aca 9d4803 3858:      sta icbll,x
3acd ad2063 3859:      lda left.in.buf+1
3ad0 9d4903 3860:      sta icblh,x
3ad3 38 3861:      sec
3ad4 a5b6 3862:      lda bufsize
3ad6 ed1f63 3863:      sbc left.in.buf
3ad9 a5b7 3864:      lda bufsize+1
3adb ed2063 3865:      sbc left.in.buf+1
3ade b00a 3866:      bcs d.got_flen
3ae0 a5b6 3867:      lda bufsize
3ae2 9d4803 3868:      sta icbll,x
3ae5 a5b7 3869:      lda bufsize+1
3ae7 9d4903 3870:      sta icblh,x
3aea a907 3871: d.got_flen: lda #bget.
3aec 9d4203 3872:      sta iccom,x
3aef 18 3873:      clc
3af0 a5b4 3874:      lda buftop
3af2 6901 3875:      adc #1
3af4 9d4403 3876:      sta icbal,x
3af7 a5b5 3877:      lda buftop+1
3af9 6900 3878:      adc #0
3afb 9d4503 3879:      sta icbah,x
3afe 2056e4 3880:      jsr ciov
3b01 8cde62 3881:      sty cioy ;Save CIO status.
3b04 60 3882:      rts
3b05      3883: ;-----
3b05      3884: ;
3b05      3885: ;
3b05      3886: open_in:
3b05      3887: ; close 1
3b05 a210 3888:      ldx #$10
3b07 20823f 3889:      jsr close.r
3b0a      3890:
3b0a a98e 3891:      lda # low infn
3b0c 9d4403 3892:      sta icbal,x
3b0f a963 3893:      lda # high infn
3b11 9d4503 3894:      sta icbah,x
3b14 a904 3895:      lda #xread
3b16 9d4a03 3896:      sta icax1,x
3b19 a903 3897:      lda #open.
3b1b 9d4203 3898:      sta iccom,x
```

```

3b1e 2056e4 3899:    jsr ciov
3b21 1003 3900:    bpl d.inok
3b23 4cca43 3901:    jmp io_err
3b26      3902:    ;
3b26 8cde62 3903: d.inok:  sty ciov
3b29 a201 3904:    ldx #1
3b2b 8edd62 3905:    stx dskopnflg
3b2e ca 3906:    dex
3b2f 8eef62 3907:    stx which.bytek
3b32 60 3908:    rts
3b33      3909: ;-----
3b33      3910:
3b33      3911: ; .PAGE "Misc Subroutines"
3b33      3912: ; .include misc
3b33      3913: ; ... MISC ...
3b33      3914: ;
3b33      3915: ;-----
3b33      3916: ;VBI routine checks to see if
3b33      3917: ;START or OPTION is pressed.
3b33      3918: ;START flips screen on/off.
3b33      3919: ;OPTION flips printer on/off.
3b33      3920: ;-----
3b33 ad1fd0 3921: vbi:    lda consol
3b36 2905 3922:    and #5
3b38 4905 3923:    eor #5
3b3a d00b 3924:    bne d.button_pressed
3b3c adf962 3925:    lda vbi.timer ;Built in
3b3f f031 3926:    beq xitvbi ;debounce.
3b41 cef962 3927:    dec vbi.timer
3b44 4c723b 3928:    jmp xitvbi
3b47      3929: ;
3b47 8df762 3930: d.button_pressed: sta button
3b4a adf962 3931:    lda vbi.timer
3b4d d023 3932:    bne xitvbi
3b4f a905 3933:    lda #5 ;Set timer.
3b51 8df962 3934:    sta vbi.timer
3b54 adf762 3935:    lda button ;See which button
3b57 c904 3936:    cmp #4
3b59 d008 3937:    bne d.start_buttn
3b5b ad1c63 3938:    lda close.p. ;It's OPTION so
3b5e 49ff 3939:    eor #$ff ;flip printer flag.
3b60 8d1c63 3940:    sta close.p.
3b63 adf762 3941: d.start_buttn: lda button
3b66 c901 3942:    cmp #1
3b68 d008 3943:    bne xitvbi
3b6a ad2f02 3944:    lda sdmctl ;It's START so
3b6d 4922 3945:    eor #$22 ;flip screen.
3b6f 8d2f02 3946:    sta sdmctl
3b72 4c62e4 3947: xitvbi: jmp xitvbv
3b75      3948: ;
3b75      3949: ;-----
3b75      3950: ;IR_AWAKE lets the user know
3b75      3951: ;that something is happening

```

```

3b75      3952: ;while the symbol table is
3b75      3953: ;is being built.
3b75      3954: ;-----
3b75      3955: ;
3b75      3956: ir_awake:
3b75      3957: ;
3b75 adfc02 3958:      lda ch
3b78 c9ef 3959:      cmp #239
3b7a d003 3960:      bne d.scr.
3b7c 4c8421 3961:      jmp start
3b7f ad2f02 3962: d.scr.: lda sdmctl
3b82 f035 3963:      beq d.waked
3b84 ee9261 3964: d.swtch: inc mymr ;This instruction
3b87 ad9261 3965:      lda mymr ;is flipped from
3b8a d007 3966:      bne d.mymrhi ;INC to DEC
3b8c a2ee 3967:      ldx #$ee ;and back as
3b8e 8e843b 3968:      stx d.swtch ;the printed line
3b91 d009 3969:      bne d.mrok ;gets to one
3b93 c911 3970: d.mymrhi: cmp #17 ;side or the
3b95 d005 3971:      bne d.mrok ;other of the
3b97 a2ce 3972:      ldx #$ce ;screen.
3b99 8e843b 3973:      stx d.swtch
3b9c 8555 3974: d.mrok: sta colcrs
3b9e 208037 3975:      jsr printsi ;Print the line.
3ba1 427569 3976: ir:      dc.b "Building Symbol Table.",CR,EM
3bb9 60 3977: d.waked: rts
3bba      3978: ;
3bba      3979: ;-----
3bba      3980: ;NORMALIZE will put 'Dn:' in
3bba      3981: ;front of a filename if it
3bba      3982: ;isn't already there. It uses
3bba      3983: ;whatever is in DN so change
3bba      3984: ;the default drive by changing
3bba      3985: ;DN+1 to desired drive #.
3bba      3986: ;Enter with the users filename
3bba      3987: ;in INLINE, the hi byte of the
3bba      3988: ;address where the legal file-
3bba      3989: ;name is to be stored in the
3bba      3990: ;Y reg and the lo byte in the
3bba      3991: ;A reg.
3bba      3992: ;-----
3bba      3993: ;
3bba      3994: normalize:
3bba      3995: ;
3bba 8de93b 3996:      sta d.tl+1 ;Set address
3bbd 8df63b 3997:      sta d.tl1+1 ;of filename.
3bc0 8d023c 3998:      sta d.tl2+1
3bc3 8cea3b 3999:      sty d.tl+2
3bc6 8cf73b 4000:      sty d.tl1+2
3bc9 8c033c 4001:      sty d.tl2+2
3bcc a000 4002:      ldy #0 ;Check to see
3bce ad6863 4003:      lda inline+2 ;if user
3bd1 c93a 4004:      cmp #':' ;entered 'Dn:'.

```

```
3bd3 f01b 4005:      beq d.fcont
3bd5 ad6763 4006:     lda inline+1
3bd8 c93a 4007:      cmp #':'
3bda d007 4008:      bne d.fchkd
3bdc ad6663 4009:     lda inline
3bdf c944 4010:      cmp #'D'
3be1 d00d 4011:      bne d.fcont
3be3 a202 4012: d.fchkd: ldx #2      ;He didn't.
3be5 bdc961 4013: d.fdnloop: lda dn,x
3be8 9dffff 4014: d.tl:   sta dummy,x
3beb ca 4015:      dex
3bec 10f7 4016:      bpl d.fdnloop
3bee a003 4017:      ldy #3
3bf0      4018: ;
3bf0 a200 4019: d.fcont: ldx #0
3bf2 bd6663 4020: d.floop: lda inline,x ;Copy the
3bf5 99ffff 4021: d.tl1:   sta dummy,y ;Users filename.
3bf8 c99b 4022:      cmp #cr
3bfa f004 4023:      beq d.fndone
3bfc e8 4024:      inx
3bfd c8 4025:      iny
3bfe d0f2 4026:      bne d.floop
3c00      4027: ;
3c00 18 4028: d.fndone: clc
3c01 adffff 4029: d.tl2:   lda dummy
3c04 c945 4030:      cmp #'E'
3c06 f004 4031:      beq d.fnsec
3c08 c953 4032:      cmp #'S'
3c0a d001 4033:      bne d.fnxit
3c0c 38 4034: d.fnsec: sec
3c0d 60 4035: d.fnxit: rts
3c0e      4036: ;-----
3c0e      4037: ;
3c0e      4038: misc_line:
3c0e      4039: ;
3c0e 20333c 4040:      jsr move_line
3c11 4cee38 4041:      jmp write_line
3c14      4042: ;
3c14      4043: dash:
3c14      4044: ;
3c14 a93c 4045:      lda # low dash_m
3c16 a062 4046:      ldy # high dash_m
3c18 a200 4047:      ldx #0
3c1a 200e3c 4048:      jsr misc_line
3c1d      4049: ;
3c1d a900 4050: onespace: lda #0
3c1f 85cc 4051: spacer: sta spaces
3c21 a95a 4052:      lda # low spacer_m
3c23 a062 4053:      ldy # high spacer_m
3c25 a200 4054:      ldx #0
3c27 200e3c 4055:      jsr misc_line
3c2a c6cc 4056:      dec spaces
3c2c 10f1 4057:      bpl spacer
```

```

3c2e 60 4058:      rts
3c2f      4059: ;
3c2f a901 4060: twospc: lda #1
3c31 d0ec 4061:      bne spacer
3c33      4062: ;-----
3c33      4063: ;
3c33      4064: move_line:
3c33      4065: ;
3c33 8d453c 4066:      sta from+1
3c36 8c463c 4067:      sty from+2
3c39 8ef362 4068:      stx xsave
3c3c      4069: ;
3c3c 20372c 4070:      jsr clr_printline
3c3f aef362 4071:      ldx xsave
3c42 a000 4072: move:  ldy #0
3c44 b9ffff 4073: from:  lda dummy,y
3c47 9d3e63 4074:      sta printline,x
3c4a c99b 4075:      cmp #cr
3c4c f004 4076:      beq d.exit_moov
3c4e e8 4077:      inx
3c4f c8 4078:      iny
3c50 d0f2 4079:      bne from
3c52      4080: ;
3c52 60 4081: d.exit_moov: rts
3c53      4082: ;-----
3c53      4083: ;
3c53      4084: get_link:
3c53      4085: ;
3c53 a987 4086:      lda # low link_m
3c55 a062 4087:      ldy # high link_m
3c57 a207 4088:      ldx #7
3c59 20333c 4089:      jsr move_line
3c5c a920 4090:      lda #space
3c5e 9d3e63 4091:      sta printline,x
3c61 ca 4092:      dex
3c62 bd3e63 4093:      lda printline,x
3c65 8d3563 4094:      sta incend
3c68 a000 4095:      ldy #0
3c6a e8 4096:      inx
3c6b 8a 4097:      txa
3c6c 48 4098:      pha
3c6d a9ae 4099:      lda # low outfn
3c6f 8d453c 4100:      sta from+1
3c72 a963 4101:      lda # high outfn
3c74 8d463c 4102:      sta from+2
3c77 20423c 4103:      jsr move
3c7a ad3563 4104:      lda incend
3c7d c922 4105:      cmp #""
3c7f d008 4106:      bne d.lnkdone
3c81 9d3e63 4107:      sta printline,x
3c84 a99b 4108:      lda #cr
3c86 9d3f63 4109:      sta printline+1,x
3c89 68 4110: d.lnkdone: pla

```

```
3c8a a8 4111:    tay
3c8b adcd61 4112:    lda da
3c8e 993f63 4113:    sta printline+1,y
3c91 60 4114:    rts
3c92 4115: ;
3c92 4116: ;-----
3c92 4117: ;
3c92 4118: compute_address:
3c92 4119: ;
3c92 a200 4120:    ldx #0
3c94 a001 4121:    ldy #1
3c96 b9c600 4122: d.adr_loop: lda pc,y
3c99 4a 4123:    lsr
3c9a 4a 4124:    lsr
3c9b 4a 4125:    lsr
3c9c 4a 4126:    lsr
3c9d 201336 4127:    jsr hex_nibble
3ca0 9de062 4128:    sta address,x
3ca3 e8 4129:    inx
3ca4 b9c600 4130:    lda pc,y
3ca7 290f 4131:    and #$0f
3ca9 201336 4132:    jsr hex_nibble
3cac 9de062 4133:    sta address,x
3caf e8 4134:    inx
3cb0 88 4135:    dey
3cb1 10e3 4136:    bpl d.adr_loop
3cb3 a920 4137:    lda #space
3cb5 9de062 4138:    sta address,x
3cb8 60 4139:    rts
3cb9 4140: ;-----
3cb9 4141: ;
3cb9 4142: sho_org:
3cb9 4143: ;
3cb9 addd62 4144:    lda dskopnflg
3cbc d008 4145:    bne d.orgit
3cbe 202a3f 4146:    jsr chk_out
3cc1 1003 4147:    bpl d.orgit
3cc3 4cca43 4148:    jmp io_err
3cc6 4149: ;
3cc6 a006 4150: d.orgit: ldy #6
3cc8 b93e63 4151: d.mvplin: lda printline,y
3ccb 996663 4152:    sta inline,y
3cce 88 4153:    dey
3ccf 10f7 4154:    bpl d.mvplin
3cd1 201d3c 4155:    jsr onespace
3cd4 ad6663 4156:    lda inline
3cd7 c920 4157:    cmp #space
3cd9 f016 4158:    beq d.ovrmv
3cdb a006 4159:    ldy #6
3cdd b96663 4160: d.rvplin: lda inline,y
3ce0 993e63 4161:    sta printline,y
3ce3 88 4162:    dey
3ce4 10f7 4163:    bpl d.rvplin
```



```

3ce6 a99b 4164:    lda #cr
3ce8 8d4563 4165:    sta printline+7
3ceb 20ee38 4166:    jsr write_line
3cee 201d3c 4167:    jsr onespace
3cf1 201d3c 4168: d.ovrmv: jsr onespace
3cf4 20372c 4169:    jsr clr_printline
3cf7 a000 4170:    ldy #0
3cf9 a207 4171:    ldx #7
3cfb b96462 4172: d.orgmloop: lda org_m,y
3cfe c99b 4173:    cmp #cr
3d00 f007 4174:    beq d.orgliz
3d02 9d3e63 4175:    sta printline,x
3d05 e8 4176:    inx
3d06 c8 4177:    iny
3d07 d0f2 4178:    bne d.orgmloop
3d09 4179: ;
3d09 e8 4180: d.orgliz: inx
3d0a e8 4181:    inx
3d0b a924 4182:    lda #'$'
3d0d 9d3e63 4183:    sta printline,x
3d10 e8 4184:    inx
3d11 a001 4185:    ldy #1
3d13 b9c600 4186: d.orgloop: lda pc,y
3d16 992663 4187:    sta lastorg,y
3d19 20fc35 4188:    jsr byte_to_hex
3d1c 88 4189:    dey
3d1d 10f4 4190:    bpl d.orgloop
3d1f 4191: ;
3d1f a99b 4192: d.orged: lda #cr
3d21 9d3e63 4193:    sta printline,x
3d24 20ee38 4194:    jsr write_line
3d27 4c2f3c 4195:    jmp twospes
3d2a 4196: ;-----
3d2a 4197: ;
3d2a 4198: get_lino:
3d2a 4199: ;
3d2a a204 4200:    ldx #4
3d2c bd3863 4201:    lda lnum,x
3d2f c930 4202:    cmp #'0'
3d31 d006 4203:    bne d.d0
3d33 a935 4204:    lda #'5'
3d35 9d3863 4205:    sta lnum,x
3d38 60 4206:    rts
3d39 4207: ;
3d39 a930 4208: d.d0:  lda #'0'
3d3b 9d3863 4209:    sta lnum,x
3d3e ca 4210:    dex
3d3f d002 4211:    bne d.linoloop
3d41 4212: ;
3d41 a204 4213: inc_lino: ldx #4
3d43 4214: ;
3d43 bc3863 4215: d.linoloop: ldy lnum,x
3d46 c8 4216:    iny

```

```

3d47 c03a 4217:    cpy #'.'
3d49 9008 4218:    bcc d.lino_ok
3d4b a930 4219:    lda #'0'
3d4d 9d3863 4220:    sta lnum,x
3d50 ca 4221:    dex
3d51 10f0 4222:    bpl d.linoloop
3d53 98 4223: d.lino_ok: tya
3d54 9d3863 4224:    sta lnum,x
3d57 60 4225:    rts
3d58 4226: ;-----
3d58 4227: ;
3d58 4228: nxt_outfn:
3d58 4229: ;
3d58 add462 4230:    lda try
3d5b f001 4231:    beq d.newext
3d5d 60 4232:    rts
3d5e 4233: ;
3d5e eed462 4234: d.newext: inc try
3d61 aee962 4235:    ldx outdot
3d64 e8 4236:    inx
3d65 e8 4237:    inx
3d66 bcae63 4238: d.incloop: ldy outfn,x
3d69 c8 4239:    iny
3d6a c03a 4240:    cpy #'.'
3d6c 900b 4241:    bcc d.storinc
3d6e a930 4242:    lda #'0'
3d70 9dae63 4243:    sta outfn,x
3d73 ca 4244:    dex
3d74 ece962 4245:    cpx outdot
3d77 d0ed 4246:    bne d.incloop
3d79 98 4247: d.storinc: tya
3d7a 9dae63 4248:    sta outfn,x
3d7d 60 4249:    rts
3d7e 4250: ;-----
3d7e 4251:
3d7e 4252:
3d7e 4253: ; .PAGE "Mini Subroutines"
3d7e 4254: ; .include mini
3d7e 4255: ; ... MINI ...
3d7e 4256: ;
3d7e 38 4257: no_more: sec
3d7f a900 4258: zmore: lda #0
3d81 8df262 4259:    sta more.
3d84 60 4260:    rts
3d85 4261: ;-----
3d85 4262: ;
3d85 18 4263: say_more: clc
3d86 a901 4264: imore: lda #1
3d88 8df262 4265:    sta more.
3d8b 60 4266:    rts
3d8c 4267: ;-----
3d8c 4268: ;
3d8c ad0663 4269: optolbl: lda operand

```

```

3d8f 85ba 4270:    sta label.adr
3d91 ad0763 4271:    lda operand+1
3d94 85bb 4272:    sta label.adr+1
3d96 60 4273:    rts
3d97 4274: ;-----
3d97 4275: ;
3d97 a5c6 4276: setprogc: lda pc
3d99 85ba 4277:    sta label.adr
3d9b a5c7 4278:    lda pc+1
3d9d 85bb 4279:    sta label.adr+1
3d9f 60 4280:    rts
3da0 4281: ;-----
3da0 4282: ;
3da0 ad0363 4283: orleft: lda left.to.do
3da3 0d0463 4284:    ora left.to.do+1
3da6 60 4285:    rts
3da7 4286: ;-----
3da7 4287: ;
3da7 38 4288: setndx: sec
3da8 a5b0 4289:    lda buffer
3daa 85b2 4290:    sta bufptr
3dac e901 4291:    sbc #1
3dae 85b4 4292:    sta buftop
3db0 a5b1 4293:    lda buffer+1
3db2 85b3 4294:    sta bufptr+1
3db4 e900 4295:    sbc #0
3db6 85b5 4296:    sta buftop+1
3db8 60 4297:    rts
3db9 4298: ;-----
3db9 4299: ;
3db9 38 4300: setbsize: sec
3dba adf562 4301:    lda freehi
3dbd e5b4 4302:    sbc buftop
3dbf 85b6 4303:    sta bufsz
3dc1 adf662 4304:    lda freehi+1
3dc4 e5b5 4305:    sbc buftop+1
3dc6 85b7 4306:    sta bufsz+1
3dc8 60 4307:    rts
3dc9 4308: ;-----
3dc9 4309: ;
3dc9 38 4310: docarry: sec
3dca a5b5 4311:    lda buftop+1
3dcc e5b1 4312:    sbc buffer+1
3dce f002 4313:    beq d.lo.
3dd0 b00a 4314:    bcs d.clc
3dd2 a5b4 4315: d.lo.: lda buftop
3dd4 e5b0 4316:    sbc buffer
3dd6 f002 4317:    beq d.sec
3dd8 b002 4318:    bcs d.clc
3dda 4319: ;
3dda 38 4320: d.sec: sec
3ddb 60 4321:    rts
3ddc 4322: ;

```

```
3ddc 18 4323: d.clc: clc
3ddd 60 4324: rts
3dde 4325: ;-----
3dde 4326: ;
3dde 20372c 4327: to_pline: jsr clr_printline
3de1 a000 4328: ldy #0
3de3 20153e 4329: d.toplp: jsr and7f
3de6 993e63 4330: sta printline,y
3de9 c8 4331: iny
3dea c006 4332: cpy #6
3dec d0f5 4333: bne d.toplp
3dee 20dc3e 4334: jsr eqlize
3df1 a4cd 4335: ldy ytemp
3df3 60 4336: rts
3df4 4337: ;-----
3df4 4338: ;
3df4 20fc35 4339: do_pline: jsr byte_to_hex
3df7 a99b 4340: pline_out: lda #cr
3df9 9d3e63 4341: sta printline,x
3dfc 4cee38 4342: jmp write_line
3dff 4343: ; JMP CLR_PRINTLINE
3dff 4344: ;-----
3dff 4345: ;
3dff a95e 4346: setl.index: lda # low label.buffer
3e01 85bc 4347: sta label.ptr
3e03 a967 4348: lda # high label.buffer
3e05 85bd 4349: sta label.ptr+1
3e07 60 4350: rts
3e08 4351: ;-----
3e08 4352: ;
3e08 b1bc 4353: c.ladr: lda (label.ptr),y
3e0a d9ba00 4354: cmp label.adr,y
3e0d 60 4355: rts
3e0e 4356: ;-----
3e0e 4357: ;
3e0e b1b8 4358: andspace: lda (syslabel),y
3e10 297f 4359: and #$7f
3e12 c920 4360: cmp #space
3e14 60 4361: rts
3e15 4362: ;-----
3e15 4363: ;
3e15 b1b8 4364: and7f: lda (syslabel),y
3e17 297f 4365: and #$7f
3e19 60 4366: rts
3e1a 4367: ;-----
3e1a 4368: ;
3e1a 18 4369: bumpsl.6: clc
3e1b a5b8 4370: lda syslabel
3e1d 6906 4371: adc #6
3e1f 85b8 4372: sta syslabel
3e21 a5b9 4373: lda syslabel+1
3e23 6900 4374: adc #0
3e25 85b9 4375: sta syslabel+1
```

```
3e27 60 4376: rts
3e28 4377: ;-----
3e28 4378: ;
3e28 a001 4379: backloop: ldy #1
3e2a 20153e 4380: jsr and7f
3e2d 48 4381: pha
3e2e 8d2863 4382: sta offset
3e31 203e3e 4383: d.bkloop: jsr backup
3e34 ce2863 4384: dec offset
3e37 d0f8 4385: bne d.bkloop
3e39 68 4386: pla
3e3a 8d2863 4387: sta offset
3e3d 60 4388: rts
3e3e 4389: ;-----
3e3e 4390: ;
3e3e 38 4391: backup: sec
3e3f a5b8 4392: lda syslabel
3e41 e906 4393: sbc #6
3e43 85b8 4394: sta syslabel
3e45 a5b9 4395: lda syslabel+1
3e47 e900 4396: sbc #0
3e49 85b9 4397: sta syslabel+1
3e4b 60 4398: rts
3e4c 4399: ;-----
3e4c 4400: ;
3e4c a900 4401: zthis.lbl: lda #0
3e4e 85be 4402: sta this.lbl
3e50 85bf 4403: sta this.lbl+1
3e52 60 4404: rts
3e53 4405: ;-----
3e53 4406: ;
3e53 b1b8 4407: ora80: lda (syslabel),y
3e55 0980 4408: ora #$80
3e57 91b8 4409: sta (syslabel),y
3e59 60 4410: rts
3e5a 4411: ;-----
3e5a 4412: ;
3e5a 4413: get_labelcount:
3e5a 4414: ;
3e5a aded62 4415: lda labels
3e5d 8d1263 4416: sta label.count
3e60 aded62 4417: lda labels+1
3e63 8d1363 4418: sta label.count+1
3e66 60 4419: rts
3e67 4420: ;-----
3e67 4421: ;
3e67 38 4422: delcount: sec
3e68 ad1263 4423: lda label.count
3e6b d003 4424: bne d.1
3e6d ce1363 4425: dec label.count+1
3e70 ce1263 4426: d.1: dec label.count
3e73 60 4427: rts
3e74 4428: ;-----
```

```
3e74      4429: ;
3e74 a900 4430: lbladrh: lda #0
3e76 85bb 4431:      sta label.adr+1
3e78 60   4432:      rts
3e79      4433: ;-----
3e79      4434: ;
3e79 e6be 4435: inths.lbl: inc this.lbl
3e7b d002 4436:      bne d.insk
3e7d e6bf 4437:      inc this.lbl+1
3e7f 60   4438: d.insk: rts
3e80      4439: ;-----
3e80      4440: ;
3e80 18   4441: point: clc
3e81 a95e 4442:      lda # low label.buffer
3e83 65be 4443:      adc this.lbl
3e85 85bc 4444:      sta label.ptr
3e87 a967 4445:      lda # high label.buffer
3e89 65bf 4446:      adc this.lbl+1
3e8b 85bd 4447:      sta label.ptr+1
3e8d a5bc 4448:      lda label.ptr
3e8f 65be 4449:      adc this.lbl
3e91 85bc 4450:      sta label.ptr
3e93 a5bd 4451:      lda label.ptr+1
3e95 65bf 4452:      adc this.lbl+1
3e97 85bd 4453:      sta label.ptr+1
3e99 a001 4454:      ldy #1
3e9b b1bc 4455:      lda (label.ptr),y
3e9d 60   4456:      rts
3e9e      4457: ;
3e9e      4458: ;-----
3e9e      4459: ;
3e9e a000 4460: getfirst: ldy #0
3ea0 b1b8 4461:      lda (syslabel),y
3ea2 60   4462:      rts
3ea3      4463: ;
3ea3 a001 4464: getsecond: ldy #1
3ea5 b1b8 4465:      lda (syslabel),y
3ea7 60   4466:      rts
3ea8      4467: ;
3ea8      4468: ;-----
3ea8      4469: ;
3ea8      4470: cheknum:
3ea8      4471: ;
3ea8 c931 4472:      cmp #'1'
3eaa 9005 4473:      bcc d.badnum
3eac c93a 4474:      cmp #':'
3eae b001 4475:      bcs d.badnum
3eb0 24   4476:      dc.b $24
3eb1 38   4477: d.badnum: sec
3eb2 60   4478:      rts
3eb3      4479: ;-----
3eb3      4480: ;
3eb3 18   4481: addtosptr: clc
```

```

3eb4 65c2 4482:      adc sptr
3eb6 85c2 4483:      sta sptr
3eb8 a5c3 4484:      lda sptr+1
3eba 6900 4485:      adc #0
3ebc 85c3 4486:      sta sptr+1
3ebe 60   4487:      rts
3ebf      4488: ;-----
3ebf      4489: ;
3ebf e6c2 4490: inw.sptr: inc sptr
3ec1 d002 4491:      bne d.A
3ec3 e6c3 4492:      inc sptr+1
3ec5 60   4493: d.A:   rts
3ec6      4494: ;-----
3ec6      4495: ;
3ec6 e6c6 4496: inw.pcountr: inc pc
3ec8 d0fb 4497:      bne d.A
3eca e6c7 4498:      inc pc+1
3ecc 60   4499:      rts
3ecd      4500: ;-----
3ecd      4501: ;
3ecd      4502: set_extender:
3ecd      4503: ;
3ecd ace962 4504:      ldy outdot
3ed0 a930 4505:      lda #'0'
3ed2 99ae63 4506:      sta outfn,y
3ed5 99af63 4507:      sta outfn+1,y
3ed8 99b063 4508:      sta outfn+2,y
3edb 60   4509:      rts
3edc      4510: ;-----
3edc      4511: ;
3edc      4512: eqlize:
3edc      4513: ;
3edc a207 4514:      ldx #7
3ede a000 4515:      ldy #0
3ee0 b97162 4516: lizloop: lda eq_m,y
3ee3 c99b 4517:      cmp #cr
3ee5 f007 4518:      beq d.lizdun
3ee7 9d3e63 4519:      sta printline,x
3eea e8   4520:      inx
3eeb c8   4521:      iny
3eec d0f2 4522:      bne lizloop
3eee      4523: ;
3eee a924 4524: d.lizdun: lda #'$'
3ef0 9d3e63 4525:      sta printline,x
3ef3 e8   4526:      inx
3ef4 60   4527:      rts
3ef5      4528: ;-----
3ef5      4529: ;
3ef5 a922 4530: light: lda #$22
3ef7 8d2f02 4531:      sta sdmctl
3efa 8dd762 4532:      sta sdm
3efd 60   4533:      rts
3efe      4534: ;-----

```

```
3efe      4535: ;
3efe      4536: scr_on:
3efe      4537: ;
3efe ad2f02 4538:      lda sdmctl
3f01 8dd762 4539:      sta sdm
3f04 a922  4540:      lda #$22
3f06 8d2f02 4541:      sta sdmctl
3f09 60    4542:      rts
3f0a      4543: ;
3f0a      4544: ;-----
3f0a      4545: ;
3f0a      4546: restore_scr:
3f0a      4547: ;
3f0a add762 4548:      lda sdm
3f0d 8d2f02 4549:      sta sdmctl
3f10 60    4550:      rts
3f11      4551: ;
3f11      4552: ;-----
3f11      4553: ;
3f11      4554: newscreen:
3f11      4555: ;
3f11 205437 4556:      jsr cls
3f14 a906  4557:      lda #6
3f16 8554  4558:      sta rowcrs
3f18 60    4559:      rts
3f19      4560: ;-----
3f19      4561: ;
3f19 ad8f63 4562: chk_in: lda infn+1
3f1c cdaf63 4563:      cmp outfn+1
3f1f d008  4564:      bne d.right
3f21 adef62 4565:      lda which.bytek
3f24 f003  4566:      beq d.right
3f26 4ca940 4567:      jmp indisk_prmpt
3f29 60    4568: d.right: rts
3f2a      4569: ;
3f2a ad8f63 4570: chk_out: lda infn+1
3f2d cdaf63 4571:      cmp outfn+1
3f30 d008  4572:      bne d.dopn.
3f32 adef62 4573:      lda which.bytek
3f35 d003  4574:      bne d.dopn.
3f37 201141 4575:      jsr outdisk_prmpt
3f3a ad6003 4576: d.dopn.: lda $0360
3f3d c9ff  4577:      cmp #em
3f3f d0e8  4578:      bne d.right
3f41 4c333a 4579:      jmp opn_disk
3f44      4580: ;-----
3f44      4581: ;
3f44 9d4203 4582: sixbytes: sta iccom,x
3f47 a906  4583:      lda #6
3f49 9d4803 4584:      sta icbll,x
3f4c a900  4585:      lda #0
3f4e 9d4903 4586:      sta icblh,x
3f51 4c56e4 4587:      jmp ciov
```



```
3f54      4588: ;-----
3f54      4589: ;
3f54 20a039 4590: blokclose: jsr putblock
3f57 4c5a3f 4591:      jmp closedisk
3f5a      4592: ;-----
3f5a      4593: ;
3f5a      4594: closedisk:
3f5a      4595: ; close 2
3f5a a220 4596:      ldx #$20
3f5c 20823f 4597:      jsr close.r
3f5f      4598:
3f5f a900 4599:      lda #0
3f61 8ddd62 4600:      sta dskopnflg
3f64 60 4601:      rts
3f65      4602: ;-----
3f65      4603: ;
3f65 a90b 4604: bput:  lda #bput.
3f67 9d4203 4605:      sta iccom,x
3f6a 4c56e4 4606:      jmp ciov
3f6d      4607: ;-----
3f6d      4608: ;
3f6d a908 4609: ciopen: lda #xwrite
3f6f 9d4a03 4610:      sta icax1,x
3f72 a903 4611: opnly: lda #open.
3f74 9d4203 4612:      sta iccom,x
3f77 4c56e4 4613:      jmp ciov
3f7a      4614: ;-----
3f7a      4615: ;
3f7a a907 4616: bget:  lda #bget.
3f7c 9d4203 4617:      sta iccom,x
3f7f 4c56e4 4618:      jmp ciov
3f82      4619: ;-----
3f82      4620: ;
3f82      4621: close.r:
3f82      4622: ;
3f82 a90c 4623:      lda #close.
3f84 9d4203 4624:      sta iccom,x
3f87 4c56e4 4625:      jmp ciov
3f8a      4626: ;
3f8a      4627: ;-----
3f8a      4628: ;CLOSEALL closes channels
3f8a      4629: ;one through seven.
3f8a      4630: ;-----
3f8a      4631: ;
3f8a      4632: closeall:
3f8a      4633: ;
3f8a a270 4634:      ldx #$70
3f8c 20823f 4635: d.closelp: jsr close.r
3f8f 8a 4636:      txa
3f90 e910 4637:      sbc #$10
3f92 aa 4638:      tax
3f93 d0f7 4639:      bne d.closelp
3f95 60 4640:      rts
```

```
3f96      4641: ;-----
3f96      4642: ;
3f96 a2ff  4643: clrstack: ldx #$ff
3f98 9a    4644:      txs
3f99 4c8421 4645:      jmp start
3f9c      4646: ;-----
3f9c      4647: ;
3f9c      4648: ;      .PAGE "Prompt Subroutines"
3f9c      4649: ;      .include talk
3f9c      4650: ;      ... TALK ...
3f9c      4651: ;
3f9c      4652: outfn_in:
3f9c      4653: ;
3f9c 208037 4654:      jsr printsi
3f9f 9b9b7f 4655:      dc.b cr,cr,tab,space
3fa3 20456e 4656:      dc.b " Enter "
3faa cfd5d4 4657:      dc.b <+128>,"OUTPUT"
3fb0 204669 4658:      dc.b " Filename",CR,TAB
3fbb 202057 4659:      dc.b " WITH NO EXTENDER!",EM
3fcf 208037 4660: d.get_it: jsr printsi
3fd2 9b9b7f 4661:      dc.b cr,cr,tab,em
3fd6      4662: ;
3fd6 a920  4663:      lda #$20
3fd8 204f38 4664:      jsr getline
3fdb c91b  4665:      cmp #esc
3fdd d003  4666:      bne d.good_out
3fdf 4c963f 4667:      jmp clrstack
3fe2      4668: ;
3fe2 ad6663 4669: d.good_out: lda inline
3fe5 c99b  4670:      cmp #cr
3fe7 d002  4671:      bne d.fixout
3fe9 38    4672:      sec
3fea 60    4673:      rts
3feb      4674: ;
3feb a9ae  4675: d.fixout: lda # low outfn ;Make sure
3fed a063  4676:      ldy # high outfn ;we have a
3fef 20ba3b 4677:      jsr normalize ;legal fname.
3ff2 b0a8  4678:      bcs outfn_in
3ff4 ad6863 4679:      lda inline+2
3ff7 c93a  4680:      cmp #'.'
3ff9 f006  4681:      beq d.ofnk
3ffb adcc61 4682:      lda d2
3ffe 8daf63 4683:      sta outfn+1
4001      4684: ;
4001      4685: ;Make sure user didn't enter
4001      4686: ;an extender stripping it if
4001      4687: ;he did then set extender to
4001      4688: ;'.000'.
4001      4689: ;
4001 84cd  4690: d.ofnk:  sty ytemp
4003 88    4691:      dey
4004 b9ae63 4692: d.dotloop: lda outfn,y
4007 c92e  4693:      cmp #'.'
```

```

4009 f007 4694:      beq d.gotdot
400b 88 4695:      dey
400c d0f6 4696:      bne d.dotloop
400e a92e 4697:      lda #'.'
4010 a4cd 4698:      ldy ytemp
4012      4699:      ;
4012 99ae63 4700: d.gotdot: sta outfn,y
4015 c8 4701:      iny
4016 8ce962 4702:      sty outdot
4019 20cd3e 4703:      jsr set_extender
401c a99b 4704:      lda #cr
401e 99b163 4705:      sta outfn+3,y
4021 a9ff 4706:      lda #em
4023 99b263 4707:      sta outfn+4,y
4026 18 4708:      clc
4027 60 4709:      rts
4028      4710: ;-----
4028      4711: ;
4028 a901 4712: get_start: lda #1
402a 8552 4713:      sta lmarg
402c 20113f 4714:      jsr newscreen
402f 208037 4715: start_adr: jsr printsi
4032 9b7f45 4716:      dc.b cr,tab,"Enter Start Add"
4043 726573 4717:      dc.b "ress: ",EM
404a      4718: ;
404a 204d38 4719:      jsr getnum
404d c91b 4720:      cmp #esc
404f d003 4721:      bne d.scont
4051 4c963f 4722:      jmp clrstack
4054      4723: ;
4054 203836 4724: d.scont: jsr digitize
4057 b009 4725:      bcs d.s_err
4059 a5ca 4726:      lda number
405b 85c8 4727:      sta s.org
405d a5cb 4728:      lda number+1
405f 85c9 4729:      sta s.org+1
4061 60 4730:      rts
4062      4731: ;
4062 202744 4732: d.s_err: jsr input_err
4065 4c2f40 4733:      jmp start_adr
4068      4734: ;-----
4068      4735: ;
4068 20113f 4736: get_infn: jsr newscreen
406b      4737: ;      position 7,6
406b a907 4738:      lda #7
406d 8555 4739:      sta colcrs
406f a906 4740:      lda #6
4071 8554 4741:      sta rowcrs
4073      4742:
4073      4743:
4073 208037 4744:      jsr printsi
4076 456e74 4745:      dc.b "Enter name of "
4084 c9ced0 4746:      dc.b <+128>,"INPUT"

```

```

4089 206669 4747:      dc.b " file.",CR,CR,TAB,EM
4093      4748: ;
4093 a920 4749:      lda #$20 ;Allow 32 chars
4095 204f38 4750:      jsr getline ;in filename.
4098 c91b 4751:      cmp #esc
409a d003 4752:      bne d.fixinfn
409c 4c963f 4753:      jmp clrstack
409f      4754: ;
409f a98e 4755: d.fixinfn: lda # low infn ;Go make it
40a1 a063 4756:      ldy # high infn ;legal if it
40a3 20ba3b 4757:      jsr normalize ;isn't already.
40a6 b0c0 4758:      bcs get_infn
40a8 60 4759:      rts ;Move on.
40a9      4760: ;-----
40a9      4761: ;INDISK_PRMPPT prompts user to
40a9      4762: ;to insert input disk.
40a9      4763: ;-----
40a9      4764: ;
40a9      4765: indisk_prmpt:
40a9      4766: ;
40a9 ad8e63 4767:      lda infn
40ac c944 4768:      cmp #'D'
40ae f001 4769:      beq d.prompt_em
40b0 60 4770:      rts
40b1      4771: ;
40b1 205837 4772: d.prompt_em: jsr :3crs
40b4 ad8f63 4773:      lda infn+1
40b7 8dd940 4774:      sta id1
40ba eef002 4775:      inc crsinh
40bd 208037 4776:      jsr printsi
40c0 7f5075 4777:      dc.b tab,"Put Input D"
40cc 69736b 4778:      dc.b "isk in Drive "
40d9 312c9b 4779: id1:   dc.b "1,",CR,EM
40dd ade562 4780:      lda both.flag
40e0 d02c 4781:      bne d.ipxit
40e2 208037 4782:      jsr printsi
40e5 7f5468 4783:      dc.b tab,"Then Pre"
40ee 737320 4784:      dc.b "ss RETURN...",EM
40fb      4785: ;
40fb 20fe3e 4786:      jsr scr_on
40fe 203738 4787:      jsr checkkey
4101 9003 4788:      bcc d.goahead
4103 4c8421 4789:      jmp start
4106      4790: ;
4106 205837 4791: d.goahead: jsr :3crs
4109 a900 4792:      lda #0
410b 8def62 4793:      sta which.bytek
410e 4c0a3f 4794: d.ipxit: jmp restore_scr
4111      4795: ;
4111      4796: ;-----
4111      4797: ;OUTDISK_PRMPPT prompts the user
4111      4798: ;to insert the output disk.
4111      4799: ;-----

```

```

4111      4800: ;
4111      4801: outdisk_prmpt:
4111      4802: ;
4111 adae63 4803:      lda outfn
4114 c944  4804:      cmp #'D'
4116 f001  4805:      beq d.prompt_er
4118 60    4806:      rts
4119      4807: ;
4119 adaf63 4808: d.prompt_er: lda outfn+1
411c 8d4541 4809:      sta od1
411f ade562 4810:      lda both.flag
4122 d003  4811:      bne d.noclr
4124 205b37 4812:      jsr :2crs
4127 208037 4813: d.noclr: jsr printsi
412a 9b7f50 4814:      dc.b cr,tab,"Put Output "
4137 446973 4815:      dc.b "Disk in Drive "
4145 312c9b 4816: od1:   dc.b "l","CR,CR
4149 7f5468 4817:      dc.b tab,"Then Press "
4155 524554 4818:      dc.b "RETURN... ",EM
415f      4819: ;
415f 8ee562 4820:      stx both.flag
4162 20fe3e 4821:      jsr scr_on
4165 203738 4822:      jsr checkkey
4168 9003  4823:      bcc d.gotcha
416a 4c8421 4824:      jmp start
416d      4825: ;
416d 205837 4826: d.gotcha: jsr :3crs
4170 eeef62 4827:      inc which.bytek
4173 4c0a3f 4828:      jmp restore_scr
4176      4829: ;-----
4176      4830: ;
4176      4831: not_binary:
4176      4832: ;
4176 20113f 4833:      jsr newscreen
4179      4834: ;      position 3,10
4179 a903  4835:      lda #3
417b 8555  4836:      sta colcrs
417d a90a  4837:      lda #10
417f 8554  4838:      sta rowcrs
4181      4839: ;
4181 208037 4840:      jsr printsi
4184 a0d4e8 4841:      dc.b <+128>," This is not"
4190 a0e1a0 4842:      dc.b <+128>," a binary lo"
419c e1e4a0 4843:      dc.b <+128>,"ad file. "
41a5 9b9bff 4844:      dc.b cr,cr,em
41a8      4845: ;
41a8 eee862 4846:      inc binflag
41ab 4c2f40 4847:      jmp start_adr
41ae      4848: ;-----
41ae      4849: ;
41ae      4850: newdisk:
41ae      4851: ;
41ae 20fe3e 4852:      jsr scr_on

```

```

41b1 205f43 4853: d.inform: jsr shoerr
41b4 205a3f 4854:      jsr closedisk
41b7 a907 4855:      lda #7
41b9 8552 4856:      sta lmarg
41bb 8555 4857:      sta colcrs
41bd adaf63 4858:      lda outfn+1
41c0 0980 4859:      ora #$80
41c2 8df841 4860:      sta d.nd
41c5 208037 4861:      jsr printsi
41c8 9b9b 4862:      dc.b cr,cr
41ca a0cff5 4863:      dc.b <+128>," Output Disk"
41d6 a0e9f3 4864:      dc.b <+128>," is Full. Pu"
41e2 f4a01b 4865:      dc.b <+128>,"t ",ESC
41e5 a0cee5 4866:      dc.b <+128>," New Disk in"
41f1 a0c4f2 4867:      dc.b <+128>," Drive "
41f8 b1a0d4 4868: d.nd:  dc.b <+128>,"1 Then ",ESC
4200 a0d0f2 4869:      dc.b <+128>," Press RETUR"
420c cea0ef 4870:      dc.b <+128>,"N or Drive "
4218 a0a01b 4871:      dc.b <+128>," ",ESC
421b a0cef5 4872:      dc.b <+128>," Number to S"
4227 f7e9f4 4873:      dc.b <+128>,"witch Drives"
4233 aea0 4874:      dc.b <+128>,". "
4235 1bff 4875:      dc.b ESC,EM
4237      4876: ;
4237 20e637 4877: d.gndrv: jsr getkey
423a c91b 4878:      cmp #esc
423c d003 4879:      bne d.ndcr.
423e 4c8421 4880:      jmp start
4241      4881: ;
4241 c99b 4882: d.ndcr.: cmp #cr
4243 f00b 4883:      beq d.opnew
4245 c931 4884:      cmp #'l'
4247 90ee 4885:      bcc d.gndrv
4249 c93a 4886:      cmp #':'
424b b0ea 4887:      bcs d.gndrv
424d 8daf63 4888:      sta outfn+1
4250 20583d 4889: d.opnew: jsr nxt_outfn
4253 20333a 4890:      jsr opn_disk
4256 1003 4891:      bpl d.newok
4258 4cb141 4892:      jmp d.inform
425b      4893: ;
425b a901 4894: d.newok: lda #1
425d 8552 4895:      sta lmarg
425f 205b37 4896:      jsr :2crs
4262 4c0a3f 4897:      jmp restore_scr
4265      4898: ;-----
4265      4899: ;
4265 200743 4900: bydone: jsr setwhat
4268      4901: ;      col 10
4268 a90a 4902:      lda #10
426a 8555 4903:      sta colcrs
426c      4904:
426c 208037 4905:      jsr printsi

```

```

426f a0c4e1 4906:      dc.b <+128>," Data File C"
427b efedf0 4907:      dc.b <+128>,"omplete. "
4284 9b9bff 4908:      dc.b cr,cr,em
4287 4cae42 4909:      jmp disdun1
428a      4910: ;
428a 200743 4911: nowhat.: jsr setwhat
428d 208037 4912:      jsr printsi
4290 a0aaaa 4913:      dc.b <+128>," *** Disassembly"
42a0 a0a0c4 4914:      dc.b <+128>," Done *** "
42ab 9b9bff 4915:      dc.b cr,cr,em
42ae 208037 4916: disdun1: jsr printsi
42b1 a0a0d0 4917:      dc.b <+128>," Press RETURN t"
42c1 efa0c3 4918:      dc.b <+128>,"o Continue, "
42ce 9b      4919:      dc.b cr
42cf a0a0ef 4920:      dc.b <+128>," or ESCAPE to E"
42df f8e9f4 4921:      dc.b <+128>,"xit to DOS  "
42ec 9b9bff 4922:      dc.b cr,cr,em
42ef 20f53e 4923:      jsr light
42f2 a901    4924:      lda #1
42f4 8552    4925:      sta lmarg
42f6      4926: ;
42f6 20dc37 4927: d.adunlp: jsr getcap
42f9 c99b    4928:      cmp #cr
42fb d003    4929:      bne d.escape.
42fd 4c8421 4930:      jmp start
4300      4931: ;
4300 c91b    4932: d.escape.: cmp #esc
4302 d0f2    4933:      bne d.adunlp
4304 4ce824 4934:      jmp exit
4307      4935: ;-----
4307      4936: ;
4307 205837 4937: setwhat: jsr :3crs
430a a007    4938:      ldy #7
430c 8455    4939:      sty colcrs
430e 88      4940:      dey
430f 8452    4941:      sty lmarg
4311 60      4942:      rts
4312      4943: ;-----
4312      4944: lblprmt:
4312      4945: ; position 8,21
4312 a908    4946:      lda #8
4314 8555    4947:      sta colcrs
4316 a915    4948:      lda #21
4318 8554    4949:      sta rowcrs
431a      4950:
431a 208037 4951:      jsr printsi
431d d0f5f4 4952:      dc.b <+128>,"Put Disk With "
432b ccc1c2 4953: lname:  dc.b <+128>,"LABEL.XLE in"
4337 ff      4954:      dc.b EM
4338      4955: ; position 8,22
4338 a908    4956:      lda #8
433a 8555    4957:      sta colcrs
433c a916    4958:      lda #22

```

```

433e 8554 4959:      sta rowcrs
4340      4960:
4340 208037 4961:      jsr printsi
4343 c4f2e9 4962:      dc.b <+128>,"Drive "
4349 b1a0d4 4963: 1x:      dc.b <+128>,"1 Then Press"
4355 a0d2c5 4964:      dc.b <+128>," RETURN "
435d ff 4965:      dc.b EM
435e      4966: ;
435e 60 4967:      rts
435f      4968: ;-----
435f      4969:
435f      4970: ; .PAGE "Error Routines"
435f      4971: ; .include errs
435f      4972: ; ... ERRS ...
435f      4973: ;
435f      4974: ;IO_ERR prints an error message
435f      4975: ;including the I/O error number
435f      4976: ;in both hex and decimal.
435f      4977: ;At the prompt, ESCAPE exits
435f      4978: ;to DOS and RETURN re-runs the
435f      4979: ;program.
435f      4980: ;-----
435f      4981: ;
435f 98 4982: shoerr: tya
4360 48 4983:      pha
4361 48 4984:      pha
4362 205b37 4985:      jsr :2crs
4365 a920 4986:      lda #space
4367 8dc743 4987:      sta d.e-1
436a 8dc643 4988:      sta d.e-2
436d 68 4989:      pla
436e 203137 4990:      jsr decimal
4371 a0ff 4991:      ldy #$ff
4373 c8 4992: d.cpydec: iny
4374 b1f3 4993:      lda (inbuff),y
4376 3007 4994:      bmi d.deccpd
4378 0980 4995:      ora #$80
437a 996663 4996:      sta inline,y
437d 30f4 4997:      bmi d.cpydec
437f 996663 4998: d.deccpd: sta inline,y
4382 c8 4999:      iny
4383 98 5000:      tya
4384 aa 5001:      tax
4385 a929 5002:      lda #')'
4387 9dc343 5003:      sta d.decerr,x
438a ca 5004:      dex
438b bd6663 5005: d.eloop: lda inline,x
438e 9dc343 5006:      sta d.decerr,x
4391 ca 5007:      dex
4392 10f7 5008:      bpl d.eloop
4394 e8 5009:      inx
4395 68 5010:      pla
4396 20fc35 5011:      jsr byte_to_hex

```



```
4399 ca 5012: dex
439a bd3e63 5013: d.hloop: lda printline,x
439d 9dbf43 5014: sta d.hexerr,x
43a0 ca 5015: dex
43a1 10f7 5016: bpl d.hloop
43a3 a211 5017: ldx #d.e-d.errline
43a5 bdb743 5018: d.elloop: lda d.errline,x
43a8 0980 5019: ora #$80
43aa 9db743 5020: sta d.errline,x
43ad ca 5021: dex
43ae 10f5 5022: bpl d.elloop
43b0 a90b 5023: lda #11
43b2 8555 5024: sta colcrs
43b4 208037 5025: jsr printsi
43b7 5026: ;
43b7 204552 5027: d.errline: dc.b " ERROR $"
43bf 202020 5028: d.hexerr: dc.b " ("
43c3 202020 5029: d.decerr: dc.b " "
43c8 ff 5030: d.e: dc.b em
43c9 5031: ;
43c9 60 5032: rts
43ca 5033: ;-----
43ca 5034: ;
43ca 205f43 5035: io_err: jsr shoerr
43cd a901 5036: lda #1
43cf 8552 5037: sta lmarg
43d1 208037 5038: jsr printsi
43d4 9b9b7f 5039: dc.b cr,cr,tab,space
43d8 507265 5040: dc.b "Press RETURN"
43e4 20746f 5041: dc.b " to Restart,"
43f0 9b9b7f 5042: dc.b cr,cr,tab,space
43f4 6f7220 5043: dc.b "or ESCAPE to"
4400 204578 5044: dc.b " Exit to DOS.",EM
440e 5045: ;
440e 20fe3e 5046: jsr scr_on
4411 203738 5047: d.kylp: jsr checkkey
4414 9003 5048: bcc d.rt
4416 4ce824 5049: jmp exit
4419 5050: ;
4419 4c8421 5051: d.rt: jmp start
441c 5052: ;
441c 5053: ;-----
441c 5054: ;INPUT_ERR tells the user
441c 5055: ;that he screwed up and to
441c 5056: ;try again.
441c 5057: ;-----
441c 5058: ;
441c 208037 5059: in_err: jsr printsi
441f 1c9c9c 5060: dc.b $1c,$9c,$9c,cr,em
4424 4c2a44 5061: jmp d.setcol
4427 5062: ;
4427 205b37 5063: input_err: jsr :2crs
442a a905 5064: d.setcol: lda #5
```

```

442c 8555 5065:      sta colcrs
442e 208037 5066:      jsr printsi
4431 c9ced0 5067:      dc.b <+128>,"INPUT ERROR."
443d aeaea0 5068:      dc.b <+128>,".. PLEASE TR"
4449 d9a0c1 5069:      dc.b <+128>,"Y AGAIN."
4451 9b9bff 5070:      dc.b cr,cr,em
4454 60 5071:      rts
4455 5072: ;-----
4455 5073:
4455 5074:
4455 5075:
4455 5076: ; ... OPTN ...
4455 5077: ;
4455 5078: ;Options menu to allow user to
4455 5079: ;customize the program. The
4455 5080: ;users choice will be deter-
4455 5081: ;mined by the row the cursor
4455 5082: ;is on when a key is pressed.
4455 5083: ;
4455 5084: options:
4455 5085: ;
4455 20fb46 5086:      jsr opmenu
4458 5087: ;
4458 5088: ;Wait for either consol button
4458 5089: ;or keypress.
4458 5090: ;
4458 5091: d.whatkey:
4458 5092: ; col 24
4458 a918 5093:      lda #24
445a 8555 5094:      sta colcrs
445c 5095:
445c a9ff 5096:      lda #$ff
445e 8dfc02 5097:      sta ch
4461 a000 5098:      ldy #0
4463 844d 5099: d.waitkey: sty 77
4465 ad1fd0 5100:      lda consol
4468 c905 5101:      cmp #5 ;SELECT?
446a f019 5102:      beq nextrow
446c c903 5103:      cmp #3 ;OPTION?
446e f02e 5104:      beq lastrow
4470 c906 5105:      cmp #6
4472 f00e 5106:      beq d.jtm
4474 adfc02 5107:      lda ch
4477 c9ff 5108:      cmp #$ff
4479 f0e8 5109:      beq d.waitkey
447b 20eb37 5110:      jsr gk1
447e c91b 5111:      cmp #esc
4480 d03d 5112:      bne d.what.
4482 4c7a46 5113: d.jtm: jmp to_menu
4485 5114: ;
4485 5115: ;-----
4485 5116: ;
4485 a001 5117: nextrow: ldy #1 ;Move

```

```
4487 84cd 5118:    sty ytemp ;cursor down one
4489 209b46 5119:    jsr normvert ;row
448c a554 5120:    lda rowcrs
448e c914 5121:    cmp #20
4490 d004 5122:    bne d.nxtok
4492 a902 5123:    lda #2
4494 8554 5124:    sta rowcrs
4496 e654 5125: d.nxtok: inc rowcrs
4498 208746 5126: d.ivj:  jsr invert
449b 4cb244 5127:    jmp d.wbit
449e      5128: ;
449e 84cd 5129: lastrow: sty ytemp ;Move
44a0 209b46 5130:    jsr normvert ;cursor up one
44a3 a554 5131:    lda rowcrs ;row.
44a5 c903 5132:    cmp #3
44a7 d004 5133:    bne d.lstok
44a9 a915 5134:    lda #21
44ab 8554 5135:    sta rowcrs
44ad c654 5136: d.lstok: dec rowcrs
44af 208746 5137:    jsr invert
44b2      5138: ;
44b2      5139: ;Wait a bit.
44b2      5140: ;
44b2 a900 5141: d.wbit:  lda #0
44b4 8514 5142:    sta 20
44b6 a907 5143:    lda #7
44b8 c514 5144: d.wbtlp: cmp 20
44ba d0fc 5145:    bne d.wbtlp
44bc 4c5844 5146: d.whj:  jmp d.whatkey
44bf      5147: ;-----
44bf      5148: ;
44bf c99b 5149: d.what.: cmp #cr
44c1 f0c2 5150:    beq nextrow
44c3 48 5151:    pha      ;Figure out
44c4 a554 5152:    lda rowcrs ;what routine
44c6 0a 5153:    asl      ;is being called
44c7 a8 5154:    tay      ;by which line cursor
44c8 b93a61 5155:    lda optn_tbl,y ;is on and load
44cb 8dd644 5156:    sta opjmp+1 ;called routines
44ce b93b61 5157:    lda optn_tbl+1,y ;address into
44d1 8dd744 5158:    sta opjmp+2 ;OPJMP's operand
44d4 68 5159:    pla      ;field.
44d5 4cffff 5160: opjmp:  jmp dummy
44d8      5161: ;-----
44d8      5162: ;
44d8 20a83e 5163: indrv:  jsr cheknum ;Change
44db b0df 5164:    bcs d.whj  ;Input drv #.
44dd 8dca61 5165:    sta dn+1
44e0 4c8649 5166:    jmp shodn
44e3      5167: ;
44e3 20a83e 5168: outdrv: jsr cheknum ;Change
44e6 b0d4 5169:    bcs d.whj  ;Output drv #.
44e8 8dcc61 5170:    sta d2
```

```
44eb 4c8649 5171:    jmp shodn
44ee      5172:    ;
44ee 20a83e 5173: asmdrv: jsr cheknum ;Change
44f1 b0c9 5174:    bcs d.whj  ;Asm drv #.
44f3 8dcd61 5175:    sta da
44f6 4c8649 5176:    jmp shodn
44f9      5177:    ;
44f9 c920 5178: flipscr: cmp #space ;Flip Screen
44fb d067 5179:    bne d.whatjmp
44fd a918 5180:    lda #24
44ff 8555 5181:    sta colcrs
4501 ad9561 5182:    lda scr
4504 4922 5183:    eor #$22  ;on/off.
4506 8d9561 5184:    sta scr
4509 20e248 5185:    jsr on_off
450c 4c8544 5186:    jmp nextrow
450f      5187:    ;
450f c920 5188: flipprt: cmp #space ;Flip Printer
4511 d051 5189:    bne d.whatjmp
4513 a918 5190:    lda #24
4515 8555 5191:    sta colcrs
4517 ad9661 5192:    lda prt
451a 49ff 5193:    eor #$ff  ;on/off.
451c 8d9661 5194:    sta prt
451f 20e248 5195:    jsr on_off
4522 4c8544 5196:    jmp nextrow
4525      5197:    ;
4525 c920 5198: fliplino: cmp #space ;Line
4527 d03b 5199:    bne d.whatjmp ;numbers
4529 ad9b61 5200:    lda lino.flag ;on/off
452c 4901 5201:    eor #1
452e 8d9b61 5202:    sta lino.flag
4531 f00c 5203:    beq d.lineoff
4533 a938 5204:    lda # low lnum
4535 8d9761 5205:    sta line.start
4538 a963 5206:    lda # high lnum
453a 8d9861 5207:    sta line.start+1
453d d00a 5208:    bne d.slno
453f      5209:    ;
453f a93e 5210: d.lineoff: lda # low printline
4541 8d9761 5211:    sta line.start
4544 a963 5212:    lda # high printline
4546 8d9861 5213:    sta line.start+1
4549 ad9b61 5214: d.slno:  lda lino.flag ;Show lino
454c 20f448 5215:    jsr yes_no ;status.
454f 4c8544 5216:    jmp nextrow
4552      5217:    ;
4552 c920 5218: asmfile: cmp #space ;Flip
4554 d00e 5219:    bne d.whatjmp ;ASM file
4556 ad9c61 5220:    lda asmflg ;flag on/off.
4559 4901 5221:    eor #1
455b 8d9c61 5222:    sta asmflg
455e 20f448 5223:    jsr yes_no
```

```
4561 4c8544 5224:      jmp nextrow
4564      5225: ;
4564 4c5844 5226: d.whatjmp: jmp d.whatkey
4567      5227: ;
4567      5228: ;
4567 20a83e 5229: linebyts: jsr cheknum ;Set bytes
456a b0f8 5230:      bcs d.whatjmp ;per line.
456c 290f 5231:      and #$0f
456e 8d9a61 5232:      sta bytes
4571 200649 5233:      jsr shobytes
4574 4c8246 5234:      jmp decnext
4577      5235: ;
4577 20364a 5236: newfsize: jsr getdata ;Change
457a 203836 5237:      jsr digitize ;output file
457d b00a 5238:      bcs d.oldfs ;size.
457f a5ca 5239:      lda number
4581 8d9d61 5240:      sta max.fsize
4584 a5cb 5241:      lda number+1
4586 8d9e61 5242:      sta max.fsize+1
4589 201149 5243: d.oldfs: jsr shofsize
458c 4c8246 5244:      jmp decnext
458f      5245: ;
458f      5246: ;
458f 20364a 5247: zflip: jsr getdata ;Change
4592 d006 5248:      bne d.mvzz ;Zero-page
4594 202449 5249: d.shz: jsr shozmark ;label marker.
4597 4c8246 5250:      jmp decnext
459a      5251: ;
459a a002 5252: d.mvzz: ldy #2
459c b96663 5253: d.n9lp: lda inline,y
459f 99a261 5254:      sta zmarker,y
45a2 88 5255:      dey
45a3 10f7 5256:      bpl d.n9lp
45a5 30ed 5257:      bmi d.shz
45a7      5258: ;
45a7      5259: ;
45a7 20364a 5260: mflip: jsr getdata ;Change
45aa d006 5261:      bne d.mvmm ;Non zero-page
45ac 203d49 5262: d.shm: jsr shomark ;label marker.
45af 4c8246 5263:      jmp decnext
45b2      5264: ;
45b2 a002 5265: d.mvmm: ldy #2
45b4 b96663 5266: d.n10lp: lda inline,y
45b7 999f61 5267:      sta marker,y
45ba 88 5268:      dey
45bb 10f7 5269:      bpl d.n10lp
45bd 30ed 5270:      bmi d.shm
45bf      5271: ;
45bf 20364a 5272: neworg: jsr getdata ;Change
45c2 f01b 5273:      beq d.oldorg ;assembler ORG
45c4 20e946 5274:      jsr addspace ;directive.
45c7 a000 5275:      ldy #0
45c9 b96663 5276: d.mvorg: lda inline,y
```

```
45cc 996462 5277:      sta org_m,y
45cf c99b 5278:      cmp #cr
45d1 f007 5279:      beq d.orgmvd
45d3 c00b 5280:      cpy #11
45d5 b003 5281:      bcs d.orgmvd
45d7 c8 5282:      iny
45d8 d0ef 5283:      bne d.mvorg
45da a99b 5284: d.orgmvd: lda #cr
45dc 996462 5285:      sta org_m,y
45df 204c49 5286: d.oldorg: jsr shoorgm
45e2 4c8246 5287:      jmp decnext
45e5      5288: ;
45e5 20364a 5289: neweq: jsr getdata ;Change
45e8 f01b 5290:      beq d.oldeqm ;EQUAL directive.
45ea 20e946 5291:      jsr addspace
45ed a000 5292:      ldy #0
45ef b96663 5293: d.mveq:  lda inline,y
45f2 997162 5294:      sta eq_m,y
45f5 c99b 5295:      cmp #cr
45f7 f007 5296:      beq d.mveqdn
45f9 c00b 5297:      cpy #11
45fb f003 5298:      beq d.mveqdn
45fd c8 5299:      iny
45fe d0ef 5300:      bne d.mveq
4600 a99b 5301: d.mveqdn: lda #cr
4602 997162 5302:      sta eq_m,y
4605 205249 5303: d.oldeqm: jsr shoeq
4608 4c8246 5304:      jmp decnext
460b      5305: ;
460b 20364a 5306: newbyte: jsr getdata ;Change
460e f01b 5307:      beq d.oldbyt ;BYTE directive.
4610 20e946 5308:      jsr addspace
4613 a000 5309:      ldy #0
4615 b96663 5310: d.mvbyt: lda inline,y
4618 997c62 5311:      sta byte_m,y
461b c99b 5312:      cmp #cr
461d f007 5313:      beq d.bytmvd
461f c00b 5314:      cpy #11
4621 b003 5315:      bcs d.bytmvd
4623 c8 5316:      iny
4624 d0ef 5317:      bne d.mvbyt
4626 a99b 5318: d.bytmvd: lda #cr
4628 997c62 5319:      sta byte_m,y
462b 205849 5320: d.oldbyt: jsr shobytm
462e 4c8246 5321:      jmp decnext
4631      5322: ;
4631      5323: ;
4631 20364a 5324: newlink: jsr getdata ;Change
4634 f018 5325:      beq d.oldlnk ;LINK directive.
4636 a000 5326:      ldy #0
4638 b96663 5327: d.mvlnk: lda inline,y
463b 998762 5328:      sta link_m,y
463e c99b 5329:      cmp #cr
```

```

4640 f007 5330:      beq d.lnkmvd
4642 c00b 5331:      cpy #11
4644 b003 5332:      bcs d.lnkmvd
4646 c8 5333:       iny
4647 d0ef 5334:      bne d.mvlnk
4649 a99b 5335: d.lnkmvd: lda #cr
464b 998762 5336:      sta link_m,y
464e 205e49 5337: d.oldlnk: jsr sholink
4651 4c8246 5338:      jmp decnext
4654      5339: ;
4654 20364a 5340: newend: jsr getdata ;Change
4657 f01b 5341:      beq d.oldend ;END directive
4659 20e946 5342:      jsr addspace
465c a000 5343:      ldy #0
465e b96663 5344: d.mvend: lda inline,y
4661 999262 5345:      sta end_m,y
4664 c99b 5346:      cmp #cr
4666 f007 5347:      beq d.endmvd
4668 c00b 5348:      cpy #11
466a b003 5349:      bcs d.endmvd
466c c8 5350:       iny
466d d0ef 5351:      bne d.mvend
466f a99b 5352: d.endmvd: lda #cr
4671 999262 5353:      sta end_m,y
4674 206449 5354: d.oldend: jsr shoend
4677 4c8246 5355:      jmp decnext
467a      5356: ;
467a a901 5357: to_menu: lda #1
467c 8552 5358:      sta lmarg
467e 60 5359:      rts
467f      5360: ;
467f 4c5844 5361: d.jwhtky: jmp d.whatkey
4682      5362: ;-----
4682      5363: ;
4682 c654 5364: decnext: dec rowcrs
4684 4c8544 5365:      jmp nextrow
4687      5366: ;-----
4687      5367: ;
4687      5368: ;Routine to invert line cursor
4687      5369: ;is on.
4687      5370: ;
4687 20bf46 5371: invert: jsr sety
468a c00f 5372: d.ivlup1: cpy #15
468c f005 5373:      beq d.iv
468e 20af46 5374:      jsr vert
4691 d0f7 5375:      bne d.ivlup1
4693      5376: ;
4693 88 5377: d.iv:  dey
4694 88 5378:      dey
4695 20af46 5379: d.ivlup2: jsr vert
4698 10fb 5380:      bpl d.ivlup2
469a 60 5381:      rts
469b      5382: ;-----

```

```
469b 5383: ;
469b 5384: ;Routine to un-inverse line
469b 5385: ;cursor is on before moving
469b 5386: ;cursor to another line.
469b 5387: ;
469b 20bf46 5388: normvert: jsr sety
469e c00f 5389: d.nvlup1: cpy #15
46a0 f005 5390: beq d.nv
46a2 20b746 5391: jsr trev
46a5 d0f7 5392: bne d.nvlup1
46a7 5393: ;
46a7 88 5394: d.nv: dey
46a8 88 5395: dey
46a9 20b746 5396: d.nvlup2: jsr trev
46ac 10fb 5397: bpl d.nvlup2
46ae 60 5398: rts
46af 5399: ;-----
46af 5400: ;
46af b1c2 5401: vert: lda (sptr),y ;Invert a
46b1 0980 5402: ora #$80 ;character and
46b3 91c2 5403: sta (sptr),y ;decrement
46b5 88 5404: dey ;Y register.
46b6 60 5405: rts
46b7 5406: ;
46b7 b1c2 5407: trev: lda (sptr),y ;Un_invert
46b9 297f 5408: and #$7f ;a character.
46bb 91c2 5409: sta (sptr),y
46bd 88 5410: dey
46be 60 5411: rts
46bf 5412: ;-----
46bf 5413: ;
46bf 5414: ;Routine to load Y reg with
46bf 5415: ;number of characters to
46bf 5416: ;invert or normalize in the
46bf 5417: ;data field of a given line
46bf 5418: ;using COLSIZE table.
46bf 5419: ;
46bf a454 5420: sety: ldy rowcrs
46c1 b96c61 5421: lda colsize,y
46c4 20d146 5422: jsr setrow
46c7 a454 5423: ldy rowcrs
46c9 b96c61 5424: lda colsize,y
46cc 18 5425: clc
46cd 690f 5426: adc #15
46cf a8 5427: tay
46d0 60 5428: rts
46d1 5429: ;
46d1 5430: ;-----
46d1 5431: ;
46d1 5432: ;Point Zero page pointer to
46d1 5433: ;line cursor is on for
46d1 5434: ;inverting/normalizing line.
46d1 5435: ;
```



```

46d1      5436: setrow:
46d1      5437: ;
46d1 a558  5438:      lda savmsc
46d3 85c2  5439:      sta sptr
46d5 a559  5440:      lda savmsc+1
46d7 85c3  5441:      sta sptr+1
46d9 a454  5442:      ldy rowcrs
46db a928  5443: d.pluslp: lda #40
46dd 20b33e 5444:      jsr addtosptr
46e0 88    5445:      dey
46e1 d0f8  5446:      bne d.pluslp
46e3 a908  5447:      lda #8
46e5 20b33e 5448:      jsr addtosptr
46e8 60    5449:      rts
46e9      5450: ;-----
46e9      5451: ;
46e9      5452: ;Add one space to a directive
46e9      5453: ;so we don't butt against it
46e9      5454: ;later.
46e9      5455: ;
46e9 b96563 5456: addspace: lda inline-1,y
46ec c920  5457:      cmp #space
46ee f00a  5458:      beq d.spaced
46f0 a920  5459:      lda #space
46f2 996663 5460:      sta inline,y
46f5 a99b  5461:      lda #cr
46f7 996763 5462:      sta inline+1,y
46fa 60    5463: d.spaced: rts
46fb      5464: ;-----
46fb      5465:
46fb      5466:
46fb      5467:
46fb      5468: ;      .PAGE "Option I/O Routines"
46fb      5469: ;      .include opio
46fb      5470:
46fb      5471: ;      ... OPIO ...
46fb      5472: ;
46fb a901  5473: opmenu: lda #1
46fd 8df002 5474:      sta crsinh
4700 20113f 5475:      jsr newscreen
4703      5476: ;      position 0,0
4703 a900  5477:      lda #0
4705 8555  5478:      sta colcrs
4707 a900  5479:      lda #0
4709 8554  5480:      sta rowcrs
470b      5481:
470b 208037 5482:      jsr printsi
470e      5483: ;
470e      5484: ;      11111111112222222222333333
470e      5485: ;      012345678901234567890123456789012345
470e d3c5cc 5486:      dc.b <+128>,"SELECT = down OPTION = up START =
4736 9b9bff 5487:      dc.b cr,cr,em
4739      5488: ;      col 8

```

```

4739 a908 5489:    lda #8
473b 8555 5490:    sta colcrs
473d      5491:
473d 8552 5492:    sta lmarg
473f 208037 5493:    jsr printsi
4742 496e70 5494:    dc.b "Input Drive ",CR
4752 4f7574 5495:    dc.b "Output Drive ",CR
4762 417373 5496:    dc.b "Assembly Drive ",CR
4772 536372 5497:    dc.b "Screen",CR
4779 507269 5498:    dc.b "Printer",CR
4781 4c696e 5499:    dc.b "Line Numbers? ",CR
4791 274153 5500:    dc.b "'ASM' File? ",CR
47a1 4c6162 5501:    dc.b "Label Type ",CR
47b1 427974 5502:    dc.b "Bytes per Line ",CR
47c1 46696c 5503:    dc.b "File Size ",CR
47d1 5a6572 5504:    dc.b "Zero Page ID ",CR
47e1 4e6f6e 5505:    dc.b "Non-ZP ID ",CR
47f1 4f5247 5506:    dc.b "ORG Directive ",CR
4801 455155 5507:    dc.b "EQU Directive ",CR
4811 425954 5508:    dc.b "BYTE Directive ",CR
4821 4c494e 5509:    dc.b "LINK Directive ",CR
4831 454e44 5510:    dc.b "END Directive ",CR
4841 536176 5511:    dc.b "Save MEGADIS",EM
484e      5512: ;
484e a903 5513:    lda #3 ;Show all the
4850 8554 5514:    sta rowcrs ;defaults.
4852      5515: ;
4852 a918 5516:    lda #24
4854 8555 5517:    sta colcrs
4856      5518:
4856 8552 5519:    sta lmarg
4858 adca61 5520:    lda dn+1 ;Input drv #.
485b 8d6d48 5521:    sta dnum
485e adcc61 5522:    lda d2 ;Output drv #.
4861 8d6f48 5523:    sta dnum+2
4864 adcd61 5524:    lda da ;Asm drv #.
4867 8d7148 5525:    sta dnum+4
486a 208037 5526:    jsr printsi
486d 319b31 5527: dnum: dc.b "1",CR,"1",CR
4871 319bff 5528:    dc.b "1",CR,EM
4874 ad9561 5529:    lda scr ;Screen.
4877 20e248 5530:    jsr on_off
487a 205e37 5531:    jsr putcr
487d ad9661 5532:    lda prt
4880 20e248 5533:    jsr on_off ;Printer.
4883 205e37 5534:    jsr putcr
4886 ad9b61 5535:    lda lino.flag ;Line numbers?
4889 20f448 5536:    jsr yes_no
488c 205e37 5537:    jsr putcr
488f ad9c61 5538:    lda asmflg ;ASM file?
4892 20f448 5539:    jsr yes_no
4895 205e37 5540:    jsr putcr
4898 ad9961 5541:    lda xe ;Show ATARI label

```

```

489b f00d 5542:    beq d.800 ;type.
489d 208037 5543:    jsr printsi
48a0 584c2f 5544:    dc.b "XL/XE",CR,EM
48a7 4cb348 5545:    jmp d.srg
48aa 208037 5546: d.800:  jsr printsi
48ad 383030 5547:    dc.b "800 ",CR,EM
48b3 200649 5548: d.srg:  jsr shobytes
48b6 201149 5549:    jsr shofsize
48b9 202449 5550:    jsr shozmark
48bc 203d49 5551:    jsr shomark
48bf 204c49 5552:    jsr shoorgm
48c2 205249 5553:    jsr shoeq
48c5 205849 5554:    jsr shobytm
48c8 205e49 5555:    jsr sholink
48cb 206449 5556:    jsr shoend
48ce      5557: ;
48ce 18 5558:    clc      ;Show cursor on
48cf a558 5559:    lda savmsc ;top line.
48d1 6958 5560:    adc #88
48d3 85c2 5561:    sta sptr
48d5 a559 5562:    lda savmsc+1
48d7 6900 5563:    adc #0
48d9 85c3 5564:    sta sptr+1
48db a903 5565:    lda #3
48dd 8554 5566:    sta rowcrs
48df 4c8746 5567:    jmp invert
48e2      5568: ;-----
48e2      5569: ;
48e2      5570: ;Routines to show various
48e2      5571: ;data and flags.
48e2      5572: ;
48e2 f008 5573: on_off: beq d.off
48e4 208037 5574:    jsr printsi
48e7 4f6e20 5575:    dc.b "On ",EM
48eb 60 5576:    rts
48ec 208037 5577: d.off:  jsr printsi
48ef 4f6666 5578:    dc.b "Off",EM
48f3 60 5579:    rts
48f4      5580: ;
48f4      5581: ;If 0 then print 'No ' else
48f4      5582: ;print 'Yes'
48f4      5583: ;
48f4 f008 5584: yes_no: beq no
48f6      5585: ;
48f6 208037 5586:    jsr printsi
48f9 596573 5587:    dc.b "Yes",EM
48fd 60 5588:    rts
48fe      5589: ;
48fe 208037 5590: no:    jsr printsi
4901 4e6f20 5591:    dc.b "No ",EM
4905 60 5592:    rts
4906      5593: ;-----
4906 ad9a61 5594: shobytes: lda bytes

```

```
4909 0930 5595:    ora #'0'    ;Show bytes
490b 206437 5596:    jsr putchar ;per line.
490e 4c5e37 5597:    jmp putcr
4911      5598:    ;
4911      5599: shofsize:
4911      5600:    ; col 24
4911 a918 5601:    lda #24
4913 8555 5602:    sta colcrs
4915      5603:
4915 ad9e61 5604:    lda max.fsize+1 ;Show output
4918 201c36 5605:    jsr show_hex ;file size
491b ad9d61 5606:    lda max.fsize
491e 202336 5607:    jsr just_sho
4921 4c5e37 5608:    jmp putcr
4924      5609:    ;
4924      5610: shozmark:
4924      5611:    ; col 24
4924 a918 5612:    lda #24
4926 8555 5613:    sta colcrs
4928      5614:
4928 ada261 5615:    lda zmarker ;Show ZP
492b 206437 5616:    jsr putchar ;marker.
492e ada361 5617:    lda zmarker+1
4931 c99b 5618: d.mrcr: cmp #cr
4933 d002 5619:    bne d.mok
4935 a920 5620:    lda #space
4937 206437 5621: d.mok: jsr putchar
493a 4c5e37 5622:    jmp putcr
493d      5623:    ;
493d      5624: shomark:
493d      5625:    ; col 24
493d a918 5626:    lda #24
493f 8555 5627:    sta colcrs
4941      5628:
4941 ad9f61 5629:    lda marker ;Show non-ZP
4944 206437 5630:    jsr putchar ;marker
4947 ada061 5631:    lda marker+1
494a d0e5 5632:    bne d.mrcr
494c      5633:    ;
494c a964 5634: shoorgm: lda # low org_m
494e a062 5635:    ldy # high org_m ;Show ORG
4950 d018 5636:    bne showit ;directive.
4952      5637:    ;
4952 a971 5638: shoeq:  lda # low eq_m
4954 a062 5639:    ldy # high eq_m ;Show EQUATE
4956 d012 5640:    bne showit ;directive.
4958      5641:    ;
4958 a97c 5642: shobytm: lda # low byte_m
495a a062 5643:    ldy # high byte_m ;Show BYTE
495c d00c 5644:    bne showit ;directive.
495e      5645:    ;
495e a987 5646: sholink: lda # low link_m
4960 a062 5647:    ldy # high link_m ;Show LINK
```

```
4962 d006 5648:      bne showit ;directive.
4964      5649: ;
4964 a992 5650: shoend: lda # low end_m
4966 a062 5651:      ldy # high end_m ;Show END
4968 d000 5652:      bne showit ;directive.
496a      5653: ;-----
496a      5654: ;
496a      5655: ;Routine prints line of up to
496a      5656: ;10 chars to screen starting
496a      5657: ;in column 24. Enter with lo
496a      5658: ;adr in A reg and hi adr in Y.
496a      5659: ;
496a a218 5660: showit: ldx #24
496c 8655 5661:      stx colcrs
496e 8d4403 5662:      sta icbal
4971 8c4503 5663:      sty icbah
4974 a90a 5664:      lda #10
4976 8d4803 5665:      sta icbll
4979 a200 5666:      ldx #0
497b 8e4903 5667:      stx icblh
497e a909 5668:      lda #putrec.
4980 8d4203 5669:      sta iccom
4983 4c56e4 5670:      jmp ciov
4986      5671: ;-----
4986      5672: ;
4986      5673: ;Show drive # in inverse.
4986      5674: ;
4986 a218 5675: shodn: ldx #24
4988 8655 5676:      stx colcrs
498a 0980 5677:      ora #$80
498c 206437 5678:      jsr putchar
498f 4c8544 5679:      jmp nextrow
4992      5680: ;-----
4992      5681: ;
4992      5682: ;Load OS label file from disk.
4992      5683: ;
4992 48 5684: load_labels: pha
4993 adca61 5685:      lda dn+1 ;Use and show right
4996 8dd661 5686:      sta l800+1 ;input drive
4999 8de361 5687:      sta lxle+1 ;number
499c 0980 5688:      ora #$80
499e 8d4943 5689:      sta lx
49a1 68 5690:      pla
49a2 c938 5691:      cmp #'8' ;Load 800 labels?
49a4 f003 5692:      beq d.do800 ;yes.
49a6 4cb849 5693:      jmp d.lxl. ;No.
49a9      5694: ;
49a9 a9d5 5695: d.do800: lda # low l800
49ab 85c2 5696:      sta sptr
49ad a961 5697:      lda # high l800
49af 85c3 5698:      sta sptr+1
49b1 a200 5699:      ldx #0
49b3 8ef362 5700:      stx xsave
```

```
49b6 f014 5701:      beq d.opnlbl
49b8      5702: ;
49b8 c958 5703: d.lxl:  cmp #'X'  ;Load XE labels?
49ba f003 5704:      beq d.doxl  ;Yes.
49bc 4c5844 5705:      jmp d.whatkey ;No... oops...
49bf      5706: ;
49bf a9e2 5707: d.doxl:  lda # low lxle
49c1 85c2 5708:      sta sptr
49c3 a961 5709:      lda # high lxle
49c5 85c3 5710:      sta sptr+1
49c7 a201 5711:      ldx #1
49c9 8ef362 5712:      stx xsave
49cc      5713: ;
49cc a000 5714: d.opnlbl: ldy #0
49ce b1c2 5715: d.oplblp: lda (sptr),y
49d0 998e63 5716:      sta infn,y
49d3 c003 5717:      cpy #3
49d5 9005 5718:      bcc d.nxtopl
49d7 0980 5719:      ora #$80
49d9 992843 5720:      sta lname-3,y
49dc c8 5721: d.nxtopl: iny
49dd c00c 5722:      cpy #12
49df d0ed 5723:      bne d.oplblp
49e1 b1c2 5724:      lda (sptr),y
49e3 998e63 5725:      sta infn,y
49e6 201243 5726:      jsr lblprmt
49e9 203738 5727:      jsr checkkey
49ec 9003 5728:      bcc d.keyok
49ee 4c5544 5729:      jmp options
49f1      5730: ;
49f1 20053b 5731: d.keyok: jsr open_in
49f4 adf362 5732: d.xeok:  lda xsave
49f7 8d9961 5733:      sta xe
49fa 208037 5734:      jsr printsi  ;Erase
49fd 1c9c9c 5735:      dc.b $1c,$9c,$9c,em ;disk prompt.
4a01 a210 5736:      ldx #chan1  ;Read 6 header
4a03 a5b0 5737:      lda buffer ;bytes
4a05 9d4403 5738:      sta icbal,x ;which we don't
4a08 a5b1 5739:      lda buffer+1 ;want.
4a0a 9d4503 5740:      sta icbah,x
4a0d a907 5741:      lda #bget.
4a0f 20443f 5742:      jsr sixbytes
4a12 1003 5743:      bpl d.loadmore
4a14 4cca43 5744:      jmp io_err
4a17 a97a 5745: d.loadmore: lda # low page0 ;Read in
4a19 9d4403 5746:      sta icbal,x ;the label file.
4a1c a94c 5747:      lda # high page0
4a1e 9d4503 5748:      sta icbah,x
4a21 a9ff 5749:      lda #$ff
4a23 9d4803 5750:      sta icbll,x
4a26 9d4903 5751:      sta icblh,x
4a29 2056e4 5752:      jsr ciov
4a2c c088 5753:      cpy #$88  ;EOF?
```

```
4a2e f003 5754:      beq d.goodload ;Yup.
4a30 4cca43 5755:      jmp io_err ;ONONO!
4a33 4c5544 5756: d.goodload: jmp options
4a36      5757: ;-----
4a36      5758: ;
4a36      5759: ;Routine to get a line of
4a36      5760: ;input from user. Character
4a36      5761: ;in A reg on entry is used as
4a36      5762: ;first char of said input.
4a36      5763: ;Input is saved in INLINE.
4a36      5764: ;On exit first char of INLINE
4a36      5765: ;is compared with a CR.
4a36      5766: ;
4a36 48      5767: getdata: pha
4a37 a901 5768:      lda #1
4a39 8df002 5769:      sta crsinh
4a3c a454 5770:      ldy rowcrs
4a3e b96c61 5771:      lda colsize,y
4a41 a8      5772:      tay
4a42 8cda62 5773:      sty maxlen
4a45      5774: ;
4a45 a99b 5775: d.clr_buf1: lda #cr ;Clear input
4a47 996663 5776:      sta inline,y ;buffer
4a4a a9a0 5777:      lda #space+$80
4a4c 206437 5778:      jsr putchar
4a4f 88      5779:      dey
4a50 d0f3 5780:      bne d.clr_buf1
4a52      5781: ;      col 24
4a52 a918 5782:      lda #24
4a54 8555 5783:      sta colcrs
4a56      5784:
4a56 68      5785:      pla
4a57 20a74a 5786:      jsr d.sho1char
4a5a a001 5787:      ldy #1
4a5c      5788: ;
4a5c 84cd 5789: d.nxletr: sty ytemp ;Save index.
4a5e 20e937 5790:      jsr gk ;Get a character.
4a61 a4cd 5791:      ldy ytemp ;Restore index.
4a63 c97e 5792:      cmp #backsp
4a65 f013 5793:      beq d.bkspc
4a67 c99b 5794:      cmp #cr
4a69 f02f 5795:      beq d.got_line
4a6b c91b 5796:      cmp #esc
4a6d f02b 5797:      beq d.got_line
4a6f ccda62 5798:      cpy maxlen
4a72 b0e8 5799:      bcs d.nxletr
4a74 20a74a 5800:      jsr d.sho1char
4a77 c8      5801:      iny
4a78 d0e2 5802:      bne d.nxletr
4a7a      5803: ;
4a7a c000 5804: d.bkspc: cpy #0 ;Handle a
4a7c f0de 5805:      beq d.nxletr ;backspace.
4a7e 98      5806:      tya
```

```

4a7f 48 5807: pha
4a80 a97e 5808: lda #backsp
4a82 206437 5809: jsr putchar
4a85 a9a0 5810: lda #space+$80
4a87 206437 5811: jsr putchar
4a8a a91e 5812: lda #$1e
4a8c 206437 5813: jsr putchar
4a8f 68 5814: pla
4a90 a8 5815: tay
4a91 88 5816: dey
4a92 a99b 5817: lda #cr
4a94 996663 5818: sta inline,y
4a97 4c5c4a 5819: jmp d.nxletr
4a9a 5820: ;
4a9a 996663 5821: d.got_line: sta inline,y
4a9d c91b 5822: cmp #esc
4a9f f005 5823: beq d.gdx
4aa1 ad6663 5824: lda inline
4aa4 c99b 5825: cmp #cr
4aa6 60 5826: d.gdx: rts
4aa7 5827: ;
4aa7 996663 5828: d.sho1char: sta inline,y
4aaa 0980 5829: ora #$80
4aac 4c6437 5830: jmp putchar
4aaf 5831: ;-----
4aaf 5832: ;
4aaf 5833: ;Routine to save customized
4aaf 5834: ;version of this program.
4aaf 5835: ;
4aaf 20a83e 5836: saveprog: jsr cheknum
4ab2 b050 5837: bcs d.nosv
4ab4 8da661 5838: sta filename+1 ;disk.
4ab7 0980 5839: ora #$80
4ab9 8ddc4a 5840: sta thsprmpt
4abc 5841: ; position 8,21
4abc a908 5842: lda #8
4abe 8555 5843: sta colcrs
4ac0 a915 5844: lda #21
4ac2 8554 5845: sta rowcrs
4ac4 5846:
4ac4 208037 5847: jsr printsi
4ac7 a0d2e5 5848: dc.b <+128>," Ready Disk "
4ad3 e9eea0 5849: dc.b <+128>,"in Drive "
4adc b1a01b 5850: thsprmpt: dc.b <+128>,"1 ",ESC
4adf ff 5851: dc.b em
4ae0 5852: ; col 8
4ae0 a908 5853: lda #8
4ae2 8555 5854: sta colcrs
4ae4 5855:
4ae4 208037 5856: jsr printsi
4ae7 a0a0a0 5857: dc.b <+128>," Then Pres"
4af3 f3a0d2 5858: dc.b <+128>,"s RETURN "
4afe ff 5859: dc.b em

```



```
4aff 203738 5860:      jsr checkkey
4b02 9003 5861:      bcc d.chk3ok
4b04 4c5544 5862: d.nosv:  jmp options ;If ESC pressed.
4b07      5863:      ;
4b07      5864: d.chk3ok:
4b07      5865:      ; close 1
4b07 a210 5866:      ldx #$10
4b09 20823f 5867:      jsr close.r
4b0c      5868:      ;
4b0c a903 5869:      lda #open. ;Open MEGADIS.COM
4b0e 9d4203 5870:      sta iccom,x ;for output.
4b11 a908 5871:      lda #xwrite
4b13 9d4a03 5872:      sta icax1,x
4b16 a9a5 5873:      lda # low filename
4b18 9d4403 5874:      sta icbal,x
4b1b a961 5875:      lda # high filename
4b1d 9d4503 5876:      sta icbah,x
4b20 2056e4 5877:      jsr ciov
4b23 1003 5878:      bpl d.fileok
4b25 4cca43 5879:      jmp io_err
4b28      5880:      ;
4b28 a966 5881: d.fileok: lda # low header ;Write
4b2a 9d4403 5882:      sta icbal,x ;header bytes
4b2d a961 5883:      lda # high header ;to output
4b2f 9d4503 5884:      sta icbah,x ;file
4b32 a90b 5885:      lda #bput.
4b34 20443f 5886:      jsr sixbytes
4b37 1003 5887:      bpl d.hedok
4b39 4cca43 5888:      jmp io_err
4b3c      5889:      ;
4b3c a900 5890: d.hedok: lda # low beginsave
4b3e 9d4403 5891:      sta icbal,x ;Write the
4b41 a921 5892:      lda # high beginsave ;file
4b43 9d4503 5893:      sta icbah,x
4b46 ad6461 5894:      lda icsize
4b49 9d4803 5895:      sta icbll,x
4b4c ad6561 5896:      lda icsize+1
4b4f 9d4903 5897:      sta icblh,x
4b52 2056e4 5898:      jsr ciov
4b55 1003 5899:      bpl d.mainok
4b57 4cca43 5900:      jmp io_err
4b5a      5901:      ;
4b5a a9c4 5902: d.mainok: lda # low savend
4b5c 9d4403 5903:      sta icbal,x
4b5f a962 5904:      lda # high savend
4b61 9d4503 5905:      sta icbah,x
4b64 a90b 5906:      lda #bput.
4b66 20443f 5907:      jsr sixbytes
4b69      5908:      ;
4b69      5909: d.savok:
4b69      5910:      ; close 1 ;File saved.
4b69 a210 5911:      ldx #$10
4b6b 20823f 5912:      jsr close.r
```

```

4b6e      5913:
4b6e 4c5544 5914:      jmp options
4b71      5915: ;-----
4b71      5916:
4b71      5917:
4b71      5918:
4b71      5919: ;      .PAGE "Directory Routine"
4b71      5920: ;      .include dirs
4b71      5921:
4b71      5922: ;      ... DIRS ...
4b71      5923: ;
4b71      5924: ;Show disk directory.
4b71      5925: ;
4b71      5926: directory:
4b71      5927: ;
4b71 8dcf61 5928:      sta dir+1 ;Save drive #.
4b74 a900  5929:      lda #0
4b76 85cd  5930:      sta ytemp
4b78 20113f 5931:      jsr newscreen
4b7b      5932: ;      col 20
4b7b a914  5933:      lda #20
4b7d 8555  5934:      sta colcrs
4b7f      5935:
4b7f a901  5936:      lda #1
4b81 8554  5937:      sta rowcrs
4b83      5938: ;      close 1
4b83 a210  5939:      ldx #$10
4b85 20823f 5940:      jsr close.r
4b88      5941:
4b88 a903  5942:      lda #open. ;Open directory.
4b8a 9d4203 5943:      sta iccom,x
4b8d a907  5944:      lda #xdir
4b8f 9d4a03 5945:      sta icax1,x
4b92 a9ce  5946:      lda # low dir
4b94 9d4403 5947:      sta icbal,x
4b97 a961  5948:      lda # high dir
4b99 9d4503 5949:      sta icbah,x
4b9c 2056e4 5950:      jsr ciov
4b9f 1003  5951:      bpl d.dirline
4ba1 4cca43 5952:      jmp io_err
4ba4      5953: ;
4ba4 a210  5954: d.dirline: ldx #chan1 ;Input 1
4ba6 a966  5955:      lda # low inline ;filename.
4ba8 9d4403 5956:      sta icbal,x
4bab a963  5957:      lda # high inline
4bad 9d4503 5958:      sta icbah,x
4bb0 a905  5959:      lda #5
4bb2 9d4203 5960:      sta iccom,x
4bb5 a928  5961:      lda #40
4bb7 9d4803 5962:      sta icbll,x
4bba a900  5963:      lda #0
4bbc 9d4903 5964:      sta icblh,x
4bbf a555  5965:      lda colcrs ;Put cursor

```

```
4bc1 c913 5966:    cmp #19    ;where we want
4bc3 b006 5967:    bcs d.nxdrln ;to show this
4bc5 a914 5968:    lda #20    ;filename.
4bc7 8555 5969:    sta colcrs
4bc9 d005 5970:    bne d.dirin
4bcb 205e37 5971: d.nxdrln: jsr putcr
4bce a210 5972:    ldx #chan1
4bd0 2056e4 5973: d.dirin:  jsr ciov
4bd3 1007 5974:    bpl d.dirlineok ;Got a line.
4bd5 c088 5975:    cpy #$88    ;End of Directory?
4bd7 f009 5976:    beq d.dirend ;Yup
4bd9 4cca43 5977:    jmp io_err
4bdc      5978: ;
4bdc 20e54b 5979: d.dirlineok: jsr Ashodline
4bdf 4ca44b 5980:    jmp d.dirline
4be2      5981: ;
4be2 4c3e4c 5982: d.dirend: jmp d.dirdun
4be5      5983: ;
4be5      5984: ;
4be5 a554 5985: Ashodline: lda rowcrs ;Check for
4be7 c915 5986:    cmp #21    ;room on screen.
4be9 902d 5987:    bcc d.shod
4beb 208037 5988:    jsr printsi ;None here so
4bee 9b7f 5989:    dc.b cr,tab ;prompt for CR.
4bf0 a0cdef 5990:    dc.b <+128>," More Files Press RETURN "
4c09 ff 5991:    dc.b em
4c0a 202f38 5992:    jsr getreturn
4c0d 20113f 5993:    jsr newscreen
4c10      5994: ;    position 1,2
4c10 a901 5995:    lda #1
4c12 8555 5996:    sta colcrs
4c14 a902 5997:    lda #2
4c16 8554 5998:    sta rowcrs
4c18      5999:
4c18      6000: ;
4c18 a966 6001: d.shod:  lda # low inline ;Show the
4c1a 8d4403 6002:    sta icbal ;filename.
4c1d a963 6003:    lda # high inline
4c1f 8d4503 6004:    sta icbah
4c22      6005: ;
4c22 a200 6006:    ldx #0
4c24 a000 6007:    ldy #0
4c26 b96663 6008: d.drnn:  lda inline,y
4c29 c99b 6009:    cmp #cr
4c2b f003 6010:    beq d.drmm
4c2d c8 6011:    iny
4c2e d0f6 6012:    bne d.drnn
4c30      6013: ;
4c30 8c4803 6014: d.drmm:  sty icbll
4c33 8e4903 6015:    stx icblh
4c36 a90b 6016:    lda #bput.
4c38 8d4203 6017:    sta iccom
4c3b 4c56e4 6018:    jmp ciov
```

```

4c3e      6019: ;
4c3e 205e37 6020: d.dirdun: jsr putcr ;End of
4c41 205e37 6021:      jsr putcr ;Directory routine.
4c44      6022: ;      col 4
4c44 a904 6023:      lda #4
4c46 8555 6024:      sta colcrs
4c48      6025:
4c48 208037 6026:      jsr printsi
4c4b a0c5ee 6027:      dc.b <+128>," End of Directory Press any key "
4c6b ff 6028:      dc.b em
4c6c 20e637 6029:      jsr getkey
4c6f 20a83e 6030:      jsr cheknum
4c72 b003 6031:      bcs d.dirxit
4c74 4c714b 6032:      jmp directory
4c77      6033: ;
4c77 4c6822 6034: d.dirxit: jmp menu
4c7a      6035: ;-----
4c7a      6036:
4c7a      6037:
4c7a      6038: ;      .OPT NO LIST
4c7a      6039: ;      .PAGE "Label Table"
4c7a      6040: ;      .include lxe
4c7a      6041:
4c7a      6042: ; ... XL/XE LABELS ...
4c7a      6043: ;
4c7a      6044: page0:
4c7a      6045: ;
4c7a 4c4e46 6046:      dc.b "LNFLG "
4c80 4e4746 6047:      dc.b "NGFLAG"
4c86 434153 6048:      dc.b "CASINI"
4c8c 200120 6049:      dc.b 32,1," "
4c92 52414d 6050:      dc.b "RAMLO "
4c98 200120 6051:      dc.b 32,1," "
4c9e 545241 6052:      dc.b "TRAMSZ"
4ca4 434d43 6053:      dc.b "CMCMD "
4caa 574152 6054:      dc.b "WARMST"
4cb0 424f4f 6055:      dc.b "BOOT? "
4cb6 444f53 6056:      dc.b "DOSVEC"
4cbc 200120 6057:      dc.b 32,1," "
4cc2 444f53 6058:      dc.b "DOSINI"
4cc8 200120 6059:      dc.b 32,1," "
4cce 415050 6060:      dc.b "APPMHI"
4cd4 200120 6061:      dc.b 32,1," "
4cda 504f4b 6062:      dc.b "POKMSK"
4ce0 42524b 6063:      dc.b "BRKKEY"
4ce6 525443 6064:      dc.b "RTCLOK"
4cec 200120 6065:      dc.b 32,1," "
4cf2 200220 6066:      dc.b 32,2," "
4cf8 425546 6067:      dc.b "BUFADR"
4cfe 200120 6068:      dc.b 32,1," "
4d04 494343 6069:      dc.b "ICCOMT"
4d0a 44534b 6070:      dc.b "DSKFMS"
4d10 200120 6071:      dc.b 32,1," "

```

4d16	44534b	6072:	dc.b "DSKUTL"
4d1c	200120	6073:	dc.b 32,1," "
4d22	414255	6074:	dc.b "ABUFPT"
4d28	200120	6075:	dc.b 32,1," "
4d2e	200220	6076:	dc.b 32,2," "
4d34	200320	6077:	dc.b 32,3," "
4d3a	494348	6078:	dc.b "ICHIDZ"
4d40	494344	6079:	dc.b "ICDNOZ"
4d46	494343	6080:	dc.b "ICCOMZ"
4d4c	494353	6081:	dc.b "ICSTAZ"
4d52	494342	6082:	dc.b "ICBALZ"
4d58	494342	6083:	dc.b "ICBAHZ"
4d5e	494350	6084:	dc.b "ICPTLZ"
4d64	494350	6085:	dc.b "ICPTHZ"
4d6a	494342	6086:	dc.b "ICBL LZ"
4d70	494342	6087:	dc.b "ICBLHZ"
4d76	494341	6088:	dc.b "ICAX1Z"
4d7c	494341	6089:	dc.b "ICAX2Z"
4d82	494341	6090:	dc.b "ICAX3Z"
4d88	494341	6091:	dc.b "ICAX4Z"
4d8e	494341	6092:	dc.b "ICAX5Z"
4d94	43494f	6093:	dc.b "CIOCHR"
4d9a	535441	6094:	dc.b "STATUS"
4da0	43484b	6095:	dc.b "CHKSUM"
4da6	425546	6096:	dc.b "BUFRLO"
4dac	425546	6097:	dc.b "BUFRHI"
4db2	424645	6098:	dc.b "BFENLO"
4db8	424645	6099:	dc.b "BFENHI"
4dbe	4c5445	6100:	dc.b "LTEMP "
4dc4	200120	6101:	dc.b 32,1," "
4dca	425546	6102:	dc.b "BUFRFL"
4dd0	524543	6103:	dc.b "RECVDN"
4dd6	584d54	6104:	dc.b "XMTDON"
4ddc	43484b	6105:	dc.b "CHKSNT"
4de2	4e4f43	6106:	dc.b "NOCKSM"
4de8	425054	6107:	dc.b "BPTR "
4dee	465459	6108:	dc.b "FTYPE "
4df4	46454f	6109:	dc.b "FEOF "
4dfa	465245	6110:	dc.b "FREQ "
4e00	534f55	6111:	dc.b "SOUNDR"
4e06	435249	6112:	dc.b "CRITIC"
4e0c	5a4255	6113:	dc.b "ZBUFP "
4e12	200120	6114:	dc.b 32,1," "
4e18	5a4452	6115:	dc.b "ZDRVA "
4e1e	200120	6116:	dc.b 32,1," "
4e24	5a5342	6117:	dc.b "ZSBA "
4e2a	200120	6118:	dc.b 32,1," "
4e30	455252	6119:	dc.b "ERRNO "
4e36	5a4348	6120:	dc.b "ZCHAIN"
4e3c	200120	6121:	dc.b 32,1," "
4e42	445354	6122:	dc.b "DSTAT "
4e48	415452	6123:	dc.b "ATTRACT"
4e4e	44524b	6124:	dc.b "DRKMSK"

```

4e54 434f4c 6125:    dc.b "COLRSH"
4e5a 54454d 6126:    dc.b "TEMP "
4e60 484f4c 6127:    dc.b "HOLD1 "
4e66 4c4d41 6128:    dc.b "LMARGN"
4e6c 524d41 6129:    dc.b "RMARGN"
4e72 524f57 6130:    dc.b "ROWCRS"
4e78 434f4c 6131:    dc.b "COLCRS"
4e7e 200120 6132:    dc.b 32,1," "
4e84 44494e 6133:    dc.b "DINDEX"
4e8a 534156 6134:    dc.b "SAVMSC"
4e90 200120 6135:    dc.b 32,1," "
4e96 4f4c44 6136:    dc.b "OLDROW"
4e9c 4f4c44 6137:    dc.b "OLDCOL"
4ea2 200120 6138:    dc.b 32,1," "
4ea8 4f4c44 6139:    dc.b "OLDCHR"
4eae 4f4c44 6140:    dc.b "OLDADR"
4eb4 200120 6141:    dc.b 32,1," "
4eba 464b44 6142:    dc.b "FKDEF "
4ec0 200120 6143:    dc.b 32,1," "
4ec6 50414c 6144:    dc.b "PALNTS"
4ecc 4c4f47 6145:    dc.b "LOGCOL"
4ed2 414452 6146:    dc.b "ADRESS"
4ed8 200120 6147:    dc.b 32,1," "
4ede 4d4c54 6148:    dc.b "MLTTMP"
4ee4 200120 6149:    dc.b 32,1," "
4eea 534156 6150:    dc.b "SAVADR"
4ef0 200120 6151:    dc.b 32,1," "
4ef6 52414d 6152:    dc.b "RAMTOP"
4efc 425546 6153:    dc.b "BUFCNT"
4f02 425546 6154:    dc.b "BUFSTR"
4f08 200120 6155:    dc.b 32,1," "
4f0e 424954 6156:    dc.b "BITMSK"
4f14 534846 6157:    dc.b "SHFAMT"
4f1a 524f57 6158:    dc.b "ROWAC "
4f20 200120 6159:    dc.b 32,1," "
4f26 434f4c 6160:    dc.b "COLAC "
4f2c 200120 6161:    dc.b 32,1," "
4f32 454e44 6162:    dc.b "ENDPT "
4f38 200120 6163:    dc.b 32,1," "
4f3e 44454c 6164:    dc.b "DELTAR"
4f44 44454c 6165:    dc.b "DELTAC"
4f4a 200120 6166:    dc.b 32,1," "
4f50 4b4559 6167:    dc.b "KEYDEF"
4f56 200120 6168:    dc.b 32,1," "
4f5c 535750 6169:    dc.b "SWPFLG"
4f62 484f4c 6170:    dc.b "HOLDCH"
4f68 494e53 6171:    dc.b "INSDAT"
4f6e 434f55 6172:    dc.b "COUNTR"
4f74 200120 6173:    dc.b 32,1," "
4f7a      6174: ;
4f7a      6175: ;-----
4f7a      6176: ;
4f7a      6177: page2:

```

4f7a	6178:	;	
4f7a	564453	6179:	dc.b "VDSLST"
4f80	200120	6180:	dc.b 32,1," "
4f86	565052	6181:	dc.b "VPRCED"
4f8c	200120	6182:	dc.b 32,1," "
4f92	56494e	6183:	dc.b "VINTER"
4f98	200120	6184:	dc.b 32,1," "
4f9e	564252	6185:	dc.b "VBREAK"
4fa4	200120	6186:	dc.b 32,1," "
4faa	564b45	6187:	dc.b "VKEYBD"
4fb0	200120	6188:	dc.b 32,1," "
4fb6	565345	6189:	dc.b "VSERIN"
4fbc	200120	6190:	dc.b 32,1," "
4fc2	565345	6191:	dc.b "VSEROR"
4fc8	200120	6192:	dc.b 32,1," "
4fce	565345	6193:	dc.b "VSEROC"
4fd4	200120	6194:	dc.b 32,1," "
4fda	565449	6195:	dc.b "VTIMR1"
4fe0	200120	6196:	dc.b 32,1," "
4fe6	565449	6197:	dc.b "VTIMR2"
4fec	200120	6198:	dc.b 32,1," "
4ff2	565449	6199:	dc.b "VTIMR4"
4ff8	200120	6200:	dc.b 32,1," "
4ffe	56494d	6201:	dc.b "VIMIRQ"
5004	200120	6202:	dc.b 32,1," "
500a	434454	6203:	dc.b "CDTMV1"
5010	200120	6204:	dc.b 32,1," "
5016	434454	6205:	dc.b "CDTMV2"
501c	200120	6206:	dc.b 32,1," "
5022	434454	6207:	dc.b "CDTMV3"
5028	200120	6208:	dc.b 32,1," "
502e	434454	6209:	dc.b "CDTMV4"
5034	200120	6210:	dc.b 32,1," "
503a	434454	6211:	dc.b "CDTMV5"
5040	200120	6212:	dc.b 32,1," "
5046	565642	6213:	dc.b "VVBLKI"
504c	200120	6214:	dc.b 32,1," "
5052	565642	6215:	dc.b "VVBLKD"
5058	200120	6216:	dc.b 32,1," "
505e	434454	6217:	dc.b "CDTMA1"
5064	200120	6218:	dc.b 32,1," "
506a	434454	6219:	dc.b "CDTMA2"
5070	200120	6220:	dc.b 32,1," "
5076	434454	6221:	dc.b "CDTMF3"
507c	535254	6222:	dc.b "SRTIMR"
5082	434454	6223:	dc.b "CDTMF4"
5088	494e54	6224:	dc.b "INTMP"
508e	434454	6225:	dc.b "CDTMF5"
5094	53444d	6226:	dc.b "SDMCTL"
509a	53444c	6227:	dc.b "SDLSTL"
50a0	53444c	6228:	dc.b "SDLSTH"
50a6	53534b	6229:	dc.b "SSKCTL"
50ac	4c434f	6230:	dc.b "LCOUNT"

50b2	4c5045	6231:	dc.b "LPENH "
50b8	4c5045	6232:	dc.b "LPENV "
50be	200120	6233:	dc.b 32,1," "
50c4	200220	6234:	dc.b 32,2," "
50ca	565049	6235:	dc.b "VPIRQ "
50d0	200120	6236:	dc.b 32,1," "
50d6	434445	6237:	dc.b "CDEVIC"
50dc	43434f	6238:	dc.b "CCOMND"
50e2	434155	6239:	dc.b "CAUX1 "
50e8	434155	6240:	dc.b "CAUX2 "
50ee	54454d	6241:	dc.b "TEMP "
50f4	455252	6242:	dc.b "ERRFLG"
50fa	44464c	6243:	dc.b "DFLAGS"
5100	444253	6244:	dc.b "DBSECT"
5106	424f4f	6245:	dc.b "BOOTAD"
510c	200120	6246:	dc.b 32,1," "
5112	434f4c	6247:	dc.b "COLDST"
5118	524543	6248:	dc.b "RECLen"
511e	44534b	6249:	dc.b "DSKTIM"
5124	504456	6250:	dc.b "PDVMSK"
512a	534850	6251:	dc.b "SHPDVS"
5130	50444d	6252:	dc.b "PDMSK "
5136	52454c	6253:	dc.b "RELADR"
513c	200120	6254:	dc.b 32,1," "
5142	505054	6255:	dc.b "PPTMPA"
5148	505054	6256:	dc.b "PPTMPX"
514e	200120	6257:	dc.b 32,1," "
5154	200220	6258:	dc.b 32,2," "
515a	200320	6259:	dc.b 32,3," "
5160	200420	6260:	dc.b 32,4," "
5166	200520	6261:	dc.b 32,5," "
516c	200620	6262:	dc.b 32,6," "
5172	200720	6263:	dc.b 32,7," "
5178	200820	6264:	dc.b 32,8," "
517e	200920	6265:	dc.b 32,9," "
5184	200a20	6266:	dc.b 32,10," "
518a	200b20	6267:	dc.b 32,11," "
5190	200c20	6268:	dc.b 32,12," "
5196	200d20	6269:	dc.b 32,13," "
519c	200e20	6270:	dc.b 32,14," "
51a2	200f20	6271:	dc.b 32,15," "
51a8	201020	6272:	dc.b 32,16," "
51ae	201120	6273:	dc.b 32,17," "
51b4	201220	6274:	dc.b 32,18," "
51ba	201320	6275:	dc.b 32,19," "
51c0	201420	6276:	dc.b 32,20," "
51c6	201520	6277:	dc.b 32,21," "
51cc	201620	6278:	dc.b 32,22," "
51d2	201720	6279:	dc.b 32,23," "
51d8	201820	6280:	dc.b 32,24," "
51de	201920	6281:	dc.b 32,25," "
51e4	201a20	6282:	dc.b 32,26," "
51ea	201b20	6283:	dc.b 32,27," "

51f0	201c20	6284:	dc.b 32,28," "
51f6	201d20	6285:	dc.b 32,29," "
51fc	434853	6286:	dc.b "CHSALT"
5202	565346	6287:	dc.b "VSFLAG"
5208	4b4559	6288:	dc.b "KEYDIS"
520e	46494e	6289:	dc.b "FINE "
5214	475052	6290:	dc.b "GPRIOR"
521a	504144	6291:	dc.b "PADDL0"
5220	504144	6292:	dc.b "PADDL1"
5226	504144	6293:	dc.b "PADDL2"
522c	504144	6294:	dc.b "PADDL3"
5232	504144	6295:	dc.b "PADDL4"
5238	504144	6296:	dc.b "PADDL5"
523e	504144	6297:	dc.b "PADDL6"
5244	504144	6298:	dc.b "PADDL7"
524a	535449	6299:	dc.b "STICK0"
5250	535449	6300:	dc.b "STICK1"
5256	535449	6301:	dc.b "STICK2"
525c	535449	6302:	dc.b "STICK3"
5262	505452	6303:	dc.b "PTRIG0"
5268	505452	6304:	dc.b "PTRIG1"
526e	505452	6305:	dc.b "PTRIG2"
5274	505452	6306:	dc.b "PTRIG3"
527a	505452	6307:	dc.b "PTRIG4"
5280	505452	6308:	dc.b "PTRIG5"
5286	505452	6309:	dc.b "PTRIG6"
528c	505452	6310:	dc.b "PTRIG7"
5292	535452	6311:	dc.b "STRIG0"
5298	535452	6312:	dc.b "STRIG1"
529e	535452	6313:	dc.b "STRIG2"
52a4	535452	6314:	dc.b "STRIG3"
52aa	484942	6315:	dc.b "HIBYTE"
52b0	200120	6316:	dc.b 32,1," "
52b6	200220	6317:	dc.b 32,2," "
52bc	494d41	6318:	dc.b "IMASK "
52c2	4a5645	6319:	dc.b "JVECK "
52c8	200120	6320:	dc.b 32,1," "
52ce	4e4557	6321:	dc.b "NEWADR"
52d4	200120	6322:	dc.b 32,1," "
52da	545854	6323:	dc.b "TXTROW"
52e0	545854	6324:	dc.b "TXTCOL"
52e6	200120	6325:	dc.b 32,1," "
52ec	54494e	6326:	dc.b "TINDEX"
52f2	545854	6327:	dc.b "TXTMSC"
52f8	200120	6328:	dc.b 32,1," "
52fe	545854	6329:	dc.b "TXTOLD"
5304	200120	6330:	dc.b 32,1," "
530a	200220	6331:	dc.b 32,2," "
5310	200320	6332:	dc.b 32,3," "
5316	200420	6333:	dc.b 32,4," "
531c	200520	6334:	dc.b 32,5," "
5322	435245	6335:	dc.b "CRETRY"
5328	484f4c	6336:	dc.b "HOLD3 "

532e	535542	6337:	dc.b "SUBTMP"
5334	484f4c	6338:	dc.b "HOLD2 "
533a	444d41	6339:	dc.b "DMASK "
5340	544d50	6340:	dc.b "TMPLBT"
5346	455343	6341:	dc.b "ESCFLG"
534c	544142	6342:	dc.b "TABMAP"
5352	200120	6343:	dc.b 32,1," "
5358	200220	6344:	dc.b 32,2," "
535e	200320	6345:	dc.b 32,3," "
5364	200420	6346:	dc.b 32,4," "
536a	200520	6347:	dc.b 32,5," "
5370	200620	6348:	dc.b 32,6," "
5376	200720	6349:	dc.b 32,7," "
537c	200820	6350:	dc.b 32,8," "
5382	200920	6351:	dc.b 32,9," "
5388	200a20	6352:	dc.b 32,10," "
538e	200b20	6353:	dc.b 32,11," "
5394	200c20	6354:	dc.b 32,12," "
539a	200d20	6355:	dc.b 32,13," "
53a0	200e20	6356:	dc.b 32,14," "
53a6	4c4f47	6357:	dc.b "LOGMAP"
53ac	200120	6358:	dc.b 32,1," "
53b2	200220	6359:	dc.b 32,2," "
53b8	200320	6360:	dc.b 32,3," "
53be	494e56	6361:	dc.b "INVFLG"
53c4	46494c	6362:	dc.b "FILFLG"
53ca	544d50	6363:	dc.b "TMPROW"
53d0	544d50	6364:	dc.b "TMPCOL"
53d6	200120	6365:	dc.b 32,1," "
53dc	534352	6366:	dc.b "SCRFLG"
53e2	484f4c	6367:	dc.b "HOLD4 "
53e8	445245	6368:	dc.b "DRETRY"
53ee	534846	6369:	dc.b "SHFLOK"
53f4	424f54	6370:	dc.b "BOTSCR"
53fa	50434f	6371:	dc.b "PCOLR0"
5400	50434f	6372:	dc.b "PCOLR1"
5406	50434f	6373:	dc.b "PCOLR2"
540c	50434f	6374:	dc.b "PCOLR3"
5412	434f4c	6375:	dc.b "COLOR0"
5418	434f4c	6376:	dc.b "COLOR1"
541e	434f4c	6377:	dc.b "COLOR2"
5424	434f4c	6378:	dc.b "COLOR3"
542a	434f4c	6379:	dc.b "COLOR4"
5430	52554e	6380:	dc.b "RUNADR"
5436	200120	6381:	dc.b 32,1," "
543c	484955	6382:	dc.b "HIUSED"
5442	200120	6383:	dc.b 32,1," "
5448	5a4849	6384:	dc.b "ZHIUSE"
544e	200120	6385:	dc.b 32,1," "
5454	474259	6386:	dc.b "GBYTEA"
545a	200120	6387:	dc.b 32,1," "
5460	4c4f41	6388:	dc.b "LOADAD"
5466	200120	6389:	dc.b 32,1," "

```

546c 5a4c4f 6390:    dc.b "ZLOADA"
5472 200120 6391:    dc.b 32,1," "
5478 445343 6392:    dc.b "DSCTLN"
547e 200120 6393:    dc.b 32,1," "
5484 41434d 6394:    dc.b "ACMISR"
548a 200120 6395:    dc.b 32,1," "
5490 4b5250 6396:    dc.b "KRPDEL"
5496 4b4559 6397:    dc.b "KEYREP"
549c 4e4f43 6398:    dc.b "NOCLIK"
54a2 48454c 6399:    dc.b "HELPGF"
54a8 444d41 6400:    dc.b "DMASAV"
54ae 504250 6401:    dc.b "PBPNT "
54b4 504255 6402:    dc.b "PBUFSZ"
54ba 52554e 6403:    dc.b "RUNAD "
54c0 200120 6404:    dc.b 32,1," "
54c6 494e49 6405:    dc.b "INITAD"
54cc 200120 6406:    dc.b 32,1," "
54d2 52414d 6407:    dc.b "RAMSIZ"
54d8 4d454d 6408:    dc.b "MEMTOP"
54de 200120 6409:    dc.b 32,1," "
54e4 4d454d 6410:    dc.b "MEMLO "
54ea 200120 6411:    dc.b 32,1," "
54f0 484e44 6412:    dc.b "HNDLOD"
54f6 445653 6413:    dc.b "DVSTAT"
54fc 200120 6414:    dc.b 32,1," "
5502 200220 6415:    dc.b 32,2," "
5508 200320 6416:    dc.b 32,3," "
550e 434241 6417:    dc.b "CBAUDL"
5514 434241 6418:    dc.b "CBAUDH"
551a 435253 6419:    dc.b "CRSINH"
5520 4b4559 6420:    dc.b "KEYDEL"
5526 434831 6421:    dc.b "CH1 "
552c 434841 6422:    dc.b "CHACT "
5532 434842 6423:    dc.b "CHBAS "
5538 4e4557 6424:    dc.b "NEWROW"
553e 4e4557 6425:    dc.b "NEWCOL"
5544 200120 6426:    dc.b 32,1," "
554a 524f57 6427:    dc.b "ROWINC"
5550 434f4c 6428:    dc.b "COLINC"
5556 434841 6429:    dc.b "CHAR "
555c 415441 6430:    dc.b "ATACHR"
5562 434820 6431:    dc.b "CH "
5568 46494c 6432:    dc.b "FILDAT"
556e 445350 6433:    dc.b "DSPFLG"
5574 535346 6434:    dc.b "SSFLAG"
557a      6435: ;
557a      6436: ;-----
557a      6437: ;
557a      6438: page3:
557a      6439: ;
557a 444445 6440:    dc.b "DDEVIC"
5580 44554e 6441:    dc.b "DUNIT "
5586 44434f 6442:    dc.b "DCOMND"

```

558c	445354	6443:	dc.b "DSTATS"
5592	444255	6444:	dc.b "DBUFLO"
5598	444255	6445:	dc.b "DBUFHI"
559e	445449	6446:	dc.b "DTIMLO"
55a4	44554e	6447:	dc.b "DUNUSE"
55aa	444259	6448:	dc.b "DBYTLO"
55b0	444259	6449:	dc.b "DBYTHI"
55b6	444155	6450:	dc.b "DAUX1 "
55bc	444155	6451:	dc.b "DAUX2 "
55c2	54494d	6452:	dc.b "TIMER1"
55c8	200120	6453:	dc.b 32,1," "
55ce	4a4d50	6454:	dc.b "JMPERS"
55d4	434153	6455:	dc.b "CASFLG"
55da	54494d	6456:	dc.b "TIMER2"
55e0	200120	6457:	dc.b 32,1," "
55e6	54454d	6458:	dc.b "TEMP1 "
55ec	54454d	6459:	dc.b "TEMP2 "
55f2	505449	6460:	dc.b "PTIMOT"
55f8	54454d	6461:	dc.b "TEMP3 "
55fe	534156	6462:	dc.b "SAVIO "
5604	54494d	6463:	dc.b "TIMFLG"
560a	535441	6464:	dc.b "STACKP"
5610	545354	6465:	dc.b "TSTAT "
5616	484154	6466:	dc.b "HATABS"
561c	200120	6467:	dc.b 32,1," "
5622	200220	6468:	dc.b 32,2," "
5628	200320	6469:	dc.b 32,3," "
562e	200420	6470:	dc.b 32,4," "
5634	200520	6471:	dc.b 32,5," "
563a	200620	6472:	dc.b 32,6," "
5640	200720	6473:	dc.b 32,7," "
5646	200820	6474:	dc.b 32,8," "
564c	200920	6475:	dc.b 32,9," "
5652	200a20	6476:	dc.b 32,10," "
5658	200b20	6477:	dc.b 32,11," "
565e	200c20	6478:	dc.b 32,12," "
5664	200d20	6479:	dc.b 32,13," "
566a	200e20	6480:	dc.b 32,14," "
5670	200f20	6481:	dc.b 32,15," "
5676	201020	6482:	dc.b 32,16," "
567c	201120	6483:	dc.b 32,17," "
5682	201220	6484:	dc.b 32,18," "
5688	201320	6485:	dc.b 32,19," "
568e	201420	6486:	dc.b 32,20," "
5694	201520	6487:	dc.b 32,21," "
569a	201620	6488:	dc.b 32,22," "
56a0	201720	6489:	dc.b 32,23," "
56a6	201820	6490:	dc.b 32,24," "
56ac	201920	6491:	dc.b 32,25," "
56b2	201a20	6492:	dc.b 32,26," "
56b8	201b20	6493:	dc.b 32,27," "
56be	201c20	6494:	dc.b 32,28," "
56c4	201d20	6495:	dc.b 32,29," "

```

56ca 201e20 6496:    dc.b 32,30,"  "
56d0 201f20 6497:    dc.b 32,31,"  "
56d6 202020 6498:    dc.b 32,32,"  "
56dc 202120 6499:    dc.b 32,33,"  "
56e2 202220 6500:    dc.b 32,34,"  "
56e8 505550 6501:    dc.b "PUPBT1"
56ee 505550 6502:    dc.b "PUPBT2"
56f4 505550 6503:    dc.b "PUPBT3"
56fa 494348 6504:    dc.b "ICHID "
5700 494344 6505:    dc.b "ICDNO "
5706 494343 6506:    dc.b "ICCOM "
570c 494353 6507:    dc.b "ICSTA "
5712 494342 6508:    dc.b "ICBAL "
5718 494342 6509:    dc.b "ICBAH "
571e 494350 6510:    dc.b "ICPTL "
5724 494350 6511:    dc.b "ICPTH "
572a 494342 6512:    dc.b "ICBLL "
5730 494342 6513:    dc.b "ICBLH "
5736 494341 6514:    dc.b "ICAX1 "
573c 494341 6515:    dc.b "ICAX2 "
5742 494341 6516:    dc.b "ICAX3 "
5748 494341 6517:    dc.b "ICAX4 "
574e 494341 6518:    dc.b "ICAX5 "
5754 494341 6519:    dc.b "ICAX6 "
575a      6520: ;
575a      6521: ;-----
575a      6522: ;
575a      6523: gtia_r:
575a      6524: ;
575a 4d3050 6525:    dc.b "M0PF "
5760 4d3150 6526:    dc.b "M1PF "
5766 4d3250 6527:    dc.b "M2PF "
576c 4d3350 6528:    dc.b "M3PF "
5772 503050 6529:    dc.b "P0PF "
5778 503150 6530:    dc.b "P1PF "
577e 503250 6531:    dc.b "P2PF "
5784 503350 6532:    dc.b "P3PF "
578a 4d3050 6533:    dc.b "M0PL "
5790 4d3150 6534:    dc.b "M1PL "
5796 4d3250 6535:    dc.b "M2PL "
579c 4d3350 6536:    dc.b "M3PL "
57a2 503050 6537:    dc.b "P0PL "
57a8 503150 6538:    dc.b "P1PL "
57ae 503250 6539:    dc.b "P2PL "
57b4 503350 6540:    dc.b "P3PL "
57ba 545249 6541:    dc.b "TRIG0 "
57c0 545249 6542:    dc.b "TRIG1 "
57c6 545249 6543:    dc.b "TRIG2 "
57cc 545249 6544:    dc.b "TRIG3 "
57d2 50414c 6545:    dc.b "PAL  "
57d8      6546: ;
57d8      6547: ;-----
57d8      6548: ;

```

```
57d8      6549: gtia_w:
57d8      6550: ;
57d8 48504f 6551:      dc.b "HPOSP0"
57de 48504f 6552:      dc.b "HPOSP1"
57e4 48504f 6553:      dc.b "HPOSP2"
57ea 48504f 6554:      dc.b "HPOSP3"
57f0 48504f 6555:      dc.b "HPOSM0"
57f6 48504f 6556:      dc.b "HPOSM1"
57fc 48504f 6557:      dc.b "HPOSM2"
5802 48504f 6558:      dc.b "HPOSM3"
5808 53495a 6559:      dc.b "SIZEP0"
580e 53495a 6560:      dc.b "SIZEP1"
5814 53495a 6561:      dc.b "SIZEP2"
581a 53495a 6562:      dc.b "SIZEP3"
5820 53495a 6563:      dc.b "SIZEM "
5826 475241 6564:      dc.b "GRAFP0"
582c 475241 6565:      dc.b "GRAFP1"
5832 475241 6566:      dc.b "GRAFP2"
5838 475241 6567:      dc.b "GRAFP3"
583e 475241 6568:      dc.b "GRAFM "
5844 434f4c 6569:      dc.b "COLPM0"
584a 434f4c 6570:      dc.b "COLPM1"
5850 434f4c 6571:      dc.b "COLPM2"
5856      6572: ;
5856      6573: ;-----
5856      6574: ;
5856      6575: pokey_r:
5856      6576: ;
5856 504f54 6577:      dc.b "POT0 "
585c 504f54 6578:      dc.b "POT1 "
5862 504f54 6579:      dc.b "POT2 "
5868 504f54 6580:      dc.b "POT3 "
586e 504f54 6581:      dc.b "POT4 "
5874 504f54 6582:      dc.b "POT5 "
587a 504f54 6583:      dc.b "POT6 "
5880 504f54 6584:      dc.b "POT7 "
5886 414c4c 6585:      dc.b "ALLPOT"
588c 4b4243 6586:      dc.b "KBCODE"
5892 52414e 6587:      dc.b "RANDOM"
5898 504f54 6588:      dc.b "POTGO "
589e 200120 6589:      dc.b 32,1," "
58a4 534552 6590:      dc.b "SERIN "
58aa 495251 6591:      dc.b "IRQST "
58b0 534b53 6592:      dc.b "SKSTAT"
58b6      6593: ;
58b6      6594: ;-----
58b6      6595: ;
58b6      6596: pokey_w:
58b6      6597: ;
58b6 415544 6598:      dc.b "AUDF1 "
58bc 415544 6599:      dc.b "AUDC1 "
58c2 415544 6600:      dc.b "AUDF2 "
58c8 415544 6601:      dc.b "AUDC2 "
```

```

58ce 415544 6602:      dc.b "AUDF3 "
58d4 415544 6603:      dc.b "AUDC3 "
58da 415544 6604:      dc.b "AUDF4 "
58e0 415544 6605:      dc.b "AUDC4 "
58e6 415544 6606:      dc.b "AUDCTL"
58ec 535449 6607:      dc.b "STIMER"
58f2 534b52 6608:      dc.b "SKRES "
58f8 200120 6609:      dc.b 32,1,"  "
58fe 200220 6610:      dc.b 32,2,"  "
5904 534552 6611:      dc.b "SEROUT"
590a 495251 6612:      dc.b "IRQEN "
5910 534b43 6613:      dc.b "SKCTL "
5916      6614: ;
5916      6615: ;-----
5916      6616: ;
5916      6617: pia:
5916      6618: ;
5916 504f52 6619:      dc.b "PORTA "
591c 504f52 6620:      dc.b "PORTB "
5922 504143 6621:      dc.b "PACTL "
5928 504243 6622:      dc.b "PBCTL "
592e      6623: ;
592e      6624: ;-----
592e      6625: ;
592e      6626: antic:
592e      6627: ;
592e 444d41 6628:      dc.b "DMACTL"
5934 434841 6629:      dc.b "CHACTL"
593a 444c49 6630:      dc.b "DLISTL"
5940 444c49 6631:      dc.b "DLISTH"
5946 485343 6632:      dc.b "HSCROL"
594c 565343 6633:      dc.b "VSCROL"
5952 200120 6634:      dc.b 32,1,"  "
5958 504d42 6635:      dc.b "PMBASE"
595e 200120 6636:      dc.b 32,1,"  "
5964 434842 6637:      dc.b "CHBASE"
596a 575359 6638:      dc.b "WSYNC "
5970 56434f 6639:      dc.b "VCOUNT"
5976 50454e 6640:      dc.b "PENH  "
597c 50454e 6641:      dc.b "PENV  "
5982 4e4d49 6642:      dc.b "NMIEN "
5988 4e4d49 6643:      dc.b "NMIST "
598e      6644: ;
598e      6645: ;-----
598e      6646: ;
598e      6647: os:
598e      6648: ;
598e 454449 6649:      dc.b "EDITRV"
5994 200120 6650:      dc.b 32,1,"  "
599a 200220 6651:      dc.b 32,2,"  "
59a0 200320 6652:      dc.b 32,3,"  "
59a6 200420 6653:      dc.b 32,4,"  "
59ac 200520 6654:      dc.b 32,5,"  "

```

```

59b2 200620 6655:    dc.b 32,6,"  "
59b8 200720 6656:    dc.b 32,7,"  "
59be 200820 6657:    dc.b 32,8,"  "
59c4 200920 6658:    dc.b 32,9,"  "
59ca 200a20 6659:    dc.b 32,10," "
59d0 200b20 6660:    dc.b 32,11," "
59d6 200c20 6661:    dc.b 32,12," "
59dc 200d20 6662:    dc.b 32,13," "
59e2 200e20 6663:    dc.b 32,14," "
59e8 200f20 6664:    dc.b 32,15," "
59ee 534352 6665:    dc.b "SCRENV"
59f4 200120 6666:    dc.b 32,1,"  "
59fa 200220 6667:    dc.b 32,2,"  "
5a00 200320 6668:    dc.b 32,3,"  "
5a06 200420 6669:    dc.b 32,4,"  "
5a0c 200520 6670:    dc.b 32,5,"  "
5a12 200620 6671:    dc.b 32,6,"  "
5a18 200720 6672:    dc.b 32,7,"  "
5a1e 200820 6673:    dc.b 32,8,"  "
5a24 200920 6674:    dc.b 32,9,"  "
5a2a 200a20 6675:    dc.b 32,10," "
5a30 200b20 6676:    dc.b 32,11," "
5a36 200c20 6677:    dc.b 32,12," "
5a3c 200d20 6678:    dc.b 32,13," "
5a42 200e20 6679:    dc.b 32,14," "
5a48 200f20 6680:    dc.b 32,15," "
5a4e 4b4559 6681:    dc.b "KEYBDV"
5a54 200120 6682:    dc.b 32,1,"  "
5a5a 200220 6683:    dc.b 32,2,"  "
5a60 200320 6684:    dc.b 32,3,"  "
5a66 200420 6685:    dc.b 32,4,"  "
5a6c 200520 6686:    dc.b 32,5,"  "
5a72 200620 6687:    dc.b 32,6,"  "
5a78 200720 6688:    dc.b 32,7,"  "
5a7e 200820 6689:    dc.b 32,8,"  "
5a84 200920 6690:    dc.b 32,9,"  "
5a8a 200a20 6691:    dc.b 32,10," "
5a90 200b20 6692:    dc.b 32,11," "
5a96 200c20 6693:    dc.b 32,12," "
5a9c 200d20 6694:    dc.b 32,13," "
5aa2 200e20 6695:    dc.b 32,14," "
5aa8 200f20 6696:    dc.b 32,15," "
5aae 505249 6697:    dc.b "PRINTV"
5ab4 200120 6698:    dc.b 32,1,"  "
5aba 200220 6699:    dc.b 32,2,"  "
5ac0 200320 6700:    dc.b 32,3,"  "
5ac6 200420 6701:    dc.b 32,4,"  "
5acc 200520 6702:    dc.b 32,5,"  "
5ad2 200620 6703:    dc.b 32,6,"  "
5ad8 200720 6704:    dc.b 32,7,"  "
5ade 200820 6705:    dc.b 32,8,"  "
5ae4 200920 6706:    dc.b 32,9,"  "
5aea 200a20 6707:    dc.b 32,10," "

```



```

5af0 200b20 6708:    dc.b 32,11,"  "
5af6 200c20 6709:    dc.b 32,12,"  "
5afc 200d20 6710:    dc.b 32,13,"  "
5b02 200e20 6711:    dc.b 32,14,"  "
5b08 200f20 6712:    dc.b 32,15,"  "
5b0e 434153 6713:    dc.b "CASETV"
5b14 200120 6714:    dc.b 32,1,"  "
5b1a 200220 6715:    dc.b 32,2,"  "
5b20 200320 6716:    dc.b 32,3,"  "
5b26 200420 6717:    dc.b 32,4,"  "
5b2c 200520 6718:    dc.b 32,5,"  "
5b32 200620 6719:    dc.b 32,6,"  "
5b38 200720 6720:    dc.b 32,7,"  "
5b3e 200820 6721:    dc.b 32,8,"  "
5b44 200920 6722:    dc.b 32,9,"  "
5b4a 200a20 6723:    dc.b 32,10," "
5b50 200b20 6724:    dc.b 32,11," "
5b56 200c20 6725:    dc.b 32,12," "
5b5c 200d20 6726:    dc.b 32,13," "
5b62 200e20 6727:    dc.b 32,14," "
5b68 200f20 6728:    dc.b 32,15," "
5b6e 444953 6729:    dc.b "DISKIV  "
5b80 44534b 6730:    dc.b "DSKINV  "
5b92 43494f 6731:    dc.b "CIOV    "
5ba4 53494f 6732:    dc.b "SIOV    "
5bb6 534554 6733:    dc.b "SETVBV  "
5bc8 535953 6734:    dc.b "SYSVBV  "
5bda 584954 6735:    dc.b "XITVBV  "
5bec 53494f 6736:    dc.b "SIOINV  "
5bfe 53454e 6737:    dc.b "SENDEV  "
5c10 494e54 6738:    dc.b "INTINV  "
5c22 43494f 6739:    dc.b "CIOINV  "
5c34 534c46 6740:    dc.b "SLFTST  "
5c46 574152 6741:    dc.b "WARMSV  "
5c58 434f4c 6742:    dc.b "COLDSV  "
5c6a 52424c 6743:    dc.b "RBLOKV  "
5c7c 43534f 6744:    dc.b "CSOPIV  "
5c8e 564354 6745:    dc.b "VCTABL"
5c94      6746: ;
5c94      6747: ;-----
5c94      6748: ;
5c94      6749: p0.labels:
5c94      6750: ;
5c94 465230 6751:    dc.b "FR0  "
5c9a 465245 6752:    dc.b "FRE  "
5ca0 465231 6753:    dc.b "FR1  "
5ca6 465232 6754:    dc.b "FR2  "
5cac 465258 6755:    dc.b "FRX  "
5cb2 434958 6756:    dc.b "CIX  "
5cb8 494e42 6757:    dc.b "INBUFF"
5cbe 2b2020 6758:    dc.b "+  "
5cc4 464c50 6759:    dc.b "FLPTR "
5cca 2b2020 6760:    dc.b "+  "

```

```

5cd0 465054 6761:      dc.b "FPTR2 "
5cd6 2b2020 6762:      dc.b "+  "
5cdc      6763: ;-----
5cdc      6764: ;
5cdc      6765: p3.labels:
5cdc      6766: ;
5cdc 535550 6767:      dc.b "SUPERF"
5ce2 434b45 6768:      dc.b "CKEY  "
5ce8 434153 6769:      dc.b "CASSBT"
5cee 434152 6770:      dc.b "CARTCK"
5cf4 444552 6771:      dc.b "DERRF "
5cfa 41434d 6772:      dc.b "ACMVAR"
5d00 424153 6773:      dc.b "BASICF"
5d06 4d494e 6774:      dc.b "MINTLK"
5d0c 47494e 6775:      dc.b "GINTLK"
5d12 43484c 6776:      dc.b "CHLINK"
5d18 2b2020 6777:      dc.b "+  "
5d1e      6778: ;-----
5d1e      6779: ;
5d1e      6780: fp.labels:
5d1e      6781: ;
5d1e 434f4c 6782:      dc.b "COLPM3"
5d24 434f4c 6783:      dc.b "COLPF0"
5d2a 434f4c 6784:      dc.b "COLPF1"
5d30 434f4c 6785:      dc.b "COLPF2"
5d36 434f4c 6786:      dc.b "COLPF3"
5d3c 434f4c 6787:      dc.b "COLBK "
5d42 505249 6788:      dc.b "PRIOR "
5d48 564445 6789:      dc.b "VDELAY"
5d4e 475241 6790:      dc.b "GRACTL"
5d54 484954 6791:      dc.b "HITCLR"
5d5a 434f4e 6792:      dc.b "CONSOL"
5d60 414650 6793:      dc.b "AFP  "
5d66 464153 6794:      dc.b "FASC  "
5d6c 494650 6795:      dc.b "IFP  "
5d72 465049 6796:      dc.b "FPI  "
5d78 5a4652 6797:      dc.b "ZFR0  "
5d7e 5a4652 6798:      dc.b "ZFR1  "
5d84 465355 6799:      dc.b "FSUB  "
5d8a 464144 6800:      dc.b "FADD  "
5d90 464d55 6801:      dc.b "FMUL  "
5d96 464449 6802:      dc.b "FDIV  "
5d9c 504c59 6803:      dc.b "PLYEVL"
5da2 464c44 6804:      dc.b "FLD0R "
5da8 464c44 6805:      dc.b "FLD0P "
5dae 464c44 6806:      dc.b "FLD1R "
5db4 464c44 6807:      dc.b "FLD1P "
5dba 465354 6808:      dc.b "FST0R "
5dc0 465354 6809:      dc.b "FST0P "
5dc6 464d4f 6810:      dc.b "FMOVE "
5dcc 455850 6811:      dc.b "EXP  "
5dd2 455850 6812:      dc.b "EXP10 "
5dd8 4c4f47 6813:      dc.b "LOG  "

```

```

5dde 4c4f47 6814:      dc.b "LOG10 "
5de4      6815: ;
5de4      6816: ;-----
5de4      6817:
5de4      6818:
5de4      6819:
5de4      6820: ;      .OPT LIST
5de4      6821: ;      .PAGE "Data Tables"
5de4      6822: ;      .include tabl
5de4      6823: ;      ...  TABL  ...
5de4      6824: ;
5de4      6825: ;Table of ATARI label info
5de4      6826: ;in the following format:
5de4      6827: ;
5de4      6828: ; 1) Address of labels.
5de4      6829: ; 2) Memory Page #.
5de4      6830: ; 3) Last adr of page used.
5de4      6831: ; 4) Read/Write flag $80=ignore
5de4      6832: ;          1=read
5de4      6833: ;          0=write.
5de4      6834: ;-----
5de4      6835: ;
5de4      6836: page.table:
5de4      6837: ;
5de4 7a4c 6838:      dc.w page0
5de6 007f80 6839:      dc.b 0,$7f,$80
5de9 7a4f 6840:      dc.w page2
5deb 02ff80 6841:      dc.b 2,$ff,$80
5dee 7a55 6842:      dc.w page3
5df0 034f80 6843:      dc.b 3,$4f,$80
5df3 5a57 6844:      dc.w gtia_r
5df5 d01401 6845:      dc.b $d0,$14,1
5df8 d857 6846:      dc.w gtia_w
5dfa d01400 6847:      dc.b $d0,$14,0
5dfd 5658 6848:      dc.w pokey_r
5dff d20f01 6849:      dc.b $d2,$0f,1
5e02 b658 6850:      dc.w pokey_w
5e04 d20f00 6851:      dc.b $d2,$0f,0
5e07 1659 6852:      dc.w pia
5e09 d30380 6853:      dc.b $d3,3,$80
5e0c 2e59 6854:      dc.w antic
5e0e d40f80 6855:      dc.b $d4,$0f,$80
5e11 8e59 6856:      dc.w os
5e13 e47f80 6857:      dc.b $e4,$7f,$80
5e16      6858: ;
5e16 32 6859: tsize: dc.b tsize-page.table      ;*-page.tab
5e17      6860:
5e17      6861:
5e17      6862: ;-----
5e17      6863: ;
5e17      6864: ;Table of addresses added to
5e17      6865: ;pages 0 and 3 and the FP adrs.
5e17      6866: ;

```

```

5e17      6867: p0.adrlst:
5e17      6868: ;
5e17 d4dae0 6869:      dc.b $d4,$da,$e0,$e6
5e1b ecf2f3 6870:      dc.b $ec,$f2,$f3,$f4
5e1f fcfdfc 6871:      dc.b $fc,$fd,$fe,$ff
5e23      6872: ;
5e23      6873: ;
5e23      6874: p3.adrlst:
5e23      6875: ;
5e23 e8e9ea 6876:      dc.b $e8,$e9,$ea,$eb
5e27 ecedf8 6877:      dc.b $ec,$ed,$f8,$f9
5e2b fafbfc 6878:      dc.b $fa,$fb,$fc
5e2e      6879: ;
5e2e      6880: fp.adrlst:
5e2e      6881: ;
5e2e 15d016 6882:      dc.w $d015,$d016,$d017,$d018
5e36 19d01a 6883:      dc.w $d019,$d01a,$d01b,$d01c
5e3e 1dd01e 6884:      dc.w $d01d,$d01e,$d01f
5e44 00d8e6 6885:      dc.w $d800,$d8e6,$d9aa,$d9d2
5e4c 44da46 6886:      dc.w $da44,$da46,$da60,$da66
5e54 dbda28 6887:      dc.w $dadb,$db28,$dd40,$dd89
5e5c 8ddd98 6888:      dc.w $dd8d,$dd98,$dd9c,$dda7
5e64 abddb6 6889:      dc.w $ddab,$ddb6,$ddc0,$ddcc
5e6c cdded1 6890:      dc.w $decdd,$ded1
5e70      6891: ;-----
5e70      6892: ;
5e70      6893: ;      Mnemonic Table
5e70      6894: ;      -----
5e70      6895: ;
5e70      6896: mnemonic.tbl:
5e70      6897: ;
5e70 414443 6898:      dc.b "ADCANDASLBCC"
5e7c 424353 6899:      dc.b "BCSBEQBITBMI"
5e88 424e45 6900:      dc.b "BNEBPLBRKBVC"
5e94 425653 6901:      dc.b "BVSCCLCLDCLI"
5ea0 434c56 6902:      dc.b "CLVCMPCPXCPY"
5eac 444543 6903:      dc.b "DECDEXDEYEOR"
5eb8 494e43 6904:      dc.b "INCINXINYJMP"
5ec4 4a5352 6905:      dc.b "JSRLDALDXLDY"
5ed0 4c5352 6906:      dc.b "LSRNOPORAPHA"
5edc 504850 6907:      dc.b "PHPPLAPLPROL"
5ee8 524f52 6908:      dc.b "RORRTIRTSSBC"
5ef4 534543 6909:      dc.b "SECEDSEISTA"
5f00 535458 6910:      dc.b "STXSTYTXSTYA"
5f0c 545358 6911:      dc.b "TSXTXATXSTYA"
5f18      6912: ;
5f18      6913: ;-----
5f18      6914: ;      .PAGE "Mnemonic pointer table"
5f18      6915: ;Table used to index into the
5f18      6916: ;mnemonic table to get the
5f18      6917: ;desired assembler mnemonic.
5f18      6918: ;
5f18      6919: mnem.pointr:

```

```

5f18      6920: ;
5f18 006800 6921: dc.b $00,$68,$00,$00
5f1c 006808 6922: dc.b $00,$68,$08,$00
5f20 6e6808 6923: dc.b $6e,$68,$08,$00
5f24 006808 6924: dc.b $00,$68,$08,$00
5f28 1d6800 6925: dc.b $1d,$68,$00,$00
5f2c 006808 6926: dc.b $00,$68,$08,$00
5f30 296800 6927: dc.b $29,$68,$00,$00
5f34 006808 6928: dc.b $00,$68,$08,$00
5f38 560500 6929: dc.b $56,$05,$00,$00
5f3c 140577 6930: dc.b $14,$05,$77,$00
5f40 740577 6931: dc.b $74,$05,$77,$00
5f44 140577 6932: dc.b $14,$05,$77,$00
5f48 170500 6933: dc.b $17,$05,$00,$00
5f4c 000577 6934: dc.b $00,$05,$77,$00
5f50 860500 6935: dc.b $86,$05,$00,$00
5f54 000577 6936: dc.b $00,$05,$77,$00
5f58 7d4700 6937: dc.b $7d,$47,$00,$00
5f5c 004762 6938: dc.b $00,$47,$62,$00
5f60 6b4762 6939: dc.b $6b,$47,$62,$00
5f64 534762 6940: dc.b $53,$47,$62,$00
5f68 234700 6941: dc.b $23,$47,$00,$00
5f6c 004762 6942: dc.b $00,$47,$62,$00
5f70 2f4700 6943: dc.b $2f,$47,$00,$00
5f74 004762 6944: dc.b $00,$47,$62,$00
5f78 800200 6945: dc.b $80,$02,$00,$00
5f7c 00027a 6946: dc.b $00,$02,$7a,$00
5f80 71027a 6947: dc.b $71,$02,$7a,$00
5f84 53027a 6948: dc.b $53,$02,$7a,$00
5f88 260200 6949: dc.b $26,$02,$00,$00
5f8c 000200 6950: dc.b $00,$02,$00,$00
5f90 8c0200 6951: dc.b $8c,$02,$00,$00
5f94 000200 6952: dc.b $00,$02,$00,$00
5f98 008f00 6953: dc.b $00,$8f,$00,$00
5f9c 958f92 6954: dc.b $95,$8f,$92,$00
5fa0 4400a1 6955: dc.b $44,$00,$a1,$00
5fa4 958f92 6956: dc.b $95,$8f,$92,$00
5fa8 0b8f00 6957: dc.b $0b,$8f,$00,$00
5fac 958f92 6958: dc.b $95,$8f,$92,$00
5fb0 a78fa4 6959: dc.b $a7,$8f,$a4,$00
5fb4 008f00 6960: dc.b $00,$8f,$00,$00
5fb8 5f595c 6961: dc.b $5f,$59,$5c,$00
5fbc 5f595c 6962: dc.b $5f,$59,$5c,$00
5fc0 9b5998 6963: dc.b $9b,$59,$98,$00
5fc4 5f595c 6964: dc.b $5f,$59,$5c,$00
5fc8 0e5900 6965: dc.b $0e,$59,$00,$00
5fcc 5f595c 6966: dc.b $5f,$59,$5c,$00
5fd0 32599e 6967: dc.b $32,$59,$9e,$00
5fd4 5f595c 6968: dc.b $5f,$59,$5c,$00
5fd8 3b3500 6969: dc.b $3b,$35,$00,$00
5fdc 3b353e 6970: dc.b $3b,$35,$3e,$00
5fe0 503541 6971: dc.b $50,$35,$41,$00
5fe4 3b353e 6972: dc.b $3b,$35,$3e,$00

```

```

5fe8 1a3500 6973:      dc.b $1a,$35,$00,$00
5fec 00353e 6974:      dc.b $00,$35,$3e,$00
5ff0 2c3500 6975:      dc.b $2c,$35,$00,$00
5ff4 00353e 6976:      dc.b $00,$35,$3e,$00
5ff8 388300 6977:      dc.b $38,$83,$00,$00
5ffc 38834a 6978:      dc.b $38,$83,$4a,$00
6000 4d8365 6979:      dc.b $4d,$83,$65,$00
6004 38834a 6980:      dc.b $38,$83,$4a,$00
6008 118300 6981:      dc.b $11,$83,$00,$00
600c 00834a 6982:      dc.b $00,$83,$4a,$00
6010 898300 6983:      dc.b $89,$83,$00,$00
6014 00834a 6984:      dc.b $00,$83,$4a,$00
6018      6985: ;-----
6018      6986: ;      .PAGE "Operator Table"
6018      6987: ;Table used to figure out what
6018      6988: ;mnemonic and addressing mode
6018      6989: ;to use.
6018      6990: ;
6018 ff36ff 6991: op.table: dc.b $ff,$36,$ff,$ff
601c ff2eae 6992:      dc.b $ff,$2e,$ae,$ff
6020 002402 6993:      dc.b $00,$24,$02,$ff
6024 ff48c8 6994:      dc.b $ff,$48,$c8,$ff
6028 2638ff 6995:      dc.b $26,$38,$ff,$ff
602c ff30b0 6996:      dc.b $ff,$30,$b0,$ff
6030 004cff 6997:      dc.b $00,$4c,$ff,$ff
6034 ff4aca 6998:      dc.b $ff,$4a,$ca,$ff
6038 4836ff 6999:      dc.b $48,$36,$ff,$ff
603c 2e2eae 7000:      dc.b $2e,$2e,$ae,$ff
6040 002402 7001:      dc.b $00,$24,$02,$ff
6044 4848c8 7002:      dc.b $48,$48,$c8,$ff
6048 2638ff 7003:      dc.b $26,$38,$ff,$ff
604c ff30b0 7004:      dc.b $ff,$30,$b0,$ff
6050 004cff 7005:      dc.b $00,$4c,$ff,$ff
6054 ff4aca 7006:      dc.b $ff,$4a,$ca,$ff
6058 0036ff 7007:      dc.b $00,$36,$ff,$ff
605c ff2eae 7008:      dc.b $ff,$2e,$ae,$ff
6060 002402 7009:      dc.b $00,$24,$02,$ff
6064 4848c8 7010:      dc.b $48,$48,$c8,$ff
6068 2638ff 7011:      dc.b $26,$38,$ff,$ff
606c ff30b0 7012:      dc.b $ff,$30,$b0,$ff
6070 004cff 7013:      dc.b $00,$4c,$ff,$ff
6074 ff4aca 7014:      dc.b $ff,$4a,$ca,$ff
6078 0036ff 7015:      dc.b $00,$36,$ff,$ff
607c ff2eae 7016:      dc.b $ff,$2e,$ae,$ff
6080 002402 7017:      dc.b $00,$24,$02,$ff
6084 5448c8 7018:      dc.b $54,$48,$c8,$ff
6088 2638ff 7019:      dc.b $26,$38,$ff,$ff
608c ff30ff 7020:      dc.b $ff,$30,$ff,$ff
6090 004cff 7021:      dc.b $00,$4c,$ff,$ff
6094 ff4aff 7022:      dc.b $ff,$4a,$ff,$ff
6098 ffb6ff 7023:      dc.b $ff,$b6,$ff,$ff
609c aeaeae 7024:      dc.b $ae,$ae,$ae,$ff
60a0 00ff00 7025:      dc.b $00,$ff,$00,$ff

```

```

60a4 c8c8c8 7026:    dc.b $c8,$c8,$c8,$ff
60a8 26b8ff 7027:    dc.b $26,$b8,$ff,$ff
60ac b0b0b2 7028:    dc.b $b0,$b0,$b2,$ff
60b0 00cc00 7029:    dc.b $00,$cc,$00,$ff
60b4 ffcaff 7030:    dc.b $ff,$ca,$ff,$ff
60b8 243624 7031:    dc.b $24,$36,$24,$ff
60bc 2e2e2e 7032:    dc.b $2e,$2e,$2e,$ff
60c0 002400 7033:    dc.b $00,$24,$00,$ff
60c4 484848 7034:    dc.b $48,$48,$48,$ff
60c8 2638ff 7035:    dc.b $26,$38,$ff,$ff
60cc 303032 7036:    dc.b $30,$30,$32,$ff
60d0 004c00 7037:    dc.b $00,$4c,$00,$ff
60d4 4a4a4c 7038:    dc.b $4a,$4a,$4c,$ff
60d8 2436ff 7039:    dc.b $24,$36,$ff,$ff
60dc 2e2eae 7040:    dc.b $2e,$2e,$ae,$ff
60e0 002400 7041:    dc.b $00,$24,$00,$ff
60e4 4848c8 7042:    dc.b $48,$48,$c8,$ff
60e8 2638ff 7043:    dc.b $26,$38,$ff,$ff
60ec ff30b0 7044:    dc.b $ff,$30,$b0,$ff
60f0 004cff 7045:    dc.b $00,$4c,$ff,$ff
60f4 ff4aca 7046:    dc.b $ff,$4a,$ca,$ff
60f8 2436ff 7047:    dc.b $24,$36,$ff,$ff
60fc 2e2eae 7048:    dc.b $2e,$2e,$ae,$ff
6100 002400 7049:    dc.b $00,$24,$00,$ff
6104 4848c8 7050:    dc.b $48,$48,$c8,$ff
6108 2638ff 7051:    dc.b $26,$38,$ff,$ff
610c ff30b0 7052:    dc.b $ff,$30,$b0,$ff
6110 004cff 7053:    dc.b $00,$4c,$ff,$ff
6114 ff4aca 7054:    dc.b $ff,$4a,$ca,$ff
6118      7055: ;-----
6118      7056: ; .PAGE "Misc tables"
6118      7057: ;Table of bit masks
6118      7058: ;
6118 010204 7059: bit.table: dc.b 1,2,4,8
611c 102040 7060:    dc.b $10,$20,$40,$80
6120      7061: ;
6120      7062: ;-----
6120      7063: ;
6120      7064: ;Table of jump addresses to
6120      7065: ;add the correct addressing
6120      7066: ;mode to a mnemonic.
6120      7067: ;
6120 3f2b44 7068: mode.tabl: dc.w reg,acc
6124 4b2b6d 7069:    dc.w imm,branch
6128 7b2b81 7070:    dc.w adress,x_indx
612c 892b91 7071:    dc.w y_indx,zpage
6130 a02bb1 7072:    dc.w x_indx1,yindx1
6134 c22bd2 7073:    dc.w ind_x,ind_x1
6138 ed2b 7074:    dc.w ind_y
613a      7075: ;
613a      7076: ;-----
613a      7077: ;
613a      7078: optn_tbl:

```

```

613a      7079: ;
613a ffffff 7080:      dc.w dummy,dummy
613e fffd8 7081:      dc.w dummy,indrv
6142 e344ee 7082:      dc.w outdrv,asmdrv
6146 f9440f 7083:      dc.w flipscr,flipprt
614a 254552 7084:      dc.w fliplino,asmfile
614e 924967 7085:      dc.w load_labels,linebytes
6152 77458f 7086:      dc.w newfsize,zflip
6156 a745bf 7087:      dc.w mflip,neworg
615a e5450b 7088:      dc.w neweq,newbyte
615e 314654 7089:      dc.w newlink,newend
6162 af4a 7090:      dc.w saveprog
6164      7091: ;-----
6164      7092:
6164      7093:
6164      7094:
6164      7095:
6164      7096: ;      .PAGE "Data Stuff"
6164      7097: ; ... DATA ...
6164      7098: ;
6164 c441 7099: icsize: dc.w savend-beginsave
6166      7100: ;
6166 ffff 7101: header: dc.w dummy
6168 0021 7102:      dc.w beginsave
616a c362 7103:      dc.w savend-1
616c      7104: ;-----
616c      7105: ;
616c 000000 7106: colsize: dc.b 0,0,0,1
6170 010103 7107:      dc.b 1,1,3,3
6174 030305 7108:      dc.b 3,3,5,1
6178 050202 7109:      dc.b 5,2,2,10
617c 0a0a0a 7110:      dc.b 10,10,10,10
6180 0000 7111:      dc.b 0,0
6182      7112: ;
6182 303132 7113: hextbl: dc.b "01234567"
618a 383941 7114:      dc.b "89ABCDEF"
6192      7115: ;-----
6192      7116: ;
6192 01 7117: mymr: dc.b 1
6193 80 7118: kolor: dc.b $80
6194 ff 7119: lum: dc.b $ff
6195 00 7120: scr: dc.b 0
6196 00 7121: prt: dc.b 0
6197 3863 7122: line.start: dc.w lnum
6199 01 7123: xe: dc.b 1
619a 04 7124: bytes: dc.b 4
619b 01 7125: lino.flag: dc.b 1
619c 01 7126: asmflg: dc.b 1
619d 0080 7127: max.fsize: dc.w $8000
619f 4c9b9b 7128: marker: dc.b "L",CR,CR
61a2 5a9b9b 7129: zmarker: dc.b "Z",CR,CR
61a5 44313a 7130: filename: dc.b "D1:MEGADIS.COM",CR
61b4 303439 7131: frstno: dc.b "04995 "

```



```

61ba 303039 7132: secndno: dc.b "00999 "
61c0      7133: ;-----
61c0      7134: ;
61c0 4b3a9b 7135: k:      dc.b "K:",CR
61c3 503a9b 7136: p:      dc.b "P:",CR
61c6 453a9b 7137: e:      dc.b "E:",CR
61c9 44383a 7138: dn:      dc.b "D8:"
61cc 38     7139: d2:      dc.b "8"
61cd 38     7140: da:      dc.b "8"
61ce 44313a 7141: dir:      dc.b "D1:*.\"",CR
61d5 44313a 7142: l800:     dc.b "D1:LABEL.800",CR
61e2 44313a 7143: lxle:     dc.b "D1:LABEL.XLE",CR
61ef      7144: ;-----
61ef      7145: ;
61ef      7146: zeq_m:
61ef 3b2020 7147:      dc.b "; Zero Page Equates",CR
6207      7148: ;
6207      7149: sequ_m:
6207 3b2020 7150:      dc.b "; System Equates",CR
621f      7151: ;
621f      7152: xref_m:
621f 3b2045 7153:      dc.b "; External reference Equates",CR
623c      7154: ;
623c      7155: dash_m:
623c 3b2d2d 7156:      dc.b ";-----",CR
625a      7157: ;
625a 3b9b    7158: spacer_m: dc.b ";",CR
625c      7159: ;
625c 204279 7160: size_m:  dc.b " Bytes.",CR
6264      7161: ;-----
6264      7162: ;
6264 2e6f72 7163: org_m:   dc.b ".org",CR," "
6271      7164: ;
6271 3d209b 7165: eq_m:    dc.b "=",CR," "
627c      7166: ;
627c 2e4259 7167: byte_m:  dc.b ".BYTE ",CR," "
6287      7168: ;
6287 2e494e 7169: link_m:  dc.b ".INCLUDE #",CR
6292      7170: ;
6292 2e454e 7171: end_m:   dc.b ".END",CR," "
629d      7172: ;
629d 444154 7173: data_m:  dc.b "DATA ",CR
62a3      7174: ;
62a3 52454d 7175: rem_m:   dc.b "REM ",CR
62a8      7176: ;
62a8 50524f 7177: code_m:  dc.b "PROC MegaCode=*()",CR
62ba      7178: ;
62ba 4d4547 7179: mega_m:  dc.b "MEGABYTES",CR
62c4      7180: ;-----
62c4      7181: ;
62c4      7182: savend:
62c4 e002    7183:      dc.w $02e0
62c6 e102    7184:      dc.w $02e1

```

```
62c8 3521 7185:      dc.w init
62ca      7186: ;-----
62ca      7187: ;
62ca      7188: dvec:  ds.b 2
62cc      7189: oldvbi: ds.b 2
62ce      7190: oldmar: ds.b 1
62cf      7191: oldcolr: ds.b 5
62d4      7192: try:   ds.b 1
62d5      7193: loadflag: ds.b 1
62d6      7194: pagex:  ds.b 1
62d7      7195: sdm:    ds.b 1
62d8      7196: all.:   ds.b 1
62d9      7197: pcflag: ds.b 1
62da      7198: maxlen: ds.b 1
62db      7199: mainfiles: ds.b 1
62dc      7200: totfiles: ds.b 1
62dd      7201: dskopnflg: ds.b 1
62de      7202: cioy:   ds.b 1
62df      7203: zp.flag: ds.b 1
62e0      7204: address: ds.b 5
62e5      7205: both.flag: ds.b 1
62e6      7206: myflag:
62e6      7207: swapflag: ds.b 1
62e7      7208: hex.flag: ds.b 1
62e8      7209: binflag: ds.b 1
62e9      7210: outdot:  ds.b 1
62ea      7211: first:   ds.b 2
62ec      7212: labels:  ds.b 2
62ee      7213: good.label: ds.b 1
62ef      7214: which.bytek: ds.b 1
62f0      7215: drives:  ds.b 1
62f1      7216: pass:    ds.b 1
62f2      7217: more.:   ds.b 1
62f3      7218: xsave:   ds.b 1
62f4      7219: op.temp: ds.b 1
62f5      7220: freehi:  ds.b 2
62f7      7221: button:  ds.b 1
62f8      7222: st.flag: ds.b 1
62f9      7223: vbi.timer: ds.b 1
62fa      7224: seg.org: ds.b 2
62fc      7225: seg.end:
62fc      7226: seg.size: ds.b 2
62fe      7227: dskon:   ds.b 1
62ff      7228: files.size: ds.b 2
6301      7229: seg.adr: ds.b 2
6303      7230: left.to.do: ds.b 2
6305      7231: opcode:  ds.b 1
6306      7232: operand: ds.b 7
630d      7233: bytemax:
630d      7234: bytes.done: ds.b 1
630e      7235: bytecount:
630e      7236: lbuf.size: ds.b 2
6310      7237: labeltop: ds.b 2
```

```
6312      7238: totaldc.bs:
6312      7239: label.count: ds.b 2
6314      7240: bytes.to.do: ds.b 1
6315      7241: xtemp:   ds.b 1
6316      7242: page.loc: ds.b 2
6318      7243: oldlino:
6318      7244: page:   ds.b 1
6319      7245: oldifile:
6319      7246: page.top: ds.b 1
631a      7247: rw.flag: ds.b 1
631b      7248: is.p.opn.: ds.b 1
631c      7249: close.p.: ds.b 1
631d      7250: sector:  ds.b 2
631f      7251: left.in.buf: ds.b 2
6321      7252: line.ptr: ds.b 2
6323      7253: linesize: ds.b 2
6325      7254: mode:   ds.b 1
6326      7255: lastorg: ds.b 2
6328      7256: offset:  ds.b 1
6329      7257: currnt.sectr: ds.b 2
632b      7258: xlo:   ds.b 1
632c      7259: xhi:   ds.b 1
632d      7260: byts.this.op: ds.b 8
6335      7261: incend:  ds.b 1
6336      7262: to.lino:  ds.b 2
6338      7263: ;-----
6338      7264: ;
6338      7265: lnum:   ds.b 6
633e      7266: printline: ds.b 40
6366      7267: inline:  ds.b 40
638e      7268: infn:   ds.b 32
63ae      7269: outfn:  ds.b 32
63ce      7270: sector.map: ds.b 144
645e      7271: label.map: ds.b 512
665e      7272: drvbuf:  ds.b 256
675e      7273: label.buffer: ds.b 5632
7d5e      7274: bufadr:  ds.b 0      ;equ  *
7d5e      7275: ;-----
7d5e      7276:
7d5e      7277: win_end:  ds.b 0
7d5e      7278: .end
```

End assembly: no errors