

!A

LLOAD RANA.L,A\$4000

*** End of Pass 1

LLOAD RANA1.L,A\$4000

LLOAD RANA2.L,A\$4000

LLOAD RANA3.L,A\$4000

LLOAD RANA4.L,A\$4000

LLOAD RANA5.L,A\$4000

LLOAD RANA.L,A\$4000

*** End of Pass 2

```
0800      1      ttl "Rana ROM Code, RANA.L"
0800      2      src "RANA.L"
0800      3      ;
0800      4      ;
0800      5      ; RANA.L
0800      6      ;
0800      7      ;
0800      8      ; Rana ROM Code
0800      9      ;
0800     10      ; 2024 February 14
0800     11      ;
0800     12      ;
0800     13      ; DOS 4.5, Build 06
0800     14      ;
0800     15      ; 2024 February 14
0800     16      ;
0800     17      ;
0800     18      ; Start of Source Code: 0x4000
0800     19      ; Start of Symbol List: 0x7800
0800     20      ;
0800     21      ;
0800     22      ; Copyright (c) 2024 February 14 by
0800     23      ; Walland Philip Vrbancic Jr
0800     24      ;
0800     25      ; 6223 East Peabody Street
0800     26      ; Long Beach, California 90808
0800     27      ; Unitied States of America
0800     28      ;
0800     29      ; All Rights Reserved
0800     30      ;
0800     31      ; This software is the confidential and
0800     32      ; proprietary intellectual property of
0800     33      ; Walland Philip Vrbancic Jr
0800     34      ;
0800     35      ;
0000     36      LOC0      epz $00
0800     37      ;
0026     38      BUFRADRZ  epz $26
0026     39      TEMPZ     epz $26
0027     40      TEMP2Z    epz $27
002A     41      CURTRKZ   epz $2A
002B     42      SLOT16Z   epz $2B
002C     43      DRVFLAG   epz $2C
002C     44      ADRDATMK  epz $2C
002C     45      ADRFIELD  epz $2C
002D     46      SECFNDZ   epz $2D
002E     47      TRKFNDZ   epz $2E
002F     48      VOLFNDZ   epz $2F
0800     49      ;
0034     50      PHASE     epz $34
0035     51      SYNCNT    epz $35
003C     52      ROMTEMPZ  epz $3C
003C     53      MOTORTIM  epz $3C
003D     54      ROMSECTR  epz $3D
003E     55      BUFADR2Z  epz $3E
003E     56      ODDBITSZ  epz $3E
003F     57      SECTORZ   epz $3F
0800     58      ;
0040     59      ROMDATA   epz $40
0040     60      TRACKZ    epz $40
```

0041	61	ROMTRACK	epz	\$41
0041	62	VOLUMEZ	epz	\$41
004A	63	IOBADR	epz	\$4A
0800	64	;		
0800	65		enz	
0800	66	;		
0001	67	DEBUG	equ	1
0800	68	;		
0000	69	ZERO	equ	\$00
00FF	70	NEGONE	equ	\$FF
0800	71	;		
0006	72	RANAVRSN	equ	\$06
0006	73	RANABLD	equ	\$06
0800	74	;		
0045	75	DOS4VRSN	equ	\$45
0006	76	DOS4BLD	equ	\$06
0800	77	;		
0007	78	SLOTMASK	equ	\$07
000F	79	SECMASK	equ	\$0F
003F	80	TRKMASK	equ	\$3F
0800	81	;		
0003	82	MAXDRIVE	equ	3
0004	83	DFLTPHAS	equ	4
0010	84	PHASMAX	equ	16
0030	85	MAXTRACK	equ	48
0800	86	;		
0006	87	HDRSYNC	equ	6
0008	88	MINSYNC	equ	8
0020	89	MAXSYNC	equ	32
0020	90	MAXRETRY	equ	32
0500	91	SYNCBITS	equ	MAXSYNC*40
0800	92	;		
0010	93	HIGHSECS	equ	\$10
0020	94	MAXSEC	equ	\$20
0800	95	;		
0010	96	ENDTRK45	equ	\$10
001C	97	ENDSEC45	equ	\$1C
0800	98	;		
D8EF	99	MOTONTIM	equ	!-10000-1
0800	100	;		
0000	101	TBLTYPE	equ	\$00
0001	102	SNUM16	equ	\$01
0002	103	DNUM	equ	\$02
0003	104	VOLEXPT	equ	\$03
0004	105	TNUM	equ	\$04
0005	106	SNUM	equ	\$05
0008	107	USRBUF	equ	\$08
000A	108	IOCBPHAS	equ	\$0A
000B	109	BYTCNT	equ	\$0B
000C	110	CMDCODE	equ	\$0C
000D	111	ERRCODE	equ	\$0D
000E	112	VOLFND	equ	\$0E
000F	113	SLOTFND	equ	\$0F
0010	114	DRVFND	equ	\$10
0800	115	;		
0000	116	RWTSSEEK	equ	\$00
0001	117	RWTSREAD	equ	\$01
0002	118	RWTSWRIT	equ	\$02
0004	119	RWTSFRMT	equ	\$04
0800	120	;		
0000	121	RWNOERR	equ	\$00

```

0008      122  RWINITER equ $08
0010      123  RWPROTER equ $10
0020      124  RWVOLERR equ $20
0030      125  RWSYNERR equ $30
0040      126  RWDRVERR equ $40
0080      127  RWREADER equ $80
0800      128  ;
0056      129  NBUF2SIZ equ $56
00AA      130  ODDBITS  equ $AA
0800      131  ;
00D5      132  ADRMARK1 equ $D5
00AA      133  ADRMARK2 equ $AA
0096      134  ADRMARK3 equ $96
0800      135  ;
00D5      136  DATMARK1 equ $D5
00AA      137  DATMARK2 equ $AA
00AD      138  DATMARK3 equ $AD
0800      139  ;
00DE      140  SLPMARK1 equ $DE
00AA      141  SLPMARK2 equ $AA
00EB      142  SLPMARK3 equ $EB
0800      143  ;
00FF      144  SYNCMARK equ $FF
0800      145  ;
0100      146  PAGESIZE equ $100
0100      147  STACK    equ $100
0800      148  ;
0112      149  SAVYREG  equ $112
0113      150  SECTOR   equ $113
0114      151  NBUFLAG  equ $114
0115      152  DNUM0    equ $115
0800      153  ;
0116      154  ENDTRK   equ $116      ; from DOS 4.X INITVALS
0117      155  ENDSEC   equ $117      ; from DOS 4.X INITVALS
0118      156  STRTSEC  equ $118
0119      157  STOPSEC  equ $119
0800      158  ;
011A      159  HOOKCODE  equ $11A      ; 6 bytes
0120      160  SECMAP   equ $120      ; 32 bytes
0800      161  ;
0300      162  NBUF2BT  equ $300
0800      163  ;
03D0      164  DOSWARM  equ $3D0
03EA      165  HOOKDOS  equ $3EA
0800      166  ;
0478      167  FINDTRK  equ $478
04F8      168  RECALCNT equ $4F8
0800      169  ;
04FB      170  XMODE    equ $4FB
0800      171  ;
0578      172  SEEKCNT  equ $578
05F8      173  RETRYCNT equ $5F8
0800      174  ;
0678      175  NEXTON   equ $678
06F8      176  NEXTOFF  equ $6F8
0800      177  ;
0778      178  SLOT16   equ $778
07F8      179  MSLOT    equ $7F8      ; set MSB if use 0xC800 space
0800      180  ;
0800      181  ;
0800      182  ; DRV.TRK, DRV.PHAS variables are indexed by slot number.

```

```

0800      183      ;
0478      184      DRV0TRK      equ      $478      ; drive 1 track
04F8      185      DRV1TRK      equ      $4F8      ; drive 2 track
0578      186      DRV2TRK      equ      $578      ; drive 3 track
05F8      187      DRV3TRK      equ      $5F8      ; drive 4 track
0800      188      ;
0678      189      DRV0PHAS      equ      $678      ; drive 1 phase
06F8      190      DRV1PHAS      equ      $6F8      ; drive 2 phase
0778      191      DRV2PHAS      equ      $778      ; drive 3 phase
07F8      192      DRV3PHAS      equ      $7F8      ; drive 4 phase
0800      193      ;
0800      194      ;
08FE      195      BOOTADR      equ      $8FE
08FF      196      BOOTPGS      equ      $8FF
0800      197      ;
0800      198      PAGE08      equ      $0800
1000      199      PAGE10      equ      $1000
2000      200      PAGE20      equ      $2000
0800      201      ;
9D00      202      NBUF1L      equ      $9D00
9E00      203      NBUF2L      equ      $9E00
0800      204      ;
DE00      205      NBUF1H      equ      $DE00
DF00      206      NBUF2H      equ      $DF00
0800      207      ;
BFF0      208      BLDVRSN      equ      $BFF0
BFF1      209      BLDNMBR      equ      $BFF1
0800      210      ;
BFF2      211      MNGDISK      equ      $BFF2
BFFA      212      INITVAL      equ      $BFFA
BFFD      213      NBUF1PG      equ      $BFFD
0800      214      ;
C082      215      ROM2WP      equ      $C082
C08B      216      RAM1WE      equ      $C08B
0800      217      ;
C080      218      PHASEOFF      equ      $C080
C081      219      PHASEON      equ      $C081
C088      220      MOTOROFF      equ      $C088
C089      221      MOTORON      equ      $C089
C08A      222      DRV0EN      equ      $C08A
C08B      223      DRV1EN      equ      $C08B
C08C      224      STROBE      equ      $C08C
C08D      225      LATCH      equ      $C08D
C08E      226      DATAIN      equ      $C08E
C08F      227      DATAOUT      equ      $C08F
0800      228      ;
C000      229      SLOTROM0      equ      $C000
C800      230      SLOTROM8      equ      $C800
0800      231      ;
CFFF      232      CLRROM      equ      $CFFF
0800      233      ;
D003      234      DISKADRS      equ      $D003
0800      235      ;
FCA8      236      WAIT      equ      $FCA8
FF58      237      IORTS      equ      $FF58
0800      238      ;
0800      239      ;
0800      240      icl      "RANA1.L"

```

```

LLOAD RANA1.L,A$4000

```

```

0800          1          ttl "Rana ROM Code, RANA1.L"
0800          2          ;
0800          3          ;
0800          4          ; RANA1.L
0800          5          ;
0800          6          ;
0800          7          .if DEBUG
2000          8          org PAGE20
2000          9          obj PAGE20
2000         10          .el
2000         11          org SLOTROM8
2000         12          obj PAGE10
2000         13          .fi
2000         14          ;
2000         15          usr
2000         16          ;
2000         17          ;
2000         18          .if DEBUG
2000         19          ;
2000         20          ; Install Rana driver into DOS 4.5.
2000         21          ;
2000 2C 8B C0     22          bit RAM1WE
2003 2C 8B C0     23          bit RAM1WE
2006           24          ;
2006 A9 20       25          lda #RANARWTS
2008 8D 0D D0    26          sta DISKADRS+2*6-1
200B           27          ;
200B A9 28       28          lda /RANARWTS
200D 8D 0E D0    29          sta DISKADRS+1+2*6-1
2010           30          ;
2010 2C 82 C0    31          bit ROM2WP
2013           32          ;
2013 4C D0 03    33          jmp DOSWARM
2016           34          ;
2016           35          ;
2016           36          dfs PAGE SIZE-*&NEGONE,ZERO
2100           37          ;
2100           38          ;
2100           39          .fi
2100           40          ;
2100           41          ;
2100           42          ; Rana hardware selects drives 1 and 2 when ROMCODE is
2100           43          ; written or drives 3 and 4 when ROMCODE+1 is written.
2100           44          ;
2100 00          45          ROMCODE hex 00
2101 00          46          hex 00
2102           47          ;
2102           48          ;
2102           49          ; Extract the slot page from the return address on the
2102           50          ; stack. Fall into SELCHEAD for track 0. Protect Y-reg.
2102           51          ;
2102 BA         52          GETSLOT tsx
2103           53          ;
2103           54          .if DEBUG
2103 A9 C6       55          lda #$C6
2105 EA         56          nop
2106           57          .el
2106           58          lda STACK+2,X
2106           59          .fi
2106           60          ;

```

```

2106 8D F8 07      61          sta MSLOT
2109              62      ;
2109 29 07          63          and #SLOTMASK
210B              64      ;
210B 0A            65          asl
210C 0A            66          asl
210D 0A            67          asl
210E 0A            68          asl
210F              69      ;
210F 85 2B          70          sta SLOT16Z
2111              71      ;
2111              72      ;
2111              73      ; Always select the lower head.
2111              74      ;
2111 A6 2B          75 SELCHEAD ldx SLOT16Z
2113              76      ;
2113 BD 81 C0       77          lda PHASEON,X
2116 BD 85 C0       78          lda PHASEON+4,X
2119              79      ;
2119 20 C0 21       80          jsr WAIT52
211C              81      ;
211C BD 80 C0       82          lda PHASEOFF,X
211F BD 84 C0       83          lda PHASEOFF+4,X
2122              84      ;
2122 18             85          clc
2123              86      ;
2123 60             87          rts
2124              88      ;
2124              89      ;
2124              90      ; Routine to save the half-phase value of a drive in a
2124              91      ; slot dependant location.
2124              92      ;
2124              93      ; Do not modify the TRK, DRV, or PHAS table addresses.
2124              94      ;
2124              95      ; First calculate the half-phase value based on track
2124              96      ; and VALSPHAS. Add PHASE/2 if SECTOR > 0x0F.
2124              97      ;
2124 85 26           98 SAVETRK  sta TEMPZ
2126              99      ;
2126 A9 00          100         lda #ZERO
2128              101      ;
2128 AC 15 01       102         ldy DNUM0
212B C0 03         103         cpy #MAXDRIVE
212D 90 0B         104         bcc >1
212F              105      ;
212F AC 13 01       106         ldy SECTOR
2132 C0 10         107         cpy #HIGHSECS
2134 90 04         108         bcc >1
2136              109      ;
2136 A5 34         110         lda PHASE
2138 4A            111         lsr
2139              112      ;
2139 18            113         clc
213A              114      ;
213A A4 34         115 ^1      ldy PHASE
213C              116      ;
213C 65 26         117 ^2      adc TEMPZ
213E              118      ;
213E 88            119         dey
213F D0 FB         120         bne <2
2141              121      ;

```

```

2141 8D 78 04    122 SAVETRK2 sta FINDTRK
2144            123 ;
2144            124 ;
2144            125 ; Access the address based on slot and drive number.
2144            126 ;
2144 A2 04        127         ldx /DRV0TRK
2146 20 17 25    128         jsr DRVINDEX
2149            129 ;
2149 B1 26        130         lda (TEMPZ),Y
214B 85 2A        131         sta CURTRKZ
214D            132 ;
214D AD 78 04    133         lda FINDTRK
2150 91 26        134         sta (TEMPZ),Y
2152            135 ;
2152 60           136         rts
2153            137 ;
2153            138 ;
2153            139         dfs NBUF2SIZ-*)&NEGONE,ZERO
2156            140 ;
2156            141 ;
2156            142 ; Write translate table.
2156            143 ;
2156 96 97 9A     144 WRNIBL    hex 96979A9B9D9E9FA6
2159 9B 9D 9E
215C 9F A6
215E A7 AB AC    145         hex A7ABACADAEAFB2B3
2161 AD AE AF
2164 B2 B3
2166 B4 B5 B6    146         hex B4B5B6B7B9BABBBC
2169 B7 B9 BA
216C BB BC
216E BD BE BF    147         hex BDBEBFCBCDCECFD3
2171 CB CD CE
2174 CF D3
2176 D6 D7 D9    148         hex D6D7D9DADBDCDDDE
2179 DA DB DC
217C DD DE
217E DF E5 E6    149         hex DFE5E6E7E9EAEBEC
2181 E7 E9 EA
2184 EB EC
2186 ED EE EF    150         hex EDEEEFF2F3F4F5F6
2189 F2 F3 F4
218C F5 F6
218E F7 F9 FA    151         hex F7F9FAFBFCFDFF
2191 FB FC FD
2194 FE FF
2196            152 ;
2196            153 ;
2196            154 ; Read translate table.
2196            155 ;
2196            156 ; Allocate 9 bytes from this table for WAIT routines.
2196            157 ;
2196 00 01        158 RDNIBL    hex 0001
2198 00 00 02    159         hex 0000020300040506
219B 03 00 04
219E 05 06
21A0 00 00 00    160         hex 00000000000000708
21A3 00 00 00
21A6 07 08
21A8 00 00 00    161         hex 000000090A0B0C0D
21AB 09 0A 0B

```

```

21AE 0C 0D
21B0 00 00 0E    162          hex 00000E0F10111213
21B3 0F 10 11
21B6 12 13
21B8 00 14 15    163          hex 001415161718191A
21BB 16 17 18
21BE 19 1A
21C0          164      ;
21C0          165      ;          hex 0000000000000000
21C0          166      ;          hex 0000001B001C1D1E
21C0          167      ;
21C0 EA          168 WAIT52    nop
21C1 EA          169          nop
21C2 20 C5 21    170 WAIT48    jsr WAIT24
21C5 20 C8 21    171 WAIT24    jsr WAIT12
21C8 60          172 WAIT12    rts
21C9          173      ;
21C9 00 00 1B    174          hex 00001B001C1D1E
21CC 00 1C 1D
21CF 1E
21D0          175      ;
21D0 00 00 00    176          hex 0000001F00002021
21D3 1F 00 00
21D6 20 21
21D8 00 22 23    177          hex 0022232425262728
21DB 24 25 26
21DE 27 28
21E0 00 00 00    178          hex 0000000000292A2B
21E3 00 00 29
21E6 2A 2B
21E8 00 2C 2D    179          hex 002C2D2E2F303132
21EB 2E 2F 30
21EE 31 32
21F0 00 00 33    180          hex 0000333435363738
21F3 34 35 36
21F6 37 38
21F8 00 39 3A    181          hex 00393A3B3C3D3E3F
21FB 3B 3C 3D
21FE 3E 3F
2200          182      ;
2200          183      ;
2200          184      icl "RANA2.L"

```

```

LLOAD RANA2.L,A$4000

```

```

2200          1          ttl "Rana ROM Source Code, RANA2.L"
2200          2          ;
2200          3          ;
2200          4          ; RANA2.L
2200          5          ;
2200          6          ;
2200          7          ; These routines are all time critical.  Be carefull with
2200          8          ; page boundries and do not change any instructions.
2200          9          ;
2200         10          ; Prepare TEMPZ for later and write HDRSYNC sync bytes and
2200         11          ; data marks 0xD5, 0xAA, and 0xAD to disk.
2200         12          ;
2200 AC 00 DF 13 WRITBUFH ldy NBUF2H
2203 20 04 23 14          jsr WRITSYN0
2206         15          ;
2206         16          ;
2206         17          ; Checksum is cleared by writing the last byte of NBUF2.
2206         18          ; Apple originally published this routine using a 36 usec
2206         19          ; entrance.  I have changed this logic.
2206         20          ;
2206         21          ; Use NBUF1H and NBUF2H buffers.
2206         22          ;
2206 AE 55 DF 23          ldx NBUF2H+NBUF2SIZ-1
2209         24          ;
2209 A0 55     25          ldy #NBUF2SIZ-1
220B D0 07    26          bne >2          ; always taken
220D         27          ;
220D         28          ;
220D         29          ; Get prior 6-bit nibble and XOR with current nibble to
220D         30          ; form index into the write translate table.
220D         31          ;
220D B9 01 DF 32 ^1      lda NBUF2H+1,Y
2210         33          ;
2210 59 00 DF 34          eor NBUF2H,Y
2213 AA      35          tax
2214         36          ;
2214 BD 56 21 37 ^2      lda WRNIBL,X
2217         38          ;
2217 AE 78 07 39          ldx SLOT16
221A         40          ;
221A 9D 8D C0 41          sta LATCH,X
221D BD 8C C0 42          lda STROBE,X
2220         43          ;
2220 88       44          dey
2221 10 EA    45          bpl <1
2223         46          ;
2223         47          ;
2223         48          ; Write NBUF1H to disk using the same logic.
2223         49          ;
2223 C8       50          iny
2224         51          ;
2224 A5 26     52          lda TEMPZ
2226         53          ;
2226 59 00 DE 54 ^3      eor NBUF1H,Y
2229 AA      55          tax
222A         56          ;
222A BD 56 21 57          lda WRNIBL,X
222D         58          ;
222D AE 78 07 59          ldx SLOT16
2230         60          ;

```

```

2230 9D 8D C0      61          sta LATCH,X
2233 BD 8C C0      62          lda STROBE,X
2236              63          ;
2236 B9 00 DE      64          lda NBUF1H,Y
2239              65          ;
2239 C8            66          iny
223A D0 EA        67          bne <3
223C              68          ;
223C F0 4E        69          beq WRITRTN          ; always taken
223E              70          ;
223E              71          ;
223E              72          ; Write a sector data field routine.  A sector data field
223E              73          ; consists of pre-nibbled data contained in NBUF1H and
223E              74          ; NBUF2H.  Check write protect sense from disk controller.
223E              75          ;
223E 38           76          WRITSCTR sec
223F              77          ;
223F BD 8D C0      78          lda LATCH,X
2242 BD 8E C0      79          lda DATAIN,X
2245              80          ;
2245 30 68         81          bmi SETREAD
2247              82          ;
2247              83          ;
2247              84          ; Check NBUFLAG for routine and buffers for WRITSCTR.
2247              85          ;
2247 A9 AD         86          lda #DATMARK3
2249              87          ;
2249 2C 14 01      88          bit NBUFLAG
224C 30 B2         89          bmi WRITBUFH
224E              90          ;
224E              91          ;
224E              92          ; Prepare TEMPZ for later and write HDRSYNC sync bytes and
224E              93          ; data marks 0xD5, 0xAA, and 0xAD to disk.
224E              94          ;
224E AC 00 9E     95          ldy NBUF2L
2251 20 04 23     96          jsr WRITSYN0
2254              97          ;
2254              98          ;
2254              99          ; Checksum is cleared by writing the last byte of NBUF2.
2254             100          ; Apple originally published this routine using a 36 usec
2254             101          ; entrance.  I have changed this logic.
2254             102          ;
2254             103          ; Use NBUF1L and NBUF2L buffers.
2254             104          ;
2254 AE 55 9E     105          ldx NBUF2L+NBUF2SIZ-1
2257             106          ;
2257 A0 55       107          ldy #NBUF2SIZ-1
2259 D0 07      108          bne >2          ; always taken
225B           109          ;
225B           110          ;
225B           111          ; Get prior 6-bit nibble and XOR with current nibble to
225B           112          ; form index into the write translate table.
225B           113          ;
225B B9 01 9E   114          ^1      lda NBUF2L+1,Y
225E           115          ;
225E 59 00 9E   116          eor NBUF2L,Y
2261 AA        117          tax
2262           118          ;
2262 BD 56 21    119          ^2      lda WRNIBL,X
2265           120          ;
2265 AE 78 07    121          ldx SLOT16

```

```

2268      122 ;
2268 9D 8D C0 123      sta LATCH,X
226B BD 8C C0 124      lda STROBE,X
226E      125 ;
226E 88      126      dey
226F 10 EA    127      bpl <1
2271      128 ;
2271      129 ;
2271      130 ; Write NBUF1L to disk using the same logic.
2271      131 ;
2271 C8      132      iny
2272      133 ;
2272 A5 26    134      lda TEMPZ
2274      135 ;
2274 59 00 9D 136 ^3    eor NBUF1L,Y
2277 AA      137      tax
2278      138 ;
2278 BD 56 21 139      lda WRNIBL,X
227B      140 ;
227B AE 78 07 141      ldx SLOT16
227E      142 ;
227E 9D 8D C0 143      sta LATCH,X
2281 BD 8C C0 144      lda STROBE,X
2284      145 ;
2284 B9 00 9D 146      lda NBUF1L,Y
2287      147 ;
2287 C8      148      iny
2288 D0 EA    149      bne <3
228A      150 ;
228A 24 26    151      bit TEMPZ
228C      152 ;
228C      153 ;
228C      154 ; Write the checksum to disk.
228C      155 ;
228C A8      156 WRITRTN tay
228D      157 ;
228D B9 56 21 158      lda WRNIBL,Y
2290      159 ;
2290 20 42 23 160      jsr WNIBL
2293      161 ;
2293 24 26    162      bit TEMPZ
2295      163 ;
2295      164 ;
2295      165 ; Write slip marks 0xDE, 0xAA, and 0xEB to disk.
2295      166 ;
2295      167 ;
2295 EA      168 WRITEXIT nop
2296 EA      169      nop
2297      170 ;
2297 A9 DE    171      lda #SLPMARK1
2299 20 41 23 172      jsr WNIBL2
229C      173 ;
229C A9 AA    174      lda #SLPMARK2
229E 20 3F 23 175      jsr WNIBL9
22A1      176 ;
22A1 A9 EB    177      lda #SLPMARK3
22A3 20 3F 23 178      jsr WNIBL9
22A6      179 ;
22A6      180 ;
22A6      181 ; Terminate the address/data footer.
22A6      182 ;

```

```

22A6 A9 FF      183          lda #SYNCMARK
22A8 20 3F 23   184          jsr WNIBL9
22AB           185          ;
22AB 20 C5 21   186          jsr WAIT24          ; allow all bits to be written
22AE EA        187          nop
22AF           188          ;
22AF           189          ;
22AF           190          ; Return to read mode.
22AF           191          ;
22AF BD 8E C0   192 SETREAD  lda DATAIN,X
22B2 BD 8C C0   193          lda STROBE,X
22B5           194          ;
22B5 60         195          rts
22B6           196          ;
22B6           197          ;
22B6           198          ; Write an address data field routine. An address data
22B6           199          ; field consists of the volume, track, sector, and checksum
22B6           200          ; values. Check write protect sense from the Disk ][.
22B6           201          ;
22B6 38         202 WRITADR  sec
22B7           203          ;
22B7 A6 2B      204          ldx SLOT16Z
22B9           205          ;
22B9 BD 8D C0   206          lda LATCH,X
22BC BD 8E C0   207          lda DATAIN,X
22BF           208          ;
22BF 30 EE      209          bmi SETREAD
22C1           210          ;
22C1           211          ;
22C1           212          ; Prepare TEMPZ for later with the address header checksum
22C1           213          ; and write Y-reg number of sync bytes. Then write data
22C1           214          ; marks 0xD5, 0xAA, and 0x96 to disk.
22C1           215          ;
22C1 A5 41      216          lda VOLUMEZ
22C3 45 40      217          eor TRACKZ
22C5 45 3F      218          eor SECTORZ
22C7 85 26      219          sta TEMPZ
22C9           220          ;
22C9 A9 96      221          lda #ADRMARK3
22CB 20 08 23   222          jsr WRITSYNC
22CE           223          ;
22CE           224          ;
22CE           225          ; Write volume, track, sector, and checksum to disk.
22CE           226          ;
22CE A5 41      227          lda VOLUMEZ
22D0 20 30 23   228          jsr WBYTE
22D3           229          ;
22D3 A5 40      230          lda TRACKZ
22D5 20 30 23   231          jsr WBYTE
22D8           232          ;
22D8 A5 3F      233          lda SECTORZ
22DA 20 30 23   234          jsr WBYTE
22DD           235          ;
22DD A5 26      236          lda TEMPZ
22DF 20 30 23   237          jsr WBYTE
22E2           238          ;
22E2 90 B1      239          bcc WRITEXIT          ; always taken
22E4           240          ;
22E4           241          ;
22E4           242          ; Sector interleave remapping table to support 32 sectors
22E4           243          ; per track.

```

```

22E4          244 ;
22E4 00 0D 0B 245 INTRLEAV hex 000D0B0907050301
22E7 09 07 05
22EA 03 01
22EC 0E 0C 0A 246          hex 0E0C0A080604020F
22EF 08 06 04
22F2 02 0F
22F4 10 1D 1B 247          hex 101D1B1917151311
22F7 19 17 15
22FA 13 11
22FC 1E 1C 1A 248          hex 1E1C1A181614121F
22FF 18 16 14
2302 12 1F
2304          249 ;
2304          250 ;
2304          251 ; Y-reg contains the number of sync bytes to write.
2304          252 ;
2304          253 ; Configure firmware to generate sync bytes. Sync bytes
2304          254 ; are 40 usecs in length and are written as %1111111100.
2304          255 ; The first sync byte must be written exactly 40 usecs
2304          256 ; later and continually written in a 40 usec loop.
2304          257 ;
2304 84 26      258 WRITSYN0 sty TEMPZ
2306          259 ;
2306 A0 06      260          ldy #HDRSYNC
2308          261 ;
2308 85 2C      262 WRITSYNC sta ADRDATMK
230A          263 ;
230A A9 FF      264          lda #SYNCMARK
230C          265 ;
230C 9D 8F C0   266          sta DATAOUT,X
230F 1D 8C C0   267          ora STROBE,X
2312          268 ;
2312 48         269          pha
2313 68         270          pla
2314          271 ;
2314 20 C5 21    272 ^1      jsr WAIT24
2317          273 ;
2317 9D 8D C0    274          sta LATCH,X
231A 1D 8C C0    275          ora STROBE,X
231D          276 ;
231D EA         277          nop
231E          278 ;
231E 88         279          dey
231F D0 F3      280          bne <1
2321          281 ;
2321          282 ;
2321          283 ; Write address/data marks 0xD5, 0xAA, and TEMP2Z to disk.
2321          284 ;
2321 A9 D5      285          lda #DATMARK1          ; same as ADRMARK1
2323 20 3F 23   286          jsr WNIBL9
2326          287 ;
2326 A9 AA      288          lda #DATMARK2          ; same as ADRMARK2
2328 20 3F 23   289          jsr WNIBL9
232B          290 ;
232B A5 2C      291          lda ADRDATMK          ; recall 3rd address/data mark
232D          292 ;
232D 18         293          clc
232E 90 0F      294          bcc WNIBL9          ; always taken
2330          295 ;
2330          296 ;

```

```

2330          297 ; Write a byte as two four-bit nibbles to the disk
2330          298 ; starting with the odd bits.
2330          299 ;
2330 48        300 WBYTE      pha
2331          301 ;
2331 4A        302          lsr
2332 05 3E     303          ora ODDBITSZ
2334          304 ;
2334 9D 8D C0  305          sta LATCH,X
2337 BD 8C C0  306          lda STROBE,X
233A          307 ;
233A          308 ;
233A          309 ; Now write the even bits.
233A          310 ;
233A 68        311          pla
233B 05 3E     312          ora ODDBITSZ
233D          313 ;
233D 48        314          pha
233E 68        315          pla
233F          316 ;
233F          317 ;
233F          318 ; Wait 9 clock cycles, then write to disk.
233F          319 ;
233F 48        320 WNIBL9    pha
2340 68        321          pla
2341          322 ;
2341          323 ;
2341          324 ; Wait 2 clock cycles, then write to disk.
2341          325 ;
2341 18        326 WNIBL2    clc
2342          327 ;
2342          328 ;
2342          329 ; Write nibble to disk.
2342          330 ;
2342 9D 8D C0  331 WNIBL      sta LATCH,X
2345 BD 8C C0  332          lda STROBE,X
2348          333 ;
2348 60        334          rts
2349          335 ;
2349          336 ;
2349          337 ; READ routine reads nibblized data from the disk and
2349          338 ; stores the data in NBUF1L/NBUF2L or NBUF1H/NBUF2H. Fail
2349          339 ; after N read attempts where N corresponds to the
2349          340 ; possibility that there may be MAXSYNC sync bytes before
2349          341 ; the data header.
2349          342 ;
2349 A9 56      343 READSCTR lda #NBUF2SIZ
234B 85 26     344          sta TEMPZ
234D          345 ;
234D A2 AD     346          ldx #DATMARK3          ; for ADRDATMK
234F          347 ;
234F A0 28     348          ldy #SYNCBITS/32
2351 A9 00     349          lda #ZERO          ; for TEMP2Z
2353          350 ;
2353 20 01 24   351          jsr READMRKS
2356 30 55     352          bmi READERR
2358          353 ;
2358 2C 14 01   354          bit NBUFLAG
235B 30 24     355          bmi READBUFH
235D          356 ;
235D          357 ;

```

```

235D          358 ; Checksum has been initialized to zero.  Read nibbles into
235D          359 ; NBUF2L.
235D          360 ;
235D C6 26     361 ^6      dec TEMPZ
235F          362 ;
235F BC 8C C0  363 ^7      ldy STROBE,X
2362 10 FB     364         bpl <7
2364          365 ;
2364 59 00 21  366         eor RDNIBL-$96,Y
2367          367 ;
2367 A4 26     368         ldy TEMPZ
2369          369 ;
2369 99 00 9E   370         sta NBUF2L,Y
236C          371 ;
236C D0 EF     372         bne <6
236E          373 ;
236E          374 ;
236E          375 ; Now read nibbles into NBUF1L.
236E          376 ;
236E BC 8C C0  377 ^8      ldy STROBE,X
2371 10 FB     378         bpl <8
2373          379 ;
2373 59 00 21  380         eor RDNIBL-$96,Y
2376          381 ;
2376 A4 26     382         ldy TEMPZ
2378          383 ;
2378 99 00 9D   384         sta NBUF1L,Y
237B          385 ;
237B E6 26     386         inc TEMPZ
237D D0 EF     387         bne <8
237F          388 ;
237F F0 22     389         beq >9                ; always taken
2381          390 ;
2381          391 ;
2381          392 READBUFH:
2381          393 ;
2381          394 ; Checksum has been initialized to zero.  Read nibbles into
2381          395 ; NBUF2H.
2381          396 ;
2381 C6 26     397 ^6      dec TEMPZ
2383          398 ;
2383 BC 8C C0  399 ^7      ldy STROBE,X
2386 10 FB     400         bpl <7
2388          401 ;
2388 59 00 21  402         eor RDNIBL-$96,Y
238B          403 ;
238B A4 26     404         ldy TEMPZ
238D          405 ;
238D 99 00 DF   406         sta NBUF2H,Y
2390          407 ;
2390 D0 EF     408         bne <6
2392          409 ;
2392          410 ;
2392          411 ; Now read nibbles into NBUF1H.
2392          412 ;
2392 BC 8C C0  413 ^8      ldy STROBE,X
2395 10 FB     414         bpl <8
2397          415 ;
2397 59 00 21  416         eor RDNIBL-$96,Y
239A          417 ;
239A A4 26     418         ldy TEMPZ

```

```

239C          419 ;
239C 99 00 DE 420          sta NBUF1H,Y
239F          421 ;
239F E6 26   422          inc TEMPZ
23A1 D0 EF   423          bne <8
23A3          424 ;
23A3          425 ;
23A3          426 ; Verify the checksum byte.
23A3          427 ;
23A3 BC 8C C0 428 ^9          ldY STROBE,X
23A6 10 FB   429          bpl <9
23A8          430 ;
23A8 D9 00 21 431          cmp RDNIBL-$96,Y
23AB F0 28   432          beq READEXIT
23AD          433 ;
23AD 38      434 READERR sec
23AE          435 ;
23AE 60      436          rts
23AF          437 ;
23AF          438 ;
23AF          439 ; Read address field. Address field nibbles are odd/even
23AF          440 ; encoded. Read over 0x400 disk nibbles before giving up.
23AF          441 ;
23AF A2 96   442 READADR ldX #ADRMARK3          ; for ADRDATMK
23B1          443 ;
23B1 A0 04   444          ldY #4
23B3 98      445          tya          ; for TEMP2Z
23B4          446 ;
23B4 20 01 24 447          jsr READMRKS
23B7 30 F4   448          bmi READERR
23B9          449 ;
23B9          450 ;
23B9          451 ; Read the four-byte address data field nibbles. First
23B9          452 ; read the 'odd' bit nibble. The C-flag is set previously
23B9          453 ; from the ADRDATMK comparison in READMRKS.
23B9          454 ;
23B9 85 27   455 ^6          sta TEMP2Z
23BB          456 ;
23BB BD 8C C0 457 ^7          lda STROBE,X
23BE 10 FB   458          bpl <7
23C0          459 ;
23C0 2A      460          rol
23C1 85 26   461          sta TEMPZ
23C3          462 ;
23C3          463 ;
23C3          464 ; Now read the 'even' bit nibble and merge the two nibbles.
23C3          465 ; Store the data byte, then update the checksum and repeat
23C3          466 ; until the entire address field is read.
23C3          467 ;
23C3 BD 8C C0 468 ^8          lda STROBE,X
23C6 10 FB   469          bpl <8
23C8          470 ;
23C8 25 26   471          and TEMPZ
23CA          472 ;
23CA 99 2C 00 473          sta ADRFIELD,Y
23CD 45 27   474          eor TEMP2Z
23CF          475 ;
23CF 88      476          dey
23D0 10 E7   477          bpl <6
23D2          478 ;
23D2          479 ;

```

```
23D2          480 ; Checksum in A-reg must be zero for no error.
23D2          481 ;
23D2 A8       482          tay
23D3 D0 D8    483          bne READERR
23D5          484 ;
23D5          485 ;
23D5          486 READEXIT:
23D5          487 ;
23D5          488 ; Check for slip mark 1.
23D5          489 ;
23D5 BD 8C C0 490 ^1      lda STROBE,X
23D8 10 FB    491          bpl <1
23DA          492 ;
23DA C9 DE    493          cmp #SLPMARK1
23DC D0 CF    494          bne READERR
23DE          495 ;
23DE EA       496          nop
23DF          497 ;
23DF          498 ;
23DF          499 ; Check for skip mark 2. Slip mark 3 is not checked.
23DF          500 ;
23DF BD 8C C0 501 ^2      lda STROBE,X
23E2 10 FB    502          bpl <2
23E4          503 ;
23E4 C9 AA    504          cmp #SLPMARK2
23E6 D0 C5    505          bne READERR
23E8          506 ;
23E8 18       507          clc
23E9          508 ;
23E9 60       509          rts
23EA          510 ;
23EA          511 ;
23EA          512          icl "RANA3.L"
```

LLOAD RANA3.L,A\$4000

```

23EA          1          ttl "Rana ROM Source Code, RANA3.L"
23EA          2          ;
23EA          3          ;
23EA          4          ; RANA3.L
23EA          5          ;
23EA          6          ;
23EA          7          ; Connect or disconnect the Rana to DOS based on the value
23EA          8          ; in the Y-reg.  If Y-reg is zero, then connect to DOS.
23EA          9          ;
23EA 20 54 24   10  HOOKRANA jsr DISKMNG
23ED          11          ;
23ED A2 05      12          ldx #EXITLEN-1
23EF          13          ;
23EF BD FB 23   14  ^1      lda EXITCODE,X
23F2 9D 1A 01   15          sta HOOKCODE,X
23F5          16          ;
23F5 CA         17          dex
23F6 10 F7      18          bpl <1
23F8          19          ;
23F8 4C 1A 01   20          jmp HOOKCODE
23FB          21          ;
23FB          22          ;
23FB 2C FF CF   23  EXITCODE bit CLRROM
23FE          24          ;
23FE 4C EA 03   25          jmp HOOKDOS
2401          26          ;
0006          27  EXITLEN equ *-EXITCODE
2401          28          ;
2401          29          ;
2401 86 2C      30  READMRKS stx ADDRATMK          ; save address/data mark 3
2403 85 27      31          sta TEMP2Z          ; save MSB disk nibble counter
2405          32          ;
2405 A6 2B      33          ldx SLOT16Z          ; recall SLOT*16 value
2407          34          ;
2407 88         35  ^1      dey          ; LSB disk nibble counter
2408 D0 04      36          bne >2
240A          37          ;
240A C6 27      38          dec TEMP2Z          ; MSB disk nibble counter
240C 30 20      39          bmi >6
240E          40          ;
240E          41          ;
240E          42          ; Check for address/data mark 1.
240E          43          ;
240E BD 8C C0   44  ^2      lda STROBE,X          ; read sequencer latch
2411 10 FB      45          bpl <2
2413          46          ;
2413 C9 D5      47  ^3      cmp #ADRMARK1          ; same as DATMARK1
2415 D0 F0      48          bne <1
2417          49          ;
2417 EA         50          nop          ; waste 2 cycles
2418          51          ;
2418          52          ;
2418          53          ; Check for address/data mark 2.
2418          54          ;
2418 BD 8C C0   55  ^4      lda STROBE,X          ; read sequencer latch
241B 10 FB      56          bpl <4
241D          57          ;
241D C9 AA      58          cmp #ADRMARK2          ; same as DATMARK2
241F D0 F2      59          bne <3
2421          60          ;

```

```

2421 A0 03      61          ldy #3                ; needed for READADR
2423           62          ;
2423           63          ;
2423           64          ; Check for address/data mark 3 saved in ADRDATMK.
2423           65          ;
2423 BD 8C C0     66 ^5          lda STROBE,X          ; read sequencer latch
2426 10 FB      67          bpl <5
2428           68          ;
2428 C5 2C      69          cmp ADRDATMK
242A D0 E7      70          bne <3
242C           71          ;
242C           72          ;
242C           73          ; Initialize the checksum to zero.
242C           74          ;
242C A9 00      75          lda #ZERO
242E           76          ;
242E 60         77 ^6          rts
242F           78          ;
242F           79          ;
242F           80          ; Check progress of Boot Stage 1. If DOS 4.X RWTS is in
242F           81          ; memory, fall into DISKMNG with Y-reg set to zero. Enter
242F           82          ; with Y-reg set to zero.
242F           83          ;
242F AD FF 08   84 CHKDOS4X lda BOOTPGS
2432 10 58     85          bpl MOVRTN
2434           86          ;
2434           87          ;
2434           88          ; Set all the DRVnTRK and DRVnPHAS locations to zero.
2434           89          ;
2434 A5 2B      90          lda SLOT16Z
2436           91          ;
2436 4A         92          lsr
2437 4A         93          lsr
2438 4A         94          lsr
2439 4A         95          lsr
243A           96          ;
243A AA        97          tax
243B           98          ;
243B 98        99          tya
243C          100         ;
243C 9D 78 04   101         sta DRV0TRK,X
243F 9D F8 04   102         sta DRV1TRK,X
2442 9D 78 05   103         sta DRV2TRK,X
2445 9D F8 05   104         sta DRV3TRK,X
2448          105         ;
2448 9D 78 06   106         sta DRV0PHAS,X
244B 9D F8 06   107         sta DRV1PHAS,X
244E 9D 78 07   108         sta DRV2PHAS,X
2451 9D F8 07   109         sta DRV3PHAS,X
2454          110         ;
2454          111         ;
2454          112         ; Check for DOS 4.X initialization values.
2454          113         ;
2454 AD F0 BF   114 DISKMNG lda BLDVRSN
2457 C9 45     115         cmp #DOS4VRSN
2459 D0 31     116         bne MOVRTN
245B          117         ;
245B AD F1 BF   118         lda BLDNMBR
245E C9 06     119         cmp #DOS4BLD
2460 D0 2A     120         bne MOVRTN
2462          121         ;

```

```

2462          122 ;
2462          123 ; Recall Y-reg: zero to connect and not zero to disconnect
2462          124 ; Rana from the DOS 4.X Disk Address Table. C-flag is set.
2462          125 ;
2462 98        126          tya
2463 F0 01     127          beq >1
2465          128 ;
2465 18        129          clc
2466          130 ;
2466 A6 2B     131 ^1      ldx SLOT16Z
2468          132 ;
2468 A0 20     133          ldy #RANARWTS
246A          134 ;
246A          135          .if DEBUG
246A A9 28     136          lda /RANARWTS
246C EA       137          nop
246D          138          .el
246D          139          lda MSLOT
246D          140          .fi
246D          141 ;
246D 6C F2 BF 142          jmp (MNGDISK)
2470          143 ;
2470          144 ;
2470          145 ; Routines to move the Disk ][ head to the track in the
2470          146 ; A-reg on the current drive. Returns with zero in A-reg.
2470          147 ;
2470 A9 00     148 MOVHEAD0 lda #ZERO
2472 AA       149          tax
2473 F0 0B     150          beq MOVHEAD          ; always taken
2475          151 ;
2475 A0 05     152 MOVHEADN ldy #SNUM-TBLTYPE
2477          153 ;
2477 B1 4A     154          lda (IOBADR),Y
2479 AA       155          tax
247A          156 ;
247A A0 04     157          ldy #TNUM-TBLTYPE
247C          158 ;
247C B1 4A     159          lda (IOBADR),Y
247E 29 3F     160          and #TRKMASK
2480          161 ;
2480 8E 13 01  162 MOVHEAD stx SECTOR          ; for SAVETRK
2483          163 ;
2483 48        164          pha
2484          165 ;
2484 20 8D 24  166          jsr MOVEHD
2487          167 ;
2487 68        168          pla
2488 85 2A     169          sta CURTRKZ
248A          170 ;
248A A9 00     171          lda #ZERO
248C          172 ;
248C 60        173 MOVRTN rts
248D          174 ;
248D          175 ;
248D          176 ; Move the disk head to the requested track in half-phase
248D          177 ; steps. Set C-flag for drive #4.
248D          178 ;
248D 20 24 21 179 MOVEHD jsr SAVETRK
2490          180 ;
2490 C5 2A     181          cmp CURTRKZ
2492 F0 F8     182          beq MOVRTN

```

```

2494          183 ;
2494          184 ;
2494          185 ; Determine the offsets to the next motor phase based on
2494          186 ; the direction the head needs to move.
2494          187 ;
2494 A2 FF      188          ldx #NEGONE
2496 8A        189          txa
2497          190 ;
2497 B0 03      191          bcs >0
2499          192 ;
2499 E8         193          inx
249A A9 01     194          lda #1
249C          195 ;
249C 8E 78 06  196 ^0          stx NEXTON
249F 8D F8 06  197          sta NEXTOFF
24A2          198 ;
24A2          199 ;
24A2          200 ; Determine the direction to move and increment/decrement
24A2          201 ; the current position.
24A2          202 ;
24A2 A0 00     203          ldy #ZERO
24A4 84 26     204          sty TEMPZ
24A6          205 ;
24A6 20 EB 24  206 ^1          jsr CHKPOS
24A9          207 ;
24A9 38        208          sec
24AA          209 ;
24AA A5 2A     210          lda CURTRKZ
24AC ED 78 04  211          sbc FINDTRK
24AF F0 1B     212          beq >6
24B1          213 ;
24B1 B0 06     214          bcs >2
24B3          215 ;
24B3 49 FF     216          eor #NEGONE
24B5          217 ;
24B5 E6 2A     218          inc CURTRKZ
24B7 90 04     219          bcc >3                      ; always taken
24B9          220 ;
24B9 69 FE     221 ^2          adc #!-2
24BB          222 ;
24BB C6 2A     223          dec CURTRKZ
24BD          224 ;
24BD C5 26     225 ^3          cmp TEMPZ
24BF 90 02     226          bcc >4
24C1          227 ;
24C1 A5 26     228          lda TEMPZ
24C3          229 ;
24C3 C9 0C     230 ^4          cmp #OTBLLEN
24C5 B0 01     231          bcs >5
24C7          232 ;
24C7 A8        233          tay
24C8          234 ;
24C8 E6 26     235 ^5          inc TEMPZ
24CA D0 DA     236          bne <1                      ; always taken
24CC          237 ;
24CC          238 ;
24CC          239 ; At final destination. Lock in half-phase or phase
24CC          240 ; position.
24CC          241 ;
24CC 20 06 25  242 ^6          jsr MSWAIT
24CF          243 ;

```

```

24CF A5 2A      244      lda CURTRKZ
24D1           245      ;
24D1 29 06      246      and #6
24D3 05 2B      247      ora SLOT16Z
24D5           248      ;
24D5 AA         249      tax
24D6           250      ;
24D6 A5 2A      251      lda CURTRKZ
24D8 4A         252      lsr
24D9           253      ;
24D9 69 00      254      adc #ZERO
24DB 29 03      255      and #3
24DD           256      ;
24DD 0A         257      asl
24DE 05 2B      258      ora SLOT16Z
24E0           259      ;
24E0 A8         260      tay
24E1           261      ;
24E1 BD 80 C0    262      lda PHASEOFF,X
24E4 B9 80 C0    263      lda PHASEOFF,Y
24E7           264      ;
24E7 A9 80      265      lda #$80
24E9 D0 1B      266      bne MSWAIT          ; always taken
24EB           267      ;
24EB           268      ;
24EB           269      ; Select the next motor phase to turn off or turn on.
24EB           270      ; Fall into MSWAIT.
24EB           271      ;
24EB A5 2A      272      CHKPOS      lda CURTRKZ
24ED           273      ;
24ED 4A         274      lsr
24EE 90 06      275      bcc >7
24F0           276      ;
24F0 ED 78 06    277      sbc NEXTON
24F3           278      ;
24F3 38         279      sec
24F4 B0 04      280      bcs >8
24F6           281      ;
24F6 6D F8 06    282      ^7      adc NEXTOFF
24F9           283      ;
24F9 18         284      clc
24FA           285      ;
24FA 2A         286      ^8      rol
24FB           287      ;
24FB 29 07      288      and #7
24FD 05 2B      289      ora SLOT16Z
24FF           290      ;
24FF AA         291      tax
2500           292      ;
2500 BD 80 C0    293      lda PHASEOFF,X
2503           294      ;
2503 B9 2B 25    295      lda ONOFFTBL,Y
2506           296      ;
2506           297      ;
2506           298      ; Routine to delay A-reg * 99 + 13 usecs. This routine
2506           299      ; must reside on the same page.
2506           300      ;
2506 38         301      MSWAIT      sec
2507           302      ;
2507 A2 11      303      ^1      ldx #17
2509           304      ;

```

```

2509 CA          305 ^2      dex
250A D0 FD      306      bne <2
250C           307      ;
250C E6 3C      308      inc MOTORTIM
250E D0 02      309      bne >3
2510           310      ;
2510 E6 3D      311      inc MOTORTIM+1
2512           312      ;
2512 E9 01      313 ^3      sbc #1
2514 D0 F1      314      bne <1
2516           315      ;
2516 60         316      rts
2517           317      ;
2517           318      ;
2517           319      ; Calculate the slot-indexed address for the current track
2517           320      ; or phase value based on slot and drive number. Enter
2517           321      ; with X-reg containing the /DRV0TRK or /DRV0PHAS value.
2517           322      ;
2517 A9 78      323 DRVINDEX lda #DRV0TRK      ; same for #DRV0PHAS
2519 85 26      324      sta TEMPZ
251B           325      ;
251B A5 2B      326      lda SLOT16Z
251D           327      ;
251D 4A         328      lsr
251E 4A         329      lsr
251F           330      ;
251F 0D 15 01   331      ora DNUM0
2522           332      ;
2522 4A         333      lsr
2523 6A         334      ror
2524           335      ;
2524 A8         336      tay
2525           337      ;
2525 90 01      338      bcc >1
2527           339      ;
2527 E8         340      inx
2528           341      ;
2528 86 27      342 ^1      stx TEMPZ+1
252A           343      ;
252A 60         344      rts
252B           345      ;
252B           346      ;
252B           347      ; PHASEON/PHASEOFF table. Time delay uses MSWAIT.
252B           348      ;
252B 40 32 2A   349 ONOFFTBL hex 40322A242120
252E 24 21 20
2531 1F 1E 1E   350      hex 1F1E1E1D1D1C
2534 1D 1D 1C
2537           351      ;
000C           352 OTBLLEN equ *-ONOFFTBL
2537           353      ;
2537           354      ;
2537           355      icl "RANA4.L"

```

LLOAD RANA4.L,A\$4000

```

2537          1          ttl "Rana ROM Code, RANA4.L"
2537          2          ;
2537          3          ;
2537          4          ; RANA4.L
2537          5          ;
2537          6          ;
2537          7          ; RWTS handler routine for Disk ][ or Rana.
2537          8          ;
2537          9          ; This Rana firmware reads sectors 0x00-0x0F every PHASE
2537         10          ; and sectors 0x10-0x1F the next PHASE/2 of that track.
2537         11          ;
2537         12          ; Get slot number and confirm NBUF1PG value.
2537         13          ;
2537 8E 78 07     14  RWTSENT stx SLOT16
253A          15          ;
253A 20 AF 22    16          jsr SETREAD          ; ensure read mode is enabled
253D          17          ;
253D A9 EF      18          lda #MOTONTIM
253F 85 3C      19          sta MOTORTIM
2541          20          ;
2541 A9 D8      21          lda /MOTONTIM
2543 85 3D      22          sta MOTORTIM+1
2545          23          ;
2545 A0 00      24          ldy #ZERO
2547 84 2C      25          sty DRVFLAG
2549          26          ;
2549 AD FD BF    27          lda NBUF1PG
254C C9 9D      28          cmp /NBUF1L
254E F0 05      29          beq >0
2550          30          ;
2550 C9 DE      31          cmp /NBUF1H
2552 D0 26      32          bne SYNERR
2554          33          ;
2554 88         34          dey
2555          35          ;
2555 8C 14 01    36          ^0 sty NBUFLAG
2558          37          ;
2558          38          ;
2558          39          ; Check whether data is changing on this controller card
2558          40          ; even though the drive motor is currently off but still
2558          41          ; possibly spinning.
2558          42          ;
2558 A0 08      43          ldy #8
255A          44          ;
255A BD 8C C0    45          ^1 lda STROBE,X
255D          46          ;
255D 20 C5 21    47          jsr WAIT24
2560          48          ;
2560 DD 8C C0    49          cmp STROBE,X
2563 D0 05      50          bne >2
2565          51          ;
2565 88         52          dey
2566 D0 F2      53          bne <1
2568          54          ;
2568 E6 2C      55          inc DRVFLAG          ; data not changing
256A          56          ;
256A          57          ;
256A          58          ; Start the motor, then adjust the drive to 0:3 and select
256A          59          ; the requested drive.
256A          60          ;

```

```

256A BD 89 C0      61 ^2      lda MOTORON,X
256D              62 ;
256D A0 02        63      ldy #DNUM-TBLTYPE
256F              64 ;
256F B1 4A        65      lda (IOBADR),Y
2571 48           66      pha
2572             67 ;
2572 38           68      sec
2573             69 ;
2573 E9 01        70      sbc #1          ; adjust to 0:3
2575             71 ;
2575 C9 04        72      cmp #4
2577 90 06        73      bcc >0
2579             74 ;
2579 68           75      pla
257A             76 ;
257A A9 30        77 SYNERR  lda #RWSYNERR
257C             78 ;
257C 4C E3 26     79      jmp ERREXIT
257F             80 ;
257F             81 ;
257F             82 ; Select drives 1:2 or 3:4.  Save adjusted drive number.
257F             83 ;
257F 8D 15 01     84 ^0      sta DNUM0
2582             85 ;
2582 4A           86      lsr
2583             87 ;
2583 A8           88      tay
2584             89 ;
2584 99 00 21     90      sta ROMCODE,Y
2587             91 ;
2587 90 01        92      bcc >3
2589             93 ;
2589 E8           94      inx
258A             95 ;
258A BD 8A C0     96 ^3      lda DRV0EN,X
258D             97 ;
258D             98 ;
258D             99 ; If the drive number has changed save the new drive
258D            100 ; number in the IOB and wait 180 msec for the old drive
258D            101 ; to come to rest.
258D            102 ;
258D A0 10        103      ldy #DRVFND-TBLTYPE
258F            104 ;
258F 68           105      pla
2590 D1 4A        106      cmp (IOBADR),Y
2592 F0 0C        107      beq >5
2594             108 ;
2594 91 4A        109      sta (IOBADR),Y
2596             110 ;
2596 A0 08        111      ldy #8
2598             112 ;
2598 20 06 25     113 ^4      jsr MSWAIT
259B             114 ;
259B 88           115      dey
259C D0 FA        116      bne <4
259E             117 ;
259E E6 2C        118      inc DRVFLAG          ; drive changed
25A0             119 ;
25A0             120 ;
25A0             121 ; If there is no data changing on this controller card or

```

```

25A0      122 ; the drive number has changed, wait for this drive motor
25A0      123 ; to come up to speed.
25A0      124 ;
25A0 A5 2C 125 ^5      lda DRVFLAG
25A2 F0 07 126          beq >7
25A4      127 ;
25A4 20 06 25 128 ^6      jsr MSWAIT
25A7      129 ;
25A7 24 3D 130          bit MOTORTIM+1
25A9 30 F9 131          bmi <6
25AB      132 ;
25AB      133 ;
25AB      134 ; Always select the lower disk head.
25AB      135 ;
25AB 20 11 21 136 ^7      jsr SELCHEAD
25AE      137 ;
25AE      138 ;
25AE      139 ; Initialize PHASE from the selected saved PHASE location
25AE      140 ; if it is not zero. If it is zero, initialize the saved
25AE      141 ; PHASE location with #DFLTPHAS.
25AE      142 ;
25AE A2 06 143          ldx /DRVOPHAS
25B0 20 17 25 144          jsr DRVINDEX
25B3      145 ;
25B3 8C 12 01 146          sty SAVYREG
25B6      147 ;
25B6 B1 26 148          lda (TEMPZ),Y
25B8 D0 04 149          bne >8
25BA      150 ;
25BA A9 04 151          lda #DFLTPHAS
25BC 91 26 152          sta (TEMPZ),Y
25BE      153 ;
25BE 85 34 154 ^8      sta PHASE
25C0      155 ;
25C0      156 ;
25C0      157 ; Obtain the requested IOCB PHASE value. If it is zero,
25C0      158 ; use PHASE. Range check the value in A-reg.
25C0      159 ;
25C0 A0 0A 160          ldy #IOCBPHAS-TBLTYPE
25C2      161 ;
25C2 B1 4A 162          lda (IOBADR),Y
25C4 D0 02 163          bne >9
25C6      164 ;
25C6 A5 34 165          lda PHASE
25C8      166 ;
25C8 C9 11 167 ^9      cmp #PHASMAX+1
25CA B0 AE 168          bcs SYNERR
25CC      169 ;
25CC      170 ;
25CC      171 ; Compare the requested PHASE value to the saved PHASE
25CC      172 ; value in PHASE. If they differ move the disk head to
25CC      173 ; track 0 using the value already in PHASE. Then set PHASE
25CC      174 ; to the requested PHASE value.
25CC      175 ;
25CC 91 4A 176          sta (IOBADR),Y
25CE      177 ;
25CE C5 34 178          cmp PHASE
25D0 F0 0C 179          beq >1
25D2      180 ;
25D2 AC 12 01 181          ldy SAVYREG
25D5      182 ;

```

```

25D5 91 26      183      sta (TEMPZ),Y
25D7 48         184      pha
25D8           185      ;
25D8 20 70 24   186      jsr MOVHEAD0
25DB           187      ;
25DB 68         188      pla
25DC 85 34      189      sta PHASE
25DE           190      ;
25DE           191      ;
25DE           192      ; Position the disk head over the requested track using
25DE           193      ; the PHASE value, verify the requested command, and
25DE           194      ; process it.
25DE           195      ;
25DE 20 75 24   196      ^1      jsr MOVHEADN
25E1           197      ;
25E1 A0 0C      198      ldy #CMDCODE-TBLTYPE
25E3           199      ;
25E3 B1 4A      200      lda (IOBADR),Y
25E5 F0 0F      201      beq >2          ; RWTSSEEK
25E7           202      ;
25E7 C9 01      203      cmp #RWTSREAD
25E9 F0 59      204      beq >7
25EB           205      ;
25EB C9 02      206      cmp #RWTSWRIT
25ED F0 0A      207      beq >3
25EF           208      ;
25EF C9 04      209      cmp #RWTSFRMT
25F1 D0 03      210      bne >2
25F3           211      ;
25F3 4C ED 26   212      jmp DISKFMT
25F6           213      ;
25F6 4C E0 26   214      ^2      jmp RWTSEXIT
25F9           215      ;
25F9           216      ;
25F9           217      ; A write command, so prenibblize the data first.
25F9           218      ;
25F9           219      ; The prenibblize routine converts 256 bytes pointed at by
25F9           220      ; BUFADR2Z to 342 6-bit nibbles of the form 00XXXXXX.
25F9           221      ;
25F9           222      ; First clear NBUF2.
25F9           223      ;
25F9 A2 55      224      ^3      ldx #NBUF2SIZ-1
25FB           225      ;
25FB A9 00      226      lda #ZERO
25FD           227      ;
25FD 2C 14 01   228      bit NBUFLAG
2600 30 22      229      bmi PRENIBLH
2602           230      ;
2602           231      ;
2602           232      ; Clear NBUF2L.
2602           233      ;
2602 9D 00 9E    234      ^4      sta NBUF2L,X
2605           235      ;
2605 CA         236      dex
2606 10 FA      237      bpl <4
2608           238      ;
2608           239      ;
2608           240      ; Process BUFADR2Z into NBUF1 and NBUF2.
2608           241      ;
2608 A0 02      242      ldy #2
260A           243      ;

```

```

260A A2 00      244 ^5      ldx #ZERO
260C           245 ;
260C 88        246 ^6      dey
260D           247 ;
260D B1 3E     248          lda (BUFADR2Z),Y
260F           249 ;
260F           250 ;
260F           251 ; Shift low order two bits into NBUF2L.
260F           252 ;
260F 4A        253          lsr
2610 3E 00 9E  254          rol NBUF2L,X
2613           255 ;
2613 4A        256          lsr
2614 3E 00 9E  257          rol NBUF2L,X
2617           258 ;
2617           259 ;
2617           260 ; Put low order six bits into NBUF1L.
2617           261 ;
2617 99 00 9D  262          sta NBUF1L,Y
261A           263 ;
261A E8        264          inx
261B           265 ;
261B E0 56     266          cpx #NBUF2SIZ
261D D0 ED     267          bne <6
261F           268 ;
261F 98        269          tya
2620 D0 E8     270          bne <5
2622           271 ;
2622 F0 20     272          beq >7                ; always taken
2624           273 ;
2624           274 ;
2624           275 ; Clear NBUF2H.
2624           276 ;
2624           277 PRENIBLH:
2624 9D 00 DF  278 ^4      sta NBUF2H,X
2627           279 ;
2627 CA        280          dex
2628 10 FA     281          bpl <4
262A           282 ;
262A           283 ;
262A           284 ; Process BUFADR2Z into NBUF1 and NBUF2.
262A           285 ;
262A A0 02     286          ldY #2
262C           287 ;
262C A2 00     288 ^5      ldx #ZERO
262E           289 ;
262E 88        290 ^6      dey
262F           291 ;
262F B1 3E     292          lda (BUFADR2Z),Y
2631           293 ;
2631           294 ;
2631           295 ; Shift low order two bits into NBUF2H.
2631           296 ;
2631 4A        297          lsr
2632 3E 00 DF  298          rol NBUF2H,X
2635           299 ;
2635 4A        300          lsr
2636 3E 00 DF  301          rol NBUF2H,X
2639           302 ;
2639           303 ;
2639           304 ; Put low order six bits into NBUF1H.

```

```

2639          305 ;
2639 99 00 DE 306      sta NBUF1H,Y
263C          307 ;
263C E8      308      inx
263D          309 ;
263D E0 56   310      cpx #NBUF2SIZ
263F D0 ED   311      bne <6
2641          312 ;
2641 98      313      tya
2642 D0 E8   314      bne <5
2644          315 ;
2644          316 ;
2644          317 ; Set up for one disk head recalibrate, 2 track seeks,
2644          318 ; and 32 sector retries.
2644          319 ;
2644 A9 02    320 ^7      lda #2
2646 8D F8 04 321      sta RECALCNT
2649          322 ;
2649 A9 02    323 ^1      lda #2
264B 8D 78 05 324      sta SEEKCNT
264E          325 ;
264E A9 20    326      lda #32
2650 8D F8 05 327      sta RETRYCNT
2653          328 ;
2653          329 ;
2653          330 ; The time it takes WRITADR to write SLPMARK3 and a
2653          331 ; SYNCMARK and call WRITSCTR is 80 usecs. The time it
2653          332 ; takes READADR to return from reading SLPMARK2 and
2653          333 ; check CMDCODE for RWTSREAD before calling WRITSCTR is
2653          334 ; 72 usecs. Both READSCTR and WRITSCTR have enough time.
2653          335 ;
2653 20 AF 23  336 ^2      jsr READADR
2656 90 19    337      bcc >5
2658          338 ;
2658 CE F8 05 339 ^3      dec RETRYCNT
265B D0 F6    340      bne <2
265D          341 ;
265D          342 ;
265D          343 ; Recalibrate the disk head. Set the disk head track as
265D          344 ; if it was on track 48. Move the disk head to track 0,
265D          345 ; then to requested track.
265D          346 ;
265D A9 40    347 ^4      lda #RWDRVERR          ; get bad drive error
265F          348 ;
265F CE F8 04 349      dec RECALCNT
2662 F0 41    350      beq RWTSERR
2664          351 ;
2664 A9 C0     352      lda #DFLTPHAS*MAXTRACK ; set track to 48
2666 20 41 21 353      jsr SAVETRK2
2669          354 ;
2669 20 70 24 355      jsr MOVHEAD0
266C          356 ;
266C 20 75 24 357      jsr MOVHEADN
266F F0 D8    358      beq <1          ; always taken
2671          359 ;
2671          360 ;
2671          361 ; Save the found volume number in the IOB. Check address
2671          362 ; field for the requested track.
2671          363 ;
2671 A0 0E     364 ^5      ldy #VOLFND-TBLTYPE
2673          365 ;

```

```

2673 A5 2F      366      lda VOLFNDZ
2675 91 4A      367      sta (IOBADR),Y
2677           368      ;
2677 A5 2E      369      lda TRKFNDZ
2679 C5 2A      370      cmp CURTRKZ
267B F0 0D      371      beq >6
267D           372      ;
267D CE 78 05   373      dec SEEKCNT
2680 F0 DB      374      beq <4
2682           375      ;
2682           376      ;
2682           377      ; Set the disk head to the track found, then to requested
2682           378      ; track.
2682           379      ;
2682 20 24 21    380      jsr SAVETRK
2685           381      ;
2685 20 75 24    382      jsr MOVHEADN
2688 F0 C9      383      beq <2                ; always taken
268A           384      ;
268A           385      ;
268A           386      ; Check for correct sector from the interleave table.
268A           387      ;
268A A0 05      388      ^6      ldy #SNUM-TBLTYPE
268C           389      ;
268C B1 4A      390      lda (IOBADR),Y
268E A8         391      tay
268F           392      ;
268F B9 E4 22   393      lda INTRLEAV,Y
2692 C5 2D      394      cmp SECFNDZ
2694 D0 BD      395      bne <2
2696           396      ;
2696           397      ;
2696           398      ; Now at requested sector for read or write operation.
2696           399      ;
2696 A0 0C      400      ldy #CMDCODE-TBLTYPE
2698           401      ;
2698 B1 4A      402      lda (IOBADR),Y
269A C9 01      403      cmp #RWTSREAD
269C F0 0A      404      beq >1
269E           405      ;
269E           406      ;
269E           407      ; Write the nibblized data to the requested sector.
269E           408      ; If C-flag returns set then write protect sense.
269E           409      ;
269E 20 3E 22   410      jsr WRITSCTR
26A1 90 3D      411      bcc RWTSEXIT
26A3           412      ;
26A3 A9 10      413      RWPERR  lda #RWPROTER        ; write protect error
26A5           414      ;
26A5 38         415      RWTSEERR sec
26A6 B0 3B      416      bcs ERREXIT                ; always taken
26A8           417      ;
26A8           418      ;
26A8           419      ; Read the requested sector and convert the disk nibbles to
26A8           420      ; bytes. The post-nibblize routine converts 342 nibbles of
26A8           421      ; the form 00XXXXXX to eight bit data bytes. The nibbles
26A8           422      ; are stored in NBUF1 and NBUF2, and the 8-bit bytes are
26A8           423      ; stored at BUFADR2Z.
26A8           424      ;
26A8 20 49 23   425      ^1      jsr READSCTR
26AB B0 AB      426      bcs <3

```

```

26AD          427 ;
26AD A0 00    428      ldy #ZERO
26AF          429 ;
26AF 2C 14 01 430      bit NBUFLAG
26B2 30 17    431      bmi POSTNIBH
26B4          432 ;
26B4          433 ;
26B4          434 ; Convert the NBUF1L and NBUF2L nibbles to bytes.
26B4          435 ;
26B4 A2 56    436 ^2      ldx #NBUF2SIZ
26B6          437 ;
26B6 CA       438 ^3      dex
26B7 30 FB    439      bmi <2
26B9          440 ;
26B9          441 ;
26B9          442 ; Get byte and shift in low order two bits from NBUF2L.
26B9          443 ;
26B9 B9 00 9D 444      lda NBUF1L,Y
26BC          445 ;
26BC 5E 00 9E 446      lsr NBUF2L,X
26BF 2A       447      rol
26C0          448 ;
26C0 5E 00 9E 449      lsr NBUF2L,X
26C3 2A       450      rol
26C4          451 ;
26C4          452 ;
26C4          453 ; Store 8-bit data byte in BUFADR2Z buffer and continue.
26C4          454 ;
26C4 91 3E    455      sta (BUFADR2Z),Y
26C6          456 ;
26C6 C8       457      iny
26C7 D0 ED    458      bne <3
26C9          459 ;
26C9 F0 15    460      beq RWTSEXIT          ; always taken
26CB          461 ;
26CB          462 ;
26CB          463 ; Convert the NBUF1H and NBUF2H nibbles to bytes.
26CB          464 ;
26CB          465 POSTNIBH:
26CB A2 56    466 ^2      ldx #NBUF2SIZ
26CD          467 ;
26CD CA       468 ^3      dex
26CE 30 FB    469      bmi <2
26D0          470 ;
26D0          471 ;
26D0          472 ; Get byte and shift in low order two bits from NBUF2H.
26D0          473 ;
26D0 B9 00 DE 474      lda NBUF1H,Y
26D3          475 ;
26D3 5E 00 DF 476      lsr NBUF2H,X
26D6 2A       477      rol
26D7          478 ;
26D7 5E 00 DF 479      lsr NBUF2H,X
26DA 2A       480      rol
26DB          481 ;
26DB          482 ;
26DB          483 ; Store 8-bit data byte in BUFADR2Z buffer and continue.
26DB          484 ;
26DB 91 3E    485      sta (BUFADR2Z),Y
26DD          486 ;
26DD C8       487      iny

```

```

26DE D0 ED      488          bne <3
26E0            489      ;
26E0            490      ;
26E0            491      ; RWTS exit with no error.  Clear C-flag.
26E0            492      ;
26E0 18          493  RWTSEXIT clc
26E1            494      ;
26E1 A9 00       495          lda #RWNOERR
26E3            496      ;
26E3            497      ;
26E3            498      ; RWTS exit with error in A-reg.
26E3            499      ;
26E3 A0 0D       500  ERREXIT  ldy #ERRCODE-TBLTYPE
26E5            501      ;
26E5 91 4A       502          sta (IOBADR),Y
26E7            503      ;
26E7 A6 2B       504          ldx SLOT16Z
26E9            505      ;
26E9 BC 88 C0    506          ldy MOTOROFF,X
26EC            507      ;
26EC 60          508          rts
26ED            509      ;
26ED            510      ;
26ED            511      ; Routine to format a diskette in the requested drive.
26ED            512      ; Initialize VOLUMEZ from VOLEXPT, ODDBITSZ for WRITADR,
26ED            513      ; and RECALCNT for one recalibration.
26ED            514      ;
26ED A0 03       515  DISKFMT  ldy #VOLEXPT-TBLTYPE
26EF            516      ;
26EF B1 4A       517          lda (IOBADR),Y
26F1 85 41       518          sta VOLUMEZ
26F3            519      ;
26F3 A9 AA       520          lda #ODDBITS
26F5 85 3E       521          sta ODDBITSZ
26F7            522      ;
26F7 A9 02       523          lda #2
26F9 8D F8 04    524          sta RECALCNT
26FC            525      ;
26FC            526      ;
26FC            527      ; Get the ENDTRK and ENDSEC values for DOS 4.X.
26FC            528      ;
26FC AD FA BF    529          lda INITVAL
26FF 85 26       530          sta TEMPZ
2701            531      ;
2701 AD FB BF    532          lda INITVAL+1
2704 85 27       533          sta TEMPZ+1
2706            534      ;
2706 A0 10       535          ldy #ENDTRK45
2708            536      ;
2708 B1 26       537          lda (TEMPZ),Y
270A 8D 16 01    538          sta ENDTRK
270D            539      ;
270D A0 1C       540          ldy #ENDSEC45
270F            541      ;
270F B1 26       542          lda (TEMPZ),Y
2711 8D 17 01    543          sta ENDSEC
2714            544      ;
2714            545      ;
2714            546      ; Begin with MAXSYNC sync bytes and set the disk head
2714            547      ; location to track 48.  Move the disk head to track 0
2714            548      ; and then to track 2 to ensure disk head stability.

```

```

2714          549 ;
2714 A9 20      550 ^1      lda #MAXSYNC
2716 85 35      551          sta SYNCNT
2718          552 ;
2718 A9 C0      553          lda #DFLTPHAS*MAXTRACK ; set track to 48
271A 20 41 21   554          jsr SAVETRK2
271D          555 ;
271D 20 70 24   556          jsr MOVHEAD0          ; returns with zero in A-reg
2720          557 ;
2720 AA         558          tax
2721 A9 02      559          lda #2
2723          560 ;
2723 20 80 24   561          jsr MOVHEAD          ; returns with zero in A-reg
2726          562 ;
2726          563 ;
2726          564 ; Initialize the NBUF1 and NBUF2 nibble buffers. NBUF2
2726          565 ; must follow NBUF1. The time to clear these buffers will
2726          566 ; help to stabilize the disk head on track 2.
2726          567 ;
2726          568 ;          lda #ZERO
2726 A8          569          tay
2727          570 ;
2727 2C 14 01    571          bit NBUFLAG
272A 30 0B      572          bmi CLRBUFRH
272C          573 ;
272C          574 ;
272C          575 ; Clear the NBUF1L and NBUF2L nibble buffers.
272C          576 ;
272C 99 00 9D    577 ^2      sta NBUF1L,Y
272F 99 56 9D    578          sta NBUF1L+NBUF2SIZ,Y
2732          579 ;
2732 C8          580          iny
2733 D0 F7      581          bne <2
2735          582 ;
2735 F0 09      583          beq >3          ; always taken
2737          584 ;
2737          585 ;
2737          586 ; Clear the NBUF1H and NBUF2H nibble buffers.
2737          587 ;
2737          588 CLRBUFRH:
2737 99 00 DE    589 ^2      sta NBUF1H,Y
273A 99 56 DE    590          sta NBUF1H+NBUF2SIZ,Y
273D          591 ;
273D C8          592          iny
273E D0 F7      593          bne <2
2740          594 ;
2740          595 ;
2740          596 ; Initialize TRACKZ and format only track 0 to establish a
2740          597 ; working value for SYNCNT. TRACKFMT returns A-reg = 0.
2740          598 ; Give SYNCNT a final adjustment and begin the disk format.
2740          599 ;
2740 85 40      600 ^3      sta TRACKZ
2742          601 ;
2742 AA         602          tax          ; for STRTSEC
2743 A0 10      603          ldy #HIGHSECS      ; for STOPSEC
2745          604 ;
2745 20 8A 27    605          jsr TRACKFMT          ; returns with zero in A-reg
2748 B0 31      606          bcs >7
274A          607 ;
274A C6 35      608          dec SYNCNT
274C C6 35      609          dec SYNCNT

```

```

274E          610 ;
274E          611 ;
274E          612 ; Format the selected track and increment track number
274E          613 ; until ENDTRK.
274E          614 ;
274E A2 00     615 ^4      ldx #ZERO                ; for STRTSEC
2750          616 ;
2750 AC 15 01  617          ldy DNUM0
2753 C0 03     618          cpy #MAXDRIVE
2755 D0 10     619          bne >5
2757          620 ;
2757          621 ;
2757          622 ; Drive #4 special processing.  If ENDSEC is 0x10, then
2757          623 ; do normal format, otherwise do special format.
2757          624 ;
2757 A0 10     625          ldy #HIGHSECS            ; for STOPSEC
2759 CC 17 01  626          cpy ENDSEC
275C F0 0C     627          beq >6
275E          628 ;
275E 20 8A 27  629          jsr TRACKFMT
2761 B0 18     630          bcs >7
2763          631 ;
2763 A5 40     632          lda TRACKZ                ; recall track number
2765 A2 10     633          ldx #HIGHSECS            ; for STRTSEC
2767          634 ;
2767 AC 17 01  635 ^5      ldy ENDSEC                ; for STOPSEC
276A          636 ;
276A 20 8A 27  637 ^6      jsr TRACKFMT
276D B0 0C     638          bcs >7
276F          639 ;
276F E6 40     640          inc TRACKZ
2771          641 ;
2771 A5 40     642          lda TRACKZ
2773 CD 16 01  643          cmp ENDTRK
2776 D0 D6     644          bne <4
2778          645 ;
2778 4C E0 26  646          jmp RWTSEXIT
277B          647 ;
277B          648 ;
277B          649 ; Error from TRACKFMT.  Try a recalibration, then exit.
277B          650 ;
277B CE F8 04  651 ^7      dec RECALCNT
277E D0 94     652          bne <1
2780          653 ;
2780 A9 08     654          lda #RWINITER            ; track init error
2782          655 ;
2782 4C E3 26  656          jmp ERREXIT
2785          657 ;
2785          658 ;
2785          659 ; Handle write protect sense error.
2785          660 ;
2785 68        661 DORWPERR pla
2786 68        662          pla
2787          663 ;
2787 4C A3 26  664          jmp RWPERR
278A          665 ;
278A          666 ;
278A          667 ; Routine to format the selected track.  Move the disk head
278A          668 ; to the selected track and allow for 32 track retries.
278A          669 ; Mark all sectors as unformatted and initialize SECTORZ to
278A          670 ; zero.

```

```

278A      671 ;
278A 8E 18 01 672 TRACKFMT stx STRTSEC
278D 8C 19 01 673          sty STOPSEC
2790      674 ;
2790 20 80 24 675          jsr MOVHEAD
2793      676 ;
2793 A9 20     677          lda #MAXRETRY
2795 8D F8 05 678          sta RETRYCNT
2798      679 ;
2798 A9 00     680 ^1      lda #ZERO
279A      681 ;
279A AC 19 01 682          ldy STOPSEC
279D      683 ;
279D 99 1F 01 684 ^2      sta SECMAP-1,Y
27A0      685 ;
27A0 88       686          dey
27A1      687 ;
27A1 CC 18 01 688          cpy STRTSEC
27A4 D0 F7    689          bne <2
27A6      690 ;
27A6 84 3F    691          sty SECTORZ
27A8      692 ;
27A8      693 ;
27A8      694 ; Begin with 128 sync bytes before sector 0 address field,
27A8      695 ; then use SYNCNT sync bytes before all other sector
27A8      696 ; address fields.
27A8      697 ;
27A8 A0 80    698          ldy #128
27AA      699 ;
27AA 2C 00 00 700          bit *-*
27AD      701          dfs !-2
27AB      702 ;
27AB      703 ;
27AB      704 ; Write an address field and exit if disk is write
27AB      705 ; protected. Then write the data field. Increment sector
27AB      706 ; number until STOPSEC.
27AB      707 ;
27AB A4 35    708 ^3      ldy SYNCNT
27AD      709 ;
27AD 20 B6 22 710          jsr WRITADR
27B0 B0 D3    711          bcs DORWPERR
27B2      712 ;
27B2 20 3E 22 713          jsr WRITSCTR
27B5      714 ;
27B5 E6 3F    715          inc SECTORZ
27B7      716 ;
27B7 A5 3F    717          lda SECTORZ
27B9 CD 19 01 718          cmp STOPSEC
27BC D0 ED    719          bne <3
27BE      720 ;
27BE      721 ;
27BE      722 ; Ensure delay between last and first sector is a least
27BE      723 ; ( SYNCNT / 2 ) * 100 usecs.
27BE      724 ;
27BE A5 35    725          lda SYNCNT
27C0 4A       726          lsr
27C1      727 ;
27C1 20 06 25 728          jsr MSWAIT
27C4      729 ;
27C4      730 ;
27C4      731 ; Read the first address field found. If STRTSEC is not

```

```

27C4          732 ; found, reduce SYNCNT and try again.
27C4          733 ;
27C4 20 AF 23 734      jsr READADR
27C7 B0 0F    735      bcs >4
27C9          736 ;
27C9 A5 2D    737      lda SECFNDZ
27CB CD 18 01 738      cmp STRTSEC
27CE F0 14    739      beq >7
27D0          740 ;
27D0 C6 35    741      dec SYNCNT
27D2          742 ;
27D2 A5 35    743      lda SYNCNT
27D4 C9 08    744      cmp #MINSYNC
27D6 90 05    745      bcc >5
27D8          746 ;
27D8          747 ;
27D8          748 ; Any error found reading an address or data field, or
27D8          749 ; finding a duplicate address field or chnage in SYNCNT
27D8          750 ; will cause the entire track to be reformatted.
27D8          751 ;
27D8 CE F8 05 752      ^4      dec RETRYCNT
27DB D0 BB    753      bne <1
27DD          754 ;
27DD 38       755      ^5      sec
27DE          756 ;
27DE 60       757      rts
27DF          758 ;
27DF          759 ;
27DF          760 ; Read the next address field and data field.
27DF          761 ;
27DF 20 AF 23 762      ^6      jsr READADR
27E2 B0 F4    763      bcs <4
27E4          764 ;
27E4 20 49 23 765      ^7      jsr READSCTR
27E7 B0 EF    766      bcs <4
27E9          767 ;
27E9          768 ;
27E9          769 ; Mark the map with the sector found and verify this
27E9          770 ; is not a duplicate sector found.
27E9          771 ;
27E9 A6 2D    772      ldx SECFNDZ
27EB          773 ;
27EB BD 20 01 774      lda SECMAP,X
27EE 30 E8    775      bmi <4
27F0          776 ;
27F0 DE 20 01 777      dec SECMAP,X
27F3          778 ;
27F3 C6 3F    779      dec SECTORZ
27F5          780 ;
27F5 A4 3F    781      ldy SECTORZ
27F7 CC 18 01 782      cpy STRTSEC
27FA D0 E3    783      bne <6
27FC          784 ;
27FC 18       785      clc
27FD          786 ;
27FD 60       787      rts
27FE          788 ;
27FE          789 ;
27FE 06       790      byt RANAVERSN
27FF 06       791      byt RANABLD
2800          792 ;

```

```
2800          793  ;  
2800          794          icl "RANA5.L"
```

```
LLOAD RANA5.L,A$4000
```

```

2800          1          ttl "Rana ROM Source Code, RANA5.L"
2800          2          ;
2800          3          ;
2800          4          ; RANA5.L
2800          5          ;
2800          6          ;
2800          7          ; This is the firmware on the Rana Disk Controller card no
2800          8          ; matter which slot it resides in.
2800          9          ;
2800         10          ;
2800         11          .if DEBUG
2800         12          .el
2800         13          phs SLOTR0M0
2800         14          .fi
2800         15          ;
2800         16          ;
2800         17          ; Signature bytes for Rana Controller Card.
2800         18          ;
2800 09 20         19          ora #$20
2802 A0 00         20          ldy #$00
2804 A2 03         21          ldx #$03
2806 86 3C         22          stx ROMTEMPZ
2808         23          ;
2808 2C FF CF         24          bit CLRROM
280B         25          ;
280B 20 02 21        26          jsr GETSLOT          ; falls into SELCHEAD
280E 90 1C         27          bcc BOOT              ; always taken
2810         28          ;
2810         29          ;
2810         30          ; Connect Rana RWTS to DOS.
2810         31          ;
2810 2C FF CF         32  ROMHOOK bit CLRROM
2813         33          ;
2813 EA            34          nop
2814         35          ;
2814 A0 00         36          ldy #ZERO
2816 F0 0E         37          beq >1                ; always taken
2818         38          ;
2818         39          ;
2818         40          ; Disconnect Rana RWTS from DOS.
2818         41          ;
2818 2C FF CF         42  ROMUHOOK bit CLRROM
281B         43          ;
281B EA            44          nop
281C         45          ;
281C A0 FF         46          ldy #NEGONE
281E 30 06         47          bmi >1                ; always taken
2820         48          ;
2820         49          ;
2820         50          ; Entry for CALLRWTS for both Disk ][ and Rana drives.
2820         51          ;
2820 2C FF CF         52  RANARWTS bit CLRROM
2823         53          ;
2823 4C 37 25        54          jmp RWTSENT
2826         55          ;
2826         56          ;
2826 20 02 21        57  ^1          jsr GETSLOT          ; falls into SELCHEAD
2829         58          ;
2829 4C EA 23        59          jmp HOOKRANA
282C         60          ;

```

```

282C      61 ;
282C      62 ; Establish bank 1, read mode, drive 1, and enable motor.
282C      63 ;
282C 8D 00 21 64 BOOT      sta ROMCODE
282F      65 ;
282F BD 8E C0 66          lda DATAIN,X
2832 BD 8C C0 67          lda STROBE,X
2835      68 ;
2835 BD 8A C0 69          lda DRV0EN,X
2838 BD 89 C0 70          lda MOTORON,X
283B      71 ;
283B      72 ;
283B      73 ; Recalibrate the disk head. Assume the disk head is at
283B      74 ; track 48. Y-reg must be less than 0x80.
283B      75 ;
283B      76 ; The disk drive uses a four phase motor to move the track
283B      77 ; I/O sensor. Each phase must be energized for a period of
283B      78 ; time, then de-energized. To cause the motor to rotate,
283B      79 ; the next phase windings are selected sequentially.
283B      80 ;
283B A0 60      81          ldy #MAXTRACK*DFLTPHAS/2
283D      82 ;
283D BD 80 C0 83 ^2      lda PHASEOFF,X
2840      84 ;
2840 98      85          tya
2841      86 ;
2841 29 03      87          and #$03
2843 0A      88          asl
2844 05 2B      89          ora SLOT16Z
2846      90 ;
2846 AA      91          tax
2847      92 ;
2847 BD 81 C0 93          lda PHASEON,X
284A      94 ;
284A A9 56      95          lda #$56
284C 20 A8 FC 96          jsr WAIT
284F      97 ;
284F 88      98          dey
2850 10 EB      99          bpl <2
2852      100 ;
2852      101 ;
2852      102 ; Initialize variables with A-reg = 0 (returned from WAIT).
2852      103 ;
2852 85 26      104          sta BUFRADRZ
2854 85 3D      105          sta ROMSECTR
2856 85 41      106          sta ROMTRACK
2858      107 ;
2858 A9 08      108          lda /PAGE08
285A 85 27      109          sta BUFRADRZ+1
285C      110 ;
285C      111 ;
285C      112 ; Boot firmware entry. If C-flag = 0, read address header.
285C      113 ; If C-flag = 1, then read data header.
285C      114 ;
285C 18      115 BOOTFW    clc
285D      116 ;
285D 08      117 BOOTFW2   php
285E      118 ;
285E      119 ;
285E      120 ; Look for first address/data mark (0xD5).
285E      121 ;

```

```

285E BD 8C C0    122  ^1      lda STROBE,X
2861 10 FB      123          bpl <1
2863           124  ;
2863 49 D5      125  ^2      eor #ADRMARK1
2865 D0 F7      126          bne <1
2867           127  ;
2867           128  ;
2867           129  ; Look for second address/data mark (0xAA).
2867           130  ;
2867 BD 8C C0    131  ^3      lda STROBE,X
286A 10 FB      132          bpl <3
286C           133  ;
286C C9 AA      134          cmp #ADRMARK2
286E D0 F3      135          bne <2
2870           136  ;
2870 EA         137          nop
2871           138  ;
2871           139  ;
2871           140  ; Look for third address or data mark (0x96 or 0xAD).
2871           141  ;
2871 BD 8C C0    142  ^4      lda STROBE,X
2874 10 FB      143          bpl <4
2876           144  ;
2876 C9 96      145          cmp #ADRMARK3
2878 F0 09      146          beq FNDADDR
287A           147  ;
287A 28         148          plp
287B 90 E0      149          bcc BOOTFW2
287D           150  ;
287D 49 AD      151          eor #DATMARK3
287F F0 25      152          beq FNDDATA
2881           153  ;
2881 D0 D9      154          bne BOOTFW          ; always taken
2883           155  ;
2883           156  ;
2883           157  ; An Address header has been found. Read in volume, track,
2883           158  ; and sector; checksum is ignored. Only looking for sector
2883           159  ; number in A-reg. ROMDATA will contain the track number.
2883           160  ;
2883 A0 03      161  FNDADDR  ldy #3
2885           162  ;
2885 85 40      163  ^1      sta ROMDATA
2887           164  ;
2887 BD 8C C0    165  ^2      lda STROBE,X
288A 10 FB      166          bpl <2
288C           167  ;
288C 2A         168          rol
288D 85 3C      169          sta ROMTEMPZ
288F           170  ;
288F BD 8C C0    171  ^3      lda STROBE,X
2892 10 FB      172          bpl <3
2894           173  ;
2894 25 3C      174          and ROMTEMPZ
2896           175  ;
2896 88         176          dey
2897 D0 EC      177          bne <1
2899           178  ;
2899           179  ;
2899           180  ; Fix stack pointer from BOOTFW and check for desired
2899           181  ; sector number.
2899           182  ;

```

```

2899 28          183          plp
289A          184          ;
289A C5 3D      185          cmp ROMSECTR
289C D0 BE      186          bne BOOTFW
289E          187          ;
289E          188          ;
289E          189          ; At correct sector. Check for desired track number.
289E          190          ;
289E A5 40      191          lda ROMDATA
28A0 C5 41      192          cmp ROMTRACK
28A2 D0 B8      193          bne BOOTFW
28A4          194          ;
28A4          195          ;
28A4          196          ; At correct track and sector. Read the following data
28A4          197          ; header and the data. C-flag is now set.
28A4          198          ;
28A4 F0 B7      199          beq BOOTFW2          ; always taken
28A6          200          ;
28A6          201          ;
28A6          202          ; A data header has been found. Read in the first 0x56
28A6          203          ; nibbles as the index into the RDNIBL table found at
28A6          204          ; 0xC896, and save the data in NBUF2BT at 0x0300.
28A6          205          ;
28A6 A0 56      206          FNDDATA ldy #NBUF2SIZ
28A8          207          ;
28A8 84 3C      208          ^1          sty ROMTEMPZ
28AA          209          ;
28AA BC 8C C0   210          ^2          ldy STROBE,X
28AD 10 FB      211          bpl <2
28AF          212          ;
28AF 59 00 21   213          eor RDNIBL-$96,Y
28B2          214          ;
28B2 A4 3C      215          ldy ROMTEMPZ
28B4          216          ;
28B4 88         217          dey
28B5          218          ;
28B5 99 00 03   219          sta NBUF2BT,Y
28B8          220          ;
28B8 D0 EE      221          bne <1
28BA          222          ;
28BA          223          ;
28BA          224          ; Read in the next 0x100 nibbles as the index into the
28BA          225          ; RDNIBL table, and save the data at the requested memory
28BA          226          ; location, thus NBUF1 is not necessary since data is
28BA          227          ; already on a page boundary.
28BA          228          ;
28BA 84 3C      229          ^3          sty ROMTEMPZ
28BC          230          ;
28BC BC 8C C0   231          ^4          ldy STROBE,X
28BF 10 FB      232          bpl <4
28C1          233          ;
28C1 59 00 21   234          eor RDNIBL-$96,Y
28C4          235          ;
28C4 A4 3C      236          ldy ROMTEMPZ
28C6          237          ;
28C6 91 26      238          sta (BUFRADRZ),Y
28C8          239          ;
28C8 C8         240          iny
28C9 D0 EF      241          bne <3
28CB          242          ;
28CB          243          ;

```

```

28CB      244 ; Finally read in checksum and test for 0.
28CB      245 ;
28CB BC 8C C0 246 ^5      ldy STROBE,X
28CE 10 FB   247          bpl <5
28D0      248 ;
28D0 59 00 21 249          eor RDNIBL-$96,Y
28D3      250 ;
28D3 D0 87   251 ^6      bne BOOTFW
28D5      252 ;
28D5      253 ;
28D5      254 ; Post nibblize the data buffer with NBUF2BT.
28D5      255 ;
28D5 A0 00   256          ldy #ZERO
28D7      257 ;
28D7 A2 56   258 ^7      ldx #NBUF2SIZ
28D9      259 ;
28D9 CA     260 ^8      dex
28DA 30 FB   261          bmi <7
28DC      262 ;
28DC B1 26   263          lda (BUFRADRZ),Y
28DE      264 ;
28DE 5E 00 03 265          lsr NBUF2BT,X
28E1 2A     266          rol
28E2      267 ;
28E2 5E 00 03 268          lsr NBUF2BT,X
28E5 2A     269          rol
28E6      270 ;
28E6 91 26   271          sta (BUFRADRZ),Y
28E8      272 ;
28E8 C8     273          iny
28E9 D0 EE   274          bne <8
28EB      275 ;
28EB      276 ;
28EB      277 ; Check if DOS 4.X RWTS is now in memory. Y-reg is zero.
28EB      278 ;
28EB 20 2F 24 279          jsr CHKDOS4X
28EE      280 ;
28EE      281 ;
28EE      282 ; Increment data buffer address and sector number.
28EE      283 ;
28EE E6 27   284          inc BUFRADRZ+1
28F0 E6 3D   285          inc ROMSECTR
28F2      286 ;
28F2      287 ;
28F2      288 ; If sector number is less than 1 found at location 0x0800,
28F2      289 ; start looking for the requested track and sector. For
28F2      290 ; sector numbers greater than 0x00, enter the Boot Stage 1
28F2      291 ; code at 0x0801.
28F2      292 ;
28F2 A5 3D   293          lda ROMSECTR
28F4 CD 00 08 294          cmp PAGE08
28F7      295 ;
28F7 A6 2B   296          ldx SLOT16Z
28F9      297 ;
28F9 90 D8   298          bcc <6 ; need non-zero value set also
28FB      299 ;
28FB 4C 01 08 300          jmp PAGE08+1 ; process boot stage 0
28FE      301 ;
28FE      302 ;
28FE 06     303          byt RANAVRSN
28FF 06     304          byt RANABLD

```

```
2900          305  ;
2900          306  ;
```

```
BSAVE RANA,A$2000,B,L$0900
```

```
2900          307          usr RANA
2900          308  ;
2900          309  ;
2900          310          stt "Rana ROM Symbol Table"
2900          311  ;
2900          312  ;
2900          313          end 111
```

```
*** End of Assembly
```

Symbol List starts at 0x7800, ends at 0x807A, used 0x087A, remaining 0x358E

Symbols unsorted:

LOC0	0000	BUFRADRZ	0026	TEMPZ	0026	TEMP2Z	0027	CURTRKZ	002A
SLOT16Z	002B	DRVFLAG	002C	ADRDATMK	002C	ADRFIELD	002C	SECFNDZ	002D
TRKFNDZ	002E	VOLFNDZ	002F	PHASE	0034	SYNCNT	0035	ROMTEMPZ	003C
MOTORTIM	003C	ROMSECTR	003D	BUFADR2Z	003E	ODDBITSZ	003E	SECTORZ	003F
ROMDATA	0040	TRACKZ	0040	ROMTRACK	0041	VOLUMEZ	0041	IOBADR	004A
DEBUG	0001	ZERO	0000	NEGONE	00FF	RANAVRSN	0006	RANABLD	0006
DOS4VRSN	0045	DOS4BLD	0006	SLOTMASK	0007	SECMASK	000F	TRKMASK	003F
MAXDRIVE	0003	DFLTPHAS	0004	PHASMAX	0010	MAXTRACK	0030	HDRSYNC	0006
MINSYNC	0008	MAXSYNC	0020	MAXRETRY	0020	SYNCBITS	0500	HIGHSECS	0010
MAXSEC	0020	ENDTRK45	0010	ENDSEC45	001C	MOTONTIM	D8EF	TBLTYPE	0000
SNUM16	0001	DNUM	0002	VOLEXPT	0003	TNUM	0004	SNUM	0005
USRBUF	0008	IOCBPHAS	000A	BYTCNT	000B	CMDCODE	000C	ERRCODE	000D
VOLFND	000E	SLOTFND	000F	DRVFND	0010	RWTSSEEK	0000	RWTSREAD	0001
RWTSWRIT	0002	RWTSFRMT	0004	RWNOERR	0000	RWINITER	0008	RWPROTER	0010
RWVOLERR	0020	RWSYNERR	0030	RWDRVERR	0040	RWREADER	0080	NBUF2SIZ	0056
ODDBITS	00AA	ADRMARK1	00D5	ADRMARK2	00AA	ADRMARK3	0096	DATMARK1	00D5
DATMARK2	00AA	DATMARK3	00AD	SLPMARK1	00DE	SLPMARK2	00AA	SLPMARK3	00EB
SYNCKMARK	00FF	PAGESIZE	0100	STACK	0100	SAVYREG	0112	SECTOR	0113
NBUFLAG	0114	DNUM0	0115	ENDTRK	0116	ENDSEC	0117	STRTSEC	0118
STOPSEC	0119	HOOKCODE	011A	SECMAP	0120	NBUF2BT	0300	DOSWARM	03D0
HOOKDOS	03EA	FINDTRK	0478	RECALCNT	04F8	XMODE	04FB	SEEKCNT	0578
RETRYCNT	05F8	NEXTON	0678	NEXTOFF	06F8	SLOT16	0778	MSLOT	07F8
DRV0TRK	0478	DRV1TRK	04F8	DRV2TRK	0578	DRV3TRK	05F8	DRV0PHAS	0678
DRV1PHAS	06F8	DRV2PHAS	0778	DRV3PHAS	07F8	BOOTADR	08FE	BOOTPGS	08FF
PAGE08	0800	PAGE10	1000	PAGE20	2000	NBUF1L	9D00	NBUF2L	9E00
NBUF1H	DE00	NBUF2H	DF00	BLDVRSN	BFF0	BLDNMBR	BFF1	MNGDISK	BFF2
INITVAL	BFFA	NBUF1PG	BFFD	ROM2WP	C082	RAM1WE	C08B	PHASEOFF	C080
PHASEON	C081	MOTOROFF	C088	MOTORON	C089	DRV0EN	C08A	DRV1EN	C08B
STROBE	C08C	LATCH	C08D	DATAIN	C08E	DATAOUT	C08F	SLOTROM0	C000
SLOTROM8	C800	CLRROM	CFFF	DISKADRS	D003	WAIT	FCA8	IORTS	FF58
ROMCODE	2100	GETSLOT	2102	SELCEHAD	2111	SAVETRK	2124	SAVETRK2	2141
WRNIBL	2156	RDNIBL	2196	WAIT52	21C0	WAIT48	21C2	WAIT24	21C5
WAIT12	21C8	WRITBUFH	2200	WRITSCTR	223E	WRITRTN	228C	WRITEEXIT	2295
SETREAD	22AF	WRITADR	22B6	INTRLEAV	22E4	WRITSYN0	2304	WRITSYNC	2308
WBYTE	2330	WNIBL9	233F	WNIBL2	2341	WNIBL	2342	READSCTR	2349
READBUFH	2381	READERR	23AD	READADR	23AF	READEXIT	23D5	HOOKRANA	23EA
EXITCODE	23FB	EXITLEN	0006	READMRKS	2401	CHKDOS4X	242F	DISKMNG	2454
MOVHEAD0	2470	MOVHEADN	2475	MOVHEAD	2480	MOVRTN	248C	MOVEHD	248D
CHKPOS	24EB	MSWAIT	2506	DRVNDX	2517	ONOFFTBL	252B	OTBLLEN	000C
RWTSENT	2537	SYNERR	257A	PRENIBLH	2624	RWPERR	26A3	RWTSEERR	26A5
POSTNIBH	26CB	RWTSEXIT	26E0	ERREXIT	26E3	DISKFMT	26ED	CLRBURFH	2737
DORWPERR	2785	TRACKFMT	278A	ROMHOOK	2810	ROMUHOOK	2818	RANARWTS	2820
BOOT	282C	BOOTFW	285C	BOOTFW2	285D	FNDADDR	2883	FNDDATA	28A6

Symbols alphabetically sorted:

ADRDATMK	002C	ADRFIELD	002C	ADRMARK1	00D5	ADRMARK2	00AA	ADRMARK3	0096
BLDNMBR	BFF1	BLDVRSN	BFF0	BOOT	282C	BOOTADR	08FE	BOOTFW	285C
BOOTFW2	285D	BOOTPGS	08FF	BUFADR2Z	003E	BUFRADRZ	0026	BYTCNT	000B
CHKDOS4X	242F	CHKPOS	24EB	CLRBURFH	2737	CLRROM	CFFF	CMDCODE	000C
CURTRKZ	002A	DATAIN	C08E	DATAOUT	C08F	DATMARK1	00D5	DATMARK2	00AA
DATMARK3	00AD	DEBUG	0001	DFLTPHAS	0004	DISKADRS	D003	DISKFMT	26ED
DISKMNG	2454	DNUM	0002	DNUM0	0115	DORWPERR	2785	DOS4BLD	0006
DOS4VRSN	0045	DOSWARM	03D0	DRV0EN	C08A	DRV0PHAS	0678	DRV0TRK	0478

DRV1EN	C08B	DRV1PHAS	06F8	DRV1TRK	04F8	DRV2PHAS	0778	DRV2TRK	0578
DRV3PHAS	07F8	DRV3TRK	05F8	DRVFLAG	002C	DRVFND	0010	DRVNDX	2517
ENDSEC	0117	ENDSEC45	001C	ENDTRK	0116	ENDTRK45	0010	ERRCODE	000D
ERREXIT	26E3	EXITCODE	23FB	EXITLEN	0006	FINDTRK	0478	FNDADDR	2883
FNDATA	28A6	GETSLOT	2102	HDRSYNC	0006	HIGHSECS	0010	HOOKCODE	011A
HOOKDOS	03EA	HOOKRANA	23EA	INITVAL	BFFA	INTRLEAV	22E4	IOBADR	004A
IOCBPHAS	000A	IORTS	FF58	LATCH	C08D	LOC0	0000	MAXDRIVE	0003
MAXRETRY	0020	MAXSEC	0020	MAXSYNC	0020	MAXTRACK	0030	MINSYNC	0008
MNGDISK	BFF2	MOTONTIM	D8EF	MOTOROFF	C088	MOTORON	C089	MOTORTIM	003C
MOVEHD	248D	MOVHEAD	2480	MOVHEAD0	2470	MOVHEADN	2475	MOVRTN	248C
MSLOT	07F8	MSWAIT	2506	NBUF1H	DE00	NBUF1L	9D00	NBUF1PG	BFFD
NBUF2BT	0300	NBUF2H	DF00	NBUF2L	9E00	NBUF2SIZ	0056	NBUFLAG	0114
NEGONE	00FF	NEXTOFF	06F8	NEXTON	0678	ODDBITS	00AA	ODDBITSZ	003E
ONOFFTBL	252B	OTBLLEN	000C	PAGE08	0800	PAGE10	1000	PAGE20	2000
PAGESIZE	0100	PHASE	0034	PHASEOFF	C080	PHASEON	C081	PHASMAX	0010
POSTNIBH	26CB	PRENIBLH	2624	RAM1WE	C08B	RANABLD	0006	RANARWTS	2820
RANAVRSN	0006	RDNIBL	2196	READADR	23AF	READBUFH	2381	READERR	23AD
REDEXIT	23D5	READMRKS	2401	READSCTR	2349	RECALCNT	04F8	RETRYCNT	05F8
ROM2WP	C082	ROMCODE	2100	ROMDATA	0040	ROMHOOK	2810	ROMSECTR	003D
ROMTEMPZ	003C	ROMTRACK	0041	ROMUHOOK	2818	RWDRVERR	0040	RWINITER	0008
RWNOERR	0000	RWPERR	26A3	RWPROTER	0010	RWREADER	0080	RWSYNERR	0030
RWTSENT	2537	RWTSERR	26A5	RWTSEXIT	26E0	RWTSFRMT	0004	RWTSREAD	0001
RWTSSEEK	0000	RWTSWRIT	0002	RWVOLERR	0020	SAVETRK	2124	SAVETRK2	2141
SAVYREG	0112	SECFNDZ	002D	SECMAP	0120	SECMASK	000F	SECTOR	0113
SECTORZ	003F	SEEKCNT	0578	SELCHD	2111	SETREAD	22AF	SLOT16	0778
SLOT16Z	002B	SLOTFND	000F	SLOTMASK	0007	SLOTROM0	C000	SLOTROM8	C800
SLPMARK1	00DE	SLPMARK2	00AA	SLPMARK3	00EB	SNUM	0005	SNUM16	0001
STACK	0100	STOPSEC	0119	STROBE	C08C	STRTSEC	0118	SYNCBITS	0500
SYNCMARK	00FF	SYNCNT	0035	SYNERR	257A	TBLTYPE	0000	TEMP2Z	0027
TEMPZ	0026	TNUM	0004	TRACKFMT	278A	TRACKZ	0040	TRKFNDZ	002E
TRKMASK	003F	USRBUF	0008	VOLEXPT	0003	VOLFND	000E	VOLFNDZ	002F
VOLUMEZ	0041	WAIT	FCA8	WAIT12	21C8	WAIT24	21C5	WAIT48	21C2
WAIT52	21C0	WBYTE	2330	WNIBL	2342	WNIBL2	2341	WNIBL9	233F
WRITADR	22B6	WRITBUFH	2200	WRITEXIT	2295	WRITRTN	228C	WRITSCTR	223E
WRITSYN0	2304	WRITSYNC	2308	WRNIBL	2156	XMODE	04FB	ZERO	0000

Symbols numerically sorted:

ZERO	0000	TBLTYPE	0000	RWTSSEEK	0000	RWNOERR	0000	LOC0	0000
SNUM16	0001	RWTSREAD	0001	DEBUG	0001	RWTSWRIT	0002	DNUM	0002
VOLEXPT	0003	MAXDRIVE	0003	TNUM	0004	RWTSFRMT	0004	DFLTPHAS	0004
SNUM	0005	RANAVRSN	0006	RANABLD	0006	HDRSYNC	0006	EXITLEN	0006
DOS4BLD	0006	SLOTMASK	0007	USRBUF	0008	RWINITER	0008	MINSYNC	0008
IOCBPHAS	000A	BYTCNT	000B	OTBLLEN	000C	CMDCODE	000C	ERRCODE	000D
VOLFND	000E	SLOTFND	000F	SECMASK	000F	RWPROTER	0010	PHASMAX	0010
HIGHSECS	0010	ENDTRK45	0010	DRVFND	0010	ENDSEC45	001C	RWVOLERR	0020
MAXSYNC	0020	MAXSEC	0020	MAXRETRY	0020	TEMPZ	0026	BUFRADRZ	0026
TEMP2Z	0027	CURTRKZ	002A	SLOT16Z	002B	DRVFLAG	002C	ADRFIELD	002C
ADRDATMK	002C	SECFNDZ	002D	TRKFNDZ	002E	VOLFNDZ	002F	RWSYNERR	0030
MAXTRACK	0030	PHASE	0034	SYNCNT	0035	ROMTEMPZ	003C	MOTORTIM	003C
ROMSECTR	003D	ODDBITSZ	003E	BUFADR2Z	003E	TRKMASK	003F	SECTORZ	003F
TRACKZ	0040	RWDRVERR	0040	ROMDATA	0040	VOLUMEZ	0041	ROMTRACK	0041
DOS4VRSN	0045	IOBADR	004A	NBUF2SIZ	0056	RWREADER	0080	ADRMARK3	0096
SLPMARK2	00AA	ODDBITS	00AA	DATMARK2	00AA	ADRMARK2	00AA	DATMARK3	00AD
DATMARK1	00D5	ADRMARK1	00D5	SLPMARK1	00DE	SLPMARK3	00EB	SYNCMARK	00FF
NEGONE	00FF	STACK	0100	PAGESIZE	0100	SAVYREG	0112	SECTOR	0113
NBUFLAG	0114	DNUM0	0115	ENDTRK	0116	ENDSEC	0117	STRTSEC	0118
STOPSEC	0119	HOOKCODE	011A	SECMAP	0120	NBUF2BT	0300	DOSWARM	03D0
HOOKDOS	03EA	FINDTRK	0478	DRV0TRK	0478	RECALCNT	04F8	DRV1TRK	04F8

XMODE	04FB	SYNCBITS	0500	SEEKCNT	0578	DRV2TRK	0578	RETRYCNT	05F8
DRV3TRK	05F8	NEXTON	0678	DRV0PHAS	0678	NEXTOFF	06F8	DRV1PHAS	06F8
SLOT16	0778	DRV2PHAS	0778	MSLOT	07F8	DRV3PHAS	07F8	PAGE08	0800
BOOTADR	08FE	BOOTPGS	08FF	PAGE10	1000	PAGE20	2000	ROMCODE	2100
GETSLOT	2102	SELCHHEAD	2111	SAVETRK	2124	SAVETRK2	2141	WRNIBL	2156
RDNIBL	2196	WAIT52	21C0	WAIT48	21C2	WAIT24	21C5	WAIT12	21C8
WRITBUFH	2200	WRITSCTR	223E	WRITRTN	228C	WRITEEXIT	2295	SETREAD	22AF
WRITADR	22B6	INTRLEAV	22E4	WRITSYN0	2304	WRITSYNC	2308	WBYTE	2330
WNIBL9	233F	WNIBL2	2341	WNIBL	2342	READSCTR	2349	READBUFH	2381
READERR	23AD	READADR	23AF	READEXIT	23D5	HOOKRANA	23EA	EXITCODE	23FB
READMRKS	2401	CHKDOS4X	242F	DISKMNG	2454	MOVHEAD0	2470	MOVHEADN	2475
MOVHEAD	2480	MOVRTN	248C	MOVEHD	248D	CHKPOS	24EB	MSWAIT	2506
DRVNDEX	2517	ONOFFTBL	252B	RWTSENT	2537	SYNERR	257A	PRENIBLH	2624
RWPERR	26A3	RWTSEERR	26A5	POSTNIBH	26CB	RWTSEXIT	26E0	ERREXIT	26E3
DISKFMT	26ED	CLRBURFH	2737	DORWPERR	2785	TRACKFMT	278A	ROMHOOK	2810
ROMUHOOK	2818	RANARWTS	2820	BOOT	282C	BOOTFW	285C	BOOTFW2	285D
FNDADDR	2883	FNDDATA	28A6	NBUF1L	9D00	NBUF2L	9E00	BLDVRSN	BFF0
BLDNMBR	BFF1	MNGDISK	BFF2	INITVAL	BFFA	NBUF1PG	BFFD	SLOTROM0	C000
PHASEOFF	C080	PHASEON	C081	ROM2WP	C082	MOTOROFF	C088	MOTORON	C089
DRV0EN	C08A	RAM1WE	C08B	DRV1EN	C08B	STROBE	C08C	LATCH	C08D
DATAIN	C08E	DATAOUT	C08F	SLOTROM8	C800	CLRROM	CFFF	DISKADRS	D003
MOTONTIM	D8EF	NBUF1H	DE00	NBUF2H	DF00	WAIT	FCA8	IORTS	FF58