```
00001                           NAM     MON59F
00002                           OPT     PAG,NOG


00005                   *** F3B2 SCRATCH, E004 I/O, SSB CARD
00006                   *** FOUR DRIVE CARD AND DCB4
00007                   *** 16X CLOCK
00008                   *** MODIFIED MAPPER
00009
00010                   ********************************************************
00011                   *                                                      *
00012                   *  Computerware version reconstructed from SSB         *
00013                   *   Find removed & Exchg added to swap map blox        *
00014                   *   Now finds and allocates all ram.                   *
00015                   *                                                      *
00016                   ********************************************************


00018                   * THIS IS A BASE LEVEL MONITOR PROGRAM FOR
00019                   * THE 6809. IT SUPPORTS ONLY THOSE FEATURES
00020                   * WHICH ARE ESSENTIAL IN A SIMPLE DEBUGGER
00021                   * MONITOR.
00022                   *
00023                   * THE ROM IS ADDRESSED AT $F800; IT INCLUDES THE
00024                   * VECTORS FOR THE CPU, AND THE MONITOR PROGRAM
00025                   * PROVIDES SPACE WITHIN ITSELF TO ACCOMMODATE A
00026                   * DISK FIRMWARE PACKAGE SUITABLE FOR COLD START
00027                   * BOOTSTRAPPING.
00028                   *
00029                   * THE SYSTEM MUST HAVE AT LEAST 4K OF RAM,
00030                   * BEFORE THE MONITOR CAN FUNCTION, BUT IT CAN
00031                   * BE ADDRESSED ANYWHERE.  NEEDS 12K FOR DOS69
00032                   * TO BOOT CORRECTLY.
00033
00034                   * IT IS SUGGESTED THAT THE USER CONNECT THE
00035                   * 'PANIC BUTTON' TO NMI, RATHER THAN RESET.
00036                   * OPERATING THE RESET "INTERRUPT" CAUSES ALL
00037                   * THE VECTORS TO BE RE-INITIALIZED, LOSING
00038                   * WHATEVER YOU MIGHT HAVE HAD SET UP. THE
00039                   * NMI VECTOR DEFAULTS TO RESTRT, WHICH MERELY
00040                   * RESETS THE STACK POINTER AND PROCEEDS TO
00041                   * THE COMMAND LEVEL. OF COURSE, IF A WAYWARD
00042                   * PROGRAM HAS DESTROYED THE MONITOR'S SCRATCH
00043                   * AREA, THEN A RESET WILL BE NECESSARY. POWERING
00044                   * DOWN WILL ACHIEVE THIS.
00045                   *
00046                   * NMI DEFAULTS TO A REG DUMP ON POWER UP
```

```
00049                    *      MONITOR FUNCTIONS
00050                    *
00051                    * A   EXAMINE & CHANGE A REGISTER
00052                    * B   EXAMINE & CHANGE B REGISTER
00053                    * C   EXAMINE & CHANGE CC REGISTER
00054                    * D   EXAMINE & CHANGE DP REGISTER
00055                    * E   EXCHANGE TWO MAPPED BLOCKS
00056                    * F   FLEX WARM START
00057                    * /F  FLEX DD BOOT
00058                    * G   GO TO USER PROGRAM
00059                    * H   (NOT USED)
00060                    * /H  HARD DISK BOOT
00061                    * I   INIT MEMORY
00062                    * J   JUMP (SUBROUTINE) TO ADDR
00063                    * /J  JUMP INDIRECT (SUB) TO ADDR
00064                    * K   SET BREAKPOINT
00065                    * L   (NOT USED)
00066                    * M   MEMORY EXAMINE & CHANGE
00067                    * N   (NOT USED)
00068                    * O   (NOT USED)
00069                    * P   EXAMINE & CHANGE PC REGISTER
00070                    * Q   QUICK START (BFD DISK BOOT)
00071                    * /Q  QUICK START (DCB4 DISK BOOT)
00072                    * R   REGISTER DUMP
00073                    * /R  RESET STACK POINTER
00074                    * S   SEARCH FOR 1,2,3 BYTE STRING (HEX)
00075                    * T   TEXT INPUT INTO MEMORY
00076                    * U   EXAMINE & CHANGE U REGISTER
00077                    * V   (NOT USED)
00078                    * W   WARM START INTO DOS
00079                    * X   EXAMINE & CHANGE X REGISTER
00080                    * Y   EXAMINE & CHANGE Y REGISTER
00081                    * Z   DISPLAY FORMATTED MEMORY DUMP
00082                    * /Z  DISPLAY MAPRAM
00083                    * CTRL K  CLEAR ALL BREAKPOINTS
00084                    * .   REOPEN LAST ADDRESS USED IN 'M'
00085                    *
00086                    *  THE FOLLOWING COMMANDS ARE USABLE WHEN A
00087                    *  MEMORY LOCATION IS OPEN IN 'M':
00088                    * :   OPEN THE ADDRESS AT M,M+1
00089                    * @   OPEN THE ADDRESS AT M-1,M
00090                    *  THESE ARE USEFUL IN TRACING INDIRECT ADDRESSING
```

```
00093                           * RELEASE CONTROL INFO
00094              0001         RELEAS  EQU     1
00095              0005         EDIT    EQU     5
00096
00097              0009         MONTH   EQU     $09
00098              0022         DAY     EQU     $22
00099              0083         YEAR    EQU     $83
00100
00101              0035         VERSON  EQU     '5
00102              0048         DVRVER  EQU     'H
00103
00104
00105                           * SYSTEM EQUATES
00106              E000         IO      EQU     $E000     I/O BASE ADDRESS
00107              FFF0         MAPRAM  EQU     $FFF0     ADDRESS MAPPER RAM
00108              FFE0         LATCH   EQU     $FFE0     PAGE LATCH FOR SSB
00109              E004         ACIAS   EQU     IO+4      PORT 1
00110              0015         ACINZ   EQU     $15       FOR TIMES 16 CLK
00111                           *ACINZ EQU $16 FOR TIMES 64 CLK
00112                           *FLIPBT EQU $0F FOR SWTP MAPPER
00113              0000         FLIPBT  EQU     0         FOR SSB MAPPER
00114                           *FLIPB2 EQU 0 FOR SWTP
00115              00F0         FLIPB2  EQU     $F0       FOR SSB
00116              F700         CLOCK   EQU     $F700     SSB RTC
00117                           *LRAFIX EQU $53 'COMB FOR SWTP
00118              0012         LRAFIX  EQU     $12       'NOP FOR SSB
00119              D283         ZWARMS  EQU     $D283     DOS WARM START
00120
00121                           * ROM MONITOR STORAGE SPACE
00122  F3BE                             ORG     $F3BE     FOR SWT COMPATIBILITY
00123              F3B2         STACK   EQU     *-12
00124  F3BE        0002         VNMI    RMB     2         NMI VECTOR
00125  F3C0        0002         VRESVD  RMB     2         RESERVED
00126  F3C2        0002         VSWI3   RMB     2         SWI3 VECTOR
00127  F3C4        0002         VSWI2   RMB     2         SWI2 VECTOR
00128  F3C6        0002         VFIRQ   RMB     2         FIRQ VECTOR
00129  F3C8        0002         VIRQ    RMB     2         IRQ VECTOR
00130  F3CA        0002         VSWI    RMB     2         SWI VECTOR
00131  F3CC        0002         SVCO    RMB     2         SERVICE TABLE BASE ADDRESS
00132  F3CE        0002         SVCL    RMB     2         SERVICE TABLE TOP ADDRESS
00133  F3D0        0010         MAPTBL  RMB     16        MEMORY MAPPING TABLE
00134  F3E0        0002         SP      RMB     2         SAVE STACK PTR ON SWI
00135  F3E2        0002         CPORT   RMB     2         CONTROL PORT ADDR
00136  F3E4        0002         MEM     RMB     2         LAST ADDR EXAMINED WITH 'M'
00137
00138              F3E6         RAME    EQU     *
00139
00140                           * DISK FIRMWARE RAM SPACE
00141
00142  F3E6        0008                 RMB     8
00143
00144                           * BREAKPOINT TABLE SPACE
00145
00146  F3EE        0012         BPTBL   RMB     6*3       BREAKPOINT TABLE
00147              F400         BPTBLE  EQU     *
```

```
00150  F800                        ORG     $F800     IT LIVES ON TOP OF THE WORLD
00151                      *
00152                      * JUMP VECTOR TABLE
00153                      *
00154  F800      FF4A              FDB     MONIT     MONITOR COLD START
00155  F802      FC2F              FDB     CMD       MONITOR SOFT START
00156  F804      F84A              FDB     INCH      GET RAW CHAR FROM TERM
00157  F806      F864              FDB     INEEE     GET CHAR & ECHO IT
00158  F808      F840              FDB     INCHK     CHECK FOR CHAR WAITING
00159  F80A      F868              FDB     OUTCH     OUTPUT A CHAR TO TERM
00160  F80C      F87F              FDB     PDATA     OUTPUT A STRING (DELIM = 04)
00161  F80E      F896              FDB     PCRLF     OUTPUT A CRLF
00162  F810      F879              FDB     PSTR      OUTPUT CRLF, THEN PRINT STRI
00163  F812      FB76              FDB     LRA       LOAD REAL ADDRESS
00164  F814      FC20              FDB     RESTRT    MONITOR WARM RESTART
00165  F816      ED09    DBOOT     FDB     $ED09     DDEN BOOT
00166  F818      E800    FBOOT     FDB     $E800
00167  F81A      EB09    HBOOT     FDB     $EB09
00168  F81C      FFC0              FDB     MOVTBL
00169
00170             F81E    JUMPE     EQU     *
00171
00172                      * DISK JUMP AREA
00173  F81E      0022              RMB     17*2
00174
```

```
00177                          * INPUT CHARACTER READY CHECK
00178
00179  F840 34   02       INCHK  PSHS   A          *CHECK FOR CHAR READY*
00180  F842 A6   9F F3E2         LDA    [CPORT]
00181  F846 84   01              ANDA   #1         RDRF
00182  F848 35   82              PULS   A,PC
00183
00184                          * INPUT CHARACTER WITH PARITY BIT
00185
00186  F84A 8D   F4   F840 INCH   BSR    INCHK     *GET A RAW CHARACTER INTO A*
00187  F84C 27   FC   F84A        BEQ    INCH      WAIT UNTIL OP TYPES ONE
00188  F84E 34   10       INCH1  PSHS   X
00189  F850 BE   F3E2             LDX    CPORT
00190  F853 A6   01               LDA    1,X
00191  F855 35   90               PULS   X,PC
00192
00193                          * INPUT A CHAR WITH CASE FOLDING
00194
00195  F857 8D   0B   F864 INEE   BSR    INEEE
00196  F859 81   61               CMPA   #'a
00197  F85B 25   06   F863        BLO    INEER
00198  F85D 81   7A               CMPA   #'z
00199  F85F 22   02   F863        BHI    INEER
00200  F861 80   20               SUBA   #$20
00201  F863 39            INEER  RTS
00202
00203                          * INPUT CHARACTER WITHOUT PARITY BIT
00204
00205  F864 8D   E4   F84A INEEE  BSR    INCH      *GET CHAR, MASK IT, ECHO IT*
00206  F866 84   7F               ANDA   #$7F
00207
00208                          * OUTPUT CHARACTER
00209
00210  F868 34   12       OUTCH  PSHS   A,X        *DISPLAY A CHAR*
00211  F86A BE   F3E2             LDX    CPORT
00212  F86D A6   84       OUTCH1 LDA    0,X       [CPORT]
00213  F86F 85   02               BITA   #2        TDRE?
00214  F871 27   FA   F86D        BEQ    OUTCH1    NOT READY, TRY AGAIN
00215  F873 35   02               PULS   A
00216  F875 A7   01               STA    1,X       WRITE TO ACIA DATA REG
00217  F877 35   90               PULS   X,PC
00218
00219                          * PRINT C/R - L/F THEN DATA STRING
00220
00221  F879 8D   1B   F896 PSTR   BSR    PCRLF     *PRINT A STRING, CRLF FIRST*
00222  F87B 20   02   F87F        BRA    PDATA
00223
00224                          * PRINT DATA STRING
00225
00226  F87D 8D   E9   F868 PDATA0 BSR    OUTCH
00227  F87F A6   80       PDATA  LDA    ,X+       *PRINT A STRING*
00228  F881 81   04               CMPA   #4        DELIMITER?
00229  F883 26   F8   F87D        BNE    PDATA0
00230  F885 39                    RTS
00231
00232                          * CHECK FOR ABORT CHARACTER THEN C/R - L/F
00233
```

```
00234  F886 8D   B8   F840 ACRLF  BSR    INCHK    *CHECK ABORT; THEN CRLF*
00235  F888 27   0C   F896        BEQ    PCRLF
00236  F88A 8D   C2   F84E        BSR    INCH1    FETCH CHAR
00237  F88C 84   7F               ANDA   #$7F
00238  F88E 81   0D               CMPA   #$D
00239  F890 1027 0397 FC2B        LBEQ   RESTR1
00240  F894 8D   B4   F84A        BSR    INCH     GET A CHAR REF CONT
00241
00242                      * PRINT C/R - L/F
00243
00244  F896 34   10       PCRLF  PSHS   X        *PRINT A CRLF*
00245  F898 8E   FD46             LDX    #CRLFS
00246  F89B 8D   E2   F87F        BSR    PDATA
00247  F89D 35   90               PULS   X,PC
00248
00249                      * GET A BYTE FROM THE CONSOLE
00250
00251  F89F BD   FB40     GBYTE  JSR    FBYTE    *GET A BYTE FROM TERM*
00252  F8A2 24   2F   F8D3        BCC    HEXR     VALID HEX
00253  F8A4 7E   FC2B             JMP    RESTR1   NOT HEX
00254
00255                      * BUILD HEX ADDRESS
00256
00257  F8A7 8D   F6   F89F BADDR  BSR    GBYTE    *BUILD A 4 DIGIT HEX NUMBER*
00258  F8A9 32   7E               LEAS   -2,S     MAKE WORK SPACE
00259  F8AB A7   E4               STA    0,S
00260  F8AD 8D   F0   F89F        BSR    GBYTE
00261  F8AF A7   61               STA    1,S
00262  F8B1 35   90               PULS   X,PC     RETURN WITH ADDR IN X
00263
00264                      * BUILD HEX LIMIT ADDRESS
00265
00266  F8B3 8D   F2   F8A7 LIMITS BSR    BADDR    *GET 2 4 DIGIT NUMBERS*
00267  F8B5 1F   12               TFR    X,Y
00268  F8B7 8D   35   F8EE        BSR    OUTS
00269  F8B9 8D   EC   F8A7        BSR    BADDR
00270  F8BB 1E   12               EXG    X,Y
00271  F8BD 20   2F   F8EE        BRA    OUTS     RETURN WITH ADDRS IN X,Y
00272
00273                      * CONVERT INPUT CHARACTER TO HEXIDECIMAL
00274
00275  F8BF 80   30       HEXDIG SUBA   #'0      *CONVERTS HEX ASCII TO BINAR
00276  F8C1 25   10   F8D3        BCS    HEXR     SET CARRY ON ERROR
00277  F8C3 8B   E9               ADDA   #$E9
00278  F8C5 25   0C   F8D3        BCS    HEXR     ERROR
00279  F8C7 8B   06               ADDA   #6
00280  F8C9 2A   04   F8CF        BPL    HEXD1    OK
00281  F8CB 8B   07               ADDA   #7
00282  F8CD 25   04   F8D3        BCS    HEXR
00283  F8CF 8B   0A       HEXD1  ADDA   #10
00284  F8D1 1C   FE               ANDCC  #$FE     *CLC
00285  F8D3 39            HEXR   RTS
00286
00287                      * PRINT LEFT NIBBLE IN HEX
00288
00289  F8D4 44            OUTHL  LSRA             *OUTPUT HEX LEFT NIBBLE*
00290  F8D5 44                   LSRA
```

```
00291  F8D6 44                        LSRA
00292  F8D7 44                        LSRA
00293
00294                          * PRINT RIGHT NIBBLE IN HEX
00295
00296  F8D8 84   0F           OUTHR   ANDA    #$F       *OUTPUT HEX RIGHT NIBBLE*
00297  F8DA 8B   90                   ADDA    #$90
00298  F8DC 19                        DAA
00299  F8DD 89   40                   ADCA    #$40
00300  F8DF 19                        DAA
00301  F8E0 20   86   F868   OUTH1    BRA     OUTCH
00302
00303                          * PRINT 2 HEXIDECIMAL DIGITS
00304
00305  F8E2 A6   84           OUT2H   LDA     0,X       *OUTPUT 2 HEX DIGITS*
00306  F8E4 8D   EE   F8D4            BSR     OUTHL
00307  F8E6 A6   80                   LDA     ,X+
00308  F8E8 20   EE   F8D8            BRA     OUTHR
00309
00310                          * PRINT 4/2 HEXIDECIMAL DIGITS
00311
00312  F8EA 8D   F6   F8E2   OUT4HS   BSR     OUT2H
00313  F8EC 8D   F4   F8E2   OUT2HS   BSR     OUT2H
00314  F8EE 86   20           OUTS    LDA     #$20
00315  F8F0 20   EE   F8E0            BRA     OUTH1
00316
00317                          * INITIALIZE MEMORY WITH CONSTANT
00318
00319  F8F2 8D   BF   F8B3   INITM    BSR     LIMITS    *INIT MEM TO GIVEN VALUE*
00320  F8F4 31   21                   LEAY    1,Y       TWEEK FINISH ADDR
00321  F8F6 34   20                   PSHS    Y         SAVE IT
00322  F8F8 8D   A5   F89F            BSR     GBYTE     GET THE VALUE
00323  F8FA A7   80           INITM1  STA     ,X+       PUT INTO MEMORY
00324  F8FC AC   E4                   CMPX    0,S       DONE?
00325  F8FE 26   FA   F8FA            BNE     INITM1    NO
00326  F900 32   62                   LEAS    2,S       POP STACK
00327  F902 39                        RTS
00328
00329                          * JUMP INDIRECT TO ADDRESS
00330
00331  F903 8D   A2   F8A7   JUMPI    BSR     BADDR
00332  F905 6E   94                   JMP     [0,X]
00333
00334                          * JUMP TO MEMORY ADDRESS
00335
00336  F907 8D   9E   F8A7   JUMP     BSR     BADDR     *JUMP (SUBROUTINE) TO ADDR*
00337  F909 6E   84                   JMP     0,X       NOTE: AN RTS RETURNS TO CMD
00338
00339                          * GOTO MEMORY ADDRESS WITH REGISTERS
00340
00341  F90B 10FE F3E0         GO      LDS     SP        *GO TO USER'S JOB (ON STACK)
00342  F90F 3B                        RTI
00343
00344                          * DISPLAY REGISTERS
00345
00346  F910 8E   FD11         REGS    LDX     #REGTTL   *REGISTER DUMP*
00347  F913 BD   F879                 JSR     PSTR
```

```
00348  F916 BE   F3E0       REGSQ  LDX    SP         POINT TO STACK
00349  F919 8D   D1   F8EC         BSR    OUT2HS     CC
00350  F91B 6D   1F               TST    -1,X       CHECK 'E' BIT
00351  F91D 2B   09   F928         BMI    REGS0      IS SET; DISPLAY ALL
00352  F91F C6   18               LDB    #24
00353  F921 8D   CB   F8EE REGSS   BSR    OUTS       TYPE SPACES FOR REGISTER VAL
00354  F923 5A                    DECB
00355  F924 26   FB   F921         BNE    REGSS
00356  F926 20   0C   F934         BRA    REGST      OUTPUT PC ONLY
00357  F928 8D   C2   F8EC REGS0   BSR    OUT2HS     A
00358  F92A 8D   C0   F8EC         BSR    OUT2HS     B
00359  F92C 8D   BE   F8EC         BSR    OUT2HS     DP
00360  F92E 8D   BA   F8EA         BSR    OUT4HS     X
00361  F930 8D   B8   F8EA         BSR    OUT4HS     Y
00362  F932 8D   B6   F8EA         BSR    OUT4HS     U
00363  F934 8D   B4   F8EA REGST   BSR    OUT4HS     PC
00364  F936 8E   F3E0             LDX    #SP
00365  F939 8D   AF   F8EA         BSR    OUT4HS     SP
00366  F93B 8D   B1   F8EE         BSR    OUTS
00367  F93D 8E   FD4C             LDX    #CCLTRS    POINT TO CC LETTERS
00368  F940 C6   08               LDB    #8         FOR 8 BITS OF CC
00369  F942 A6   9F F3E0          LDA    [SP]       GET CC
00370  F946 34   02               PSHS   A          PUT INTO WORK AREA
00371  F948 68   E4       REGS2   ASL    0,S        MOVE BIT TO CARRY
00372  F94A A6   80               LDA    ,X+        GET CC LETTER
00373  F94C 25   02   F950         BCS    REGS3      THE BIT WAS 1
00374  F94E 86   20               LDA    #$20       IF BIT=0, SPACE
00375  F950 BD   F868     REGS3   JSR    OUTCH
00376  F953 8D   99   F8EE         BSR    OUTS
00377  F955 5A                    DECB
00378  F956 26   F0   F948         BNE    REGS2
00379  F958 35   82               PULS   A,PC       POP THE STACK
00380
00381                      *  NMI HANDLER
00382
00383  F95A 10FF F3E0     SNMI    STS    SP
00384  F95E 20   1B   F97B         BRA    SSWI2
00385
00386                      * SOFTWARE INTERRUPT HANDLER
00387
00388  F960 10FF F3E0     SSWI    STS    SP         *DEFAULT SWI (BREAKPOINT) HA
00389  F964 10AE 6A               LDY    10,S       GET PC
00390  F967 31   3F               LEAY   -1,Y       ADJUST
00391  F969 10AF 6A               STY    10,S       ADJUSTED PC TO STACK
00392  F96C 8E   F3EE             LDX    #BPTBL
00393  F96F 10AC 84       SSWI1   CMPY   0,X        PC MATCH TABLE?
00394  F972 27   0C   F980         BEQ    SSWI3      YES
00395  F974 30   03               LEAX   3,X        NO, TRY NEXT ENTRY
00396  F976 8C   F400             CMPX   #BPTBLE    END OF TABLE?
00397  F979 26   F4   F96F         BNE    SSWI1      NO
00398  F97B 8D   93   F910 SSWI2   BSR    REGS       YES, WASN'T BREAKPOINT
00399  F97D 7E   FC2F             JMP    CMD
00400
00401
00402  F980 A6   02       SSWI3   LDA    2,X        GET OPCODE
00403  F982 A7   A4               STA    0,Y        RESTORE OPCODE
00404  F984 86   FF               LDA    #$FF
```

```
00405  F986 A7   84              STA    0,X      CLEAR ADDR IN TABLE
00406  F988 A7   01              STA    1,X
00407  F98A 86   23              LDA    #'#      FLAG THE REGISTER DUMP
00408  F98C BD   F868            JSR    OUTCH    AS A BREAKPOINT
00409  F98F 20   EA   F97B       BRA    SSWI2    DISPLAY REGISTERS
00410
00411                      * REGISTER EXAMINE & CHANGE SECTION
00412
00413                      * A REGISTER
00414
00415  F991 BE   F3E0     AREG   LDX    SP
00416  F994 30   01              LEAX   1,X
00417  F996 20   0C   F9A4       BRA    RMODV
00418
00419                      * B REGISTER
00420
00421  F998 BE   F3E0     BREG   LDX    SP
00422  F99B 30   02              LEAX   2,X
00423  F99D 20   05   F9A4       BRA    RMODV
00424
00425                      * DP REGISTER
00426
00427  F99F BE   F3E0     DPREG  LDX    SP
00428  F9A2 30   03              LEAX   3,X
00429
00430  F9A4 6D   9F F3E0  RMODV  TST    [SP]     CHECK 'E' BIT
00431  F9A8 2B   1E   F9C8       BMI    RMOD     SET; DO THE REGISTER
00432  F9AA 39                   RTS             NOT SET; DO NOTHING
00433
00434                      * X REGISTER
00435
00436  F9AB BE   F3E0     XREG   LDX    SP
00437  F9AE 30   04              LEAX   4,X
00438  F9B0 20   0C   F9BE       BRA    RMOD2V
00439
00440                      * Y REGISTER
00441
00442  F9B2 BE   F3E0     YREG   LDX    SP
00443  F9B5 30   06              LEAX   6,X
00444  F9B7 20   05   F9BE       BRA    RMOD2V
00445
00446                      * U REGISTER
00447
00448  F9B9 BE   F3E0     UREG   LDX    SP
00449  F9BC 30   08              LEAX   8,X
00450
00451  F9BE 6D   9F F3E0  RMOD2V TST    [SP]     CHECK 'E' BIT
00452  F9C2 2B   1A   F9DE       BMI    RMOD2    SET; DO THE REGISTER
00453  F9C4 39                   RTS             NOT SET; DO NOTHING
00454
00455                      * CC REGISTER
00456
00457  F9C5 BE   F3E0     CREG   LDX    SP
00458
00459  F9C8 BD   F8EC     RMOD   JSR    OUT2HS   DISPLAY REG VALUE FROM STACK
00460  F9CB BD   F89F            JSR    GBYTE    GET NEW VALUE
00461  F9CE A7   1F              STA    -1,X     PUT IT INTO STACK
```

```
00462  F9D0 39                    RTS
00463
00464                     * PC REGISTER
00465
00466  F9D1 BE   F3E0     PCREG   LDX    SP
00467  F9D4 30   01               LEAX   1,X
00468  F9D6 6D   9F F3E0          TST    [SP]       CHECK 'E' BIT
00469  F9DA 2A   02   F9DE        BPL    RMOD2      CLEAR; DO PC AFTER CC
00470  F9DC 30   09               LEAX   9,X        SET; DO NORMAL
00471
00472  F9DE BD   F8EA     RMOD2   JSR    OUT4HS     DISPLAY 2 BYTE REG.
00473  F9E1 31   1E               LEAY   -2,X       SAVE POINTER TO REG.
00474  F9E3 BD   F8A7             JSR    BADDR
00475  F9E6 AF   A4               STX    0,Y        PUT IT INTO STACK
00476  F9E8 39           RMODX   RTS
00477
00478                     * MEMORY EXAMINE AND CHANGE
00479
00480  F9E9 BD   F8A7     EXAM    JSR    BADDR      *MEMORY EXAMINE & CHANGE*
00481  F9EC BF   F3E4     EXAM1   STX    MEM        SAVE ADDR FOR RXAM
00482  F9EF BD   F896     RXAM    JSR    PCRLF      *REOPEN LAST ADDR*
00483  F9F2 8E   F3E4             LDX    #MEM
00484  F9F5 BD   F8EA             JSR    OUT4HS     DISPLAY IT
00485  F9F8 BE   F3E4             LDX    MEM
00486  F9FB BD   F8EC             JSR    OUT2HS     DISPLAY THE DATA (BUMP X)
00487  F9FE BD   F857             JSR    INEE       GET A CHAR
00488  FA01 81   0D               CMPA   #$D        WAS IT CR?
00489  FA03 27   E3   F9E8        BEQ    RMODX      YES
00490  FA05 81   55               CMPA   #'U
00491  FA07 26   04   FA0D        BNE    EXAM4
00492  FA09 30   1E               LEAX   -2,X       BACK UP ONE LOC
00493  FA0B 20   DF   F9EC        BRA    EXAM1
00494  FA0D 81   3A       EXAM4   CMPA   #':
00495  FA0F 26   04   FA15        BNE    EXAM5
00496  FA11 AE   1F               LDX    -1,X       OPEN INDIRECT M,M+1
00497  FA13 20   D7   F9EC        BRA    EXAM1
00498  FA15 81   40       EXAM5   CMPA   #'@
00499  FA17 26   04   FA1D        BNE    EXAM6
00500  FA19 AE   1E               LDX    -2,X       OPEN INDIRECT M-1,M
00501  FA1B 20   CF   F9EC        BRA    EXAM1
00502  FA1D BD   FB43     EXAM6   JSR    FBYTA      GET A BYTE
00503  FA20 25   CA   F9EC        BCS    EXAM1      NOT HEX; DO NEXT
00504  FA22 BE   F3E4             LDX    MEM
00505  FA25 A7   80               STA    ,X+        PUT NEW VALUE INTO MEM
00506  FA27 A1   1F               CMPA   -1,X
00507  FA29 27   C1   F9EC        BEQ    EXAM1      STORED OK
00508  FA2B BD   F8EE             JSR    OUTS
00509  FA2E 86   3F               LDA    #'?
00510  FA30 BD   F868             JSR    OUTCH
00511  FA33 20   B7   F9EC        BRA    EXAM1
00512
00513
00514                     * FIND A STRING IN MEMORY
00515
00516  FA35 BD   F8B3     FINDST  JSR    >LIMITS    GET BOUNDS
00517  FA38 34   20               PSHS   Y          SAVE FROM ADDR
00518  FA3A 32   7C               LEAS   -4,S       MAKE WORKSPACE
```

```
00519  FA3C 1F   43              TFR   S,U       SAVE POINTER
00520  FA3E 6F   E2              CLR   ,-S       INIT COUNTER
00521  FA40 6F   E2              CLR   ,-S
00522  FA42 BD   FB40            JSR   FBYTE     GET A BYTE
00523  FA45 25   39    FA80      BCS   FINDS9    INVALID, QUIT
00524  FA47 A7   C0      FINDS1  STA   ,U+       SAVE
00525  FA49 BD   FB3D            JSR   NBYTE     GET NEXT BYTE
00526  FA4C 25   0A    FA58      BCS   FINDS2    INVALID, START MATCHING
00527  FA4E 6C   E4              INC   ,S        SET COUNT
00528  FA50 E6   E4              LDB   ,S
00529  FA52 C1   03              CMPB  #$03      FULL?
00530  FA54 25   F1    FA47      BLO   FINDS1    NO
00531  FA56 A7   C4              STA   ,U        YES, SAVE LAST BYTE
00532  FA58 E6   E4      FINDS2  LDB   ,S        MOVE COUNT
00533  FA5A E7   61              STB   1,S       TO TEMP
00534  FA5C 1F   13              TFR   X,U       INIT TEMP POINTERS
00535  FA5E 1F   42              TFR   S,Y
00536  FA60 31   22              LEAY  2,Y       POINT TO 1ST BYTE
00537  FA62 A6   C0      FINDS3  LDA   ,U+
00538  FA64 A1   A0              CMPA  ,Y+       BYTES MATCH?
00539  FA66 26   10    FA78      BNE   FINDS6    NO
00540  FA68 6A   61              DEC   1,S       UPDATE COUNT
00541  FA6A 2A   F6    FA62      BPL   FINDS3    MORE TO CHECK
00542  FA6C BD   F886            JSR   >ACRLF    GOT A MATCH; PRINT IT
00543  FA6F 34   10              PSHS  X
00544  FA71 1F   41              TFR   S,X       POINT TO ADDR
00545  FA73 BD   F8EA            JSR   >OUT4HS
00546  FA76 35   10              PULS  X
00547  FA78 AC   66      FINDS6  CMPX  6,S       DONE?
00548  FA7A 24   04    FA80      BHS   FINDS9    YES
00549  FA7C 30   01              LEAX  1,X
00550  FA7E 20   D8    FA58      BRA   FINDS2
00551
00552  FA80 32   68      FINDS9  LEAS  8,S       FIX STACK
00553  FA82 39           FINDSR  RTS
00554
00555                     * DUMP MAPTBL
00556
00557  FA83 BD   F896    DUMPMT  JSR   PCRLF
00558  FA86 5F                   CLRB
00559  FA87 8D   23    FAAC DUMPT1 BSR   JOUTS
00560  FA89 1F   98              TFR   B,A
00561  FA8B 8D   1C    FAA9      BSR   OUT1HS
00562  FA8D 5C                   INCB
00563  FA8E C1   10              CMPB  #16
00564  FA90 26   F5    FA87      BNE   DUMPT1
00565
00566  FA92 BD   F896            JSR   PCRLF
00567  FA95 C6   10              LDB   #16
00568  FA97 8E   F3D0            LDX   #MAPTBL
00569  FA9A A6   84      DUMPT2  LDA   0,X
00570  FA9C BD   F8D4            JSR   OUTHL
00571  FA9F A6   80              LDA   ,X+
00572  FAA1 88   00              EORA  #FLIPBT
00573  FAA3 8D   04    FAA9      BSR   OUT1HS
00574  FAA5 5A                   DECB
00575  FAA6 26   F2    FA9A      BNE   DUMPT2
```

```
00576  FAA8 39                        RTS
00577
00578
00579  FAA9 BD   F8D8       OUT1HS JSR    OUTHR
00580  FAAC 7E   F8EE       JOUTS  JMP    OUTS
00581
00582
00583
00584                       * ASCII MEMORY DUMP
00585
00586  FAAF BD   F8B3       DUMPM  JSR    LIMITS   *DISPLAY A FORMATTED MEMORY
00587  FAB2 1F   10                TFR    X,D      FORCE START & FINISH
00588  FAB4 C4   F0                ANDB   #$F0     TO A MOD 16 BOUNDARY
00589  FAB6 1F   01                TFR    D,X
00590  FAB8 1F   20                TFR    Y,D
00591  FABA CA   0F                ORB    #$0F
00592  FABC 1F   02                TFR    D,Y
00593  FABE 31   21                LEAY   1,Y      INC FA
00594  FAC0 34   20                PSHS   Y        FA TO STACK
00595  FAC2 BD   F886       DUMPM1 JSR    ACRLF
00596  FAC5 34   10                PSHS   X        SAVE POINTER FOR ASCII OUTPU
00597  FAC7 1F   41                TFR    S,X
00598  FAC9 BD   F8EA              JSR    OUT4HS   DISPLAY ADDR
00599  FACC AE   E4                LDX    0,S
00600  FACE C6   10                LDB    #16      DISPLAY 16 BYTES
00601  FAD0 BD   F8EC       DUMPM2 JSR    OUT2HS   DISPLAY DATA
00602  FAD3 5A                     DECB
00603  FAD4 26   FA   FAD0         BNE    DUMPM2
00604  FAD6 35   10                PULS   X        RECOVER POINTER
00605  FAD8 BD   F8EE              JSR    OUTS
00606  FADB C6   10                LDB    #16      DISPLAY DATA AS ASCII
00607  FADD A6   80         DUMPM3 LDA    ,X+
00608  FADF 84   7F                ANDA   #$7F
00609  FAE1 81   7F                CMPA   #$7F
00610  FAE3 27   04   FAE9         BEQ    DUMPM4
00611  FAE5 81   20                CMPA   #$20
00612  FAE7 2C   02   FAEB         BGE    DUMPM5
00613  FAE9 86   2E         DUMPM4 LDA    #'.      NO, USE DOT
00614  FAEB BD   F868       DUMPM5 JSR    OUTCH
00615  FAEE 5A                     DECB
00616  FAEF 26   EC   FADD         BNE    DUMPM3
00617  FAF1 AC   E4                CMPX   0,S      REACHED FA?
00618  FAF3 26   CD   FAC2         BNE    DUMPM1   NO
00619  FAF5 35   90                PULS   X,PC     POP STACK & RETURN
00620  FAF7 39                     RTS
00621
00622                       * BREAKPOINT HANDLER
00623
00624  FAF8 108E F3EE       BREAK  LDY    #BPTBL   *SET BREAKPOINT*
00625  FAFC EC   A4         BRK1   LDD    0,Y      CHECK ADDR IN TABLE
00626  FAFE C3   0001              ADDD   #1
00627  FB01 27   0E   FB11         BEQ    BRK4     EMPTY SPOT; USE IT
00628  FB03 31   23                LEAY   3,Y      POINT TO NEXT ENTRY
00629  FB05 108C F400              CMPY   #BPTBLE
00630  FB09 25   F1   FAFC         BLO    BRK1     TRY NEXT
00631  FB0B 8E   FD54              LDX    #FULL    NONE LEFT
00632  FB0E 7E   F87F              JMP    PDATA
```

```
00633  FB11 BD   F8A7       BRK4   JSR    BADDR    GET BREAK ADDR
00634  FB14 AF   A4                STX    0,Y      PUT INTO TABLE
00635  FB16 A6   84                LDA    0,X      GET OPCODE
00636  FB18 A7   22                STA    2,Y      SAVE IN TABLE
00637  FB1A 86   3F                LDA    #$3F
00638  FB1C A7   84                STA    0,X      PUT SWI AT ADDR
00639  FB1E 39                     RTS
00640
00641                       * CLEAR BREAKPOINTS
00642
00643  FB1F 8E   F3EE       UNBRK  LDX    #BPTBL   *CLEAR ALL BREAKPOINTS*
00644  FB22 86   FF                LDA    #$FF
00645  FB24 10AE 84         UNBRK1 LDY    0,X      GET ADDR
00646  FB27 108C FFFF              CMPY   #$FFFF   EMPTY?
00647  FB2B 27   04   FB31         BEQ    UNBRK2   YES
00648  FB2D E6   02                LDB    2,X      GET OPCODE
00649  FB2F E7   A4                STB    0,Y      PUT INTO MEMORY
00650  FB31 A7   80         UNBRK2 STA    ,X+      CLEAR THE ENTRY
00651  FB33 A7   80                STA    ,X+
00652  FB35 A7   80                STA    ,X+
00653  FB37 8C   F400              CMPX   #BPTBLE  END OF TABLE?
00654  FB3A 26   E8   FB24         BNE    UNBRK1   NO
00655  FB3C 39                     RTS
00656
00657                       * GET A HEXIDECIMAL DIGIT
00658
00659  FB3D BD   F8EE       NBYTE  JSR    OUTS
00660  FB40 BD   F857       FBYTE  JSR    INEE     GET A BYTE (NO RESTRT)
00661  FB43 BD   F8BF       FBYTA  JSR    HEXDIG   IS IT GOOD ?
00662  FB46 25   10   FB58         BCS    FBYTS    NO
00663  FB48 48                     ASLA
00664  FB49 48                     ASLA
00665  FB4A 48                     ASLA
00666  FB4B 48                     ASLA
00667  FB4C 34   02                PSHS   A        SAVE 1ST NIBBLE
00668  FB4E BD   F857              JSR    INEE
00669  FB51 BD   F8BF              JSR    HEXDIG   IS IT GOOD ?
00670  FB54 25   03   FB59         BCS    FBYTR    NO
00671  FB56 AB   E0                ADDA   0,S+     COMBINE NIBBLES
00672  FB58 39              FBYTS  RTS              RETURN WITH CARRY CLEAR
00673
00674  FB59 35   82         FBYTR  PULS   A,PC     CARRY SET ON ERR
00675
00676                       * FLEX DD BOOT
00677  FB5B AD   9F F818    BOOTF  JSR    [FBOOT]
00678  FB5F 20   0A   FB6B         BRA    BOOTE
00679
00680                       * HARD DISK BOOT
00681  FB61 AD   9F F81A    BOOTH  JSR    [HBOOT]
00682  FB65 20   04   FB6B         BRA    BOOTE
00683
00684                       * DOS DBL DEN BOOT
00685  FB67 AD   9F F816    BOOTD  JSR    [DBOOT]
00686  FB6B 34   14         BOOTE  PSHS   B,X      ERR CODE, T & S
00687  FB6D 8E   FD49              LDX    #ERRMSG
00688  FB70 BD   F879              JSR    PSTR
00689  FB73 35   14                PULS   B,X
```

```
00690  FB75 3F                        SWI
00691
00692                          * LOAD READ ADDRESS
00693
00694  FB76 34   36            LRA    PSHS   A,B,X,Y  *LOAD REAL ADDRESS*
00695  FB78 A6   62                   LDA    2,S       GET MSB OF X
00696  FB7A 44                        LSRA             GET MS NIBBLE OF X (A12-A15)
00697  FB7B 44                        LSRA
00698  FB7C 44                        LSRA
00699  FB7D 44                        LSRA
00700  FB7E 108E F3D0                 LDY    #MAPTBL
00701  FB82 E6   A6                   LDB    A,Y       GET MAP FOR THIS ADDR INTO P
00702  FB84 54                        LSRB             GET MS NIBBLE OF B (A16-A19
00703  FB85 54                        LSRB
00704  FB86 54                        LSRB
00705  FB87 54                        LSRB
00706  FB88 E7   E4                   STB    0,S       PUT INTO A'S POSITION ON STA
00707  FB8A E6   A6                   LDB    A,Y       GET MAP AGAIN
00708  FB8C      12                   FCB    LRAFIX
00709  FB8D 58                        ASLB             GET LS NIBBLE, COMPLEMENTED
00710  FB8E 58                        ASLB             (THIS IS A12-A15, TRANSLATED
00711  FB8F 58                        ASLB
00712  FB90 58                        ASLB
00713  FB91 A6   62                   LDA    2,S       GET MS BYTE OF X
00714  FB93 84   0F                   ANDA   #$F       KILL A12-A15
00715  FB95 A7   62                   STA    2,S       PUT IT BACK
00716  FB97 EA   62                   ORB    2,S       SET MS NIBBLE TO A12-A15 TRA
00717  FB99 E7   62                   STB    2,S       PUT IT BACK INTO X'S POSITIO
00718  FB9B 35   B6                   PULS   A,B,X,Y,PC RETURN MODIFIED A,X
00719
00720
00721  FB9D BD   F8A7          TEXT   JSR    >BADDR
00722  FBA0 BD   F896                 JSR    >PCRLF
00723  FBA3 BD   F864          TEXT1  JSR    >INEEE
00724  FBA6 81   04                   CMPA   #$04
00725  FBA8 27   04    FBAE           BEQ    TEXTR
00726  FBAA A7   80                   STA    ,X+
00727  FBAC 20   F5    FBA3           BRA    TEXT1
00728
00729  FBAE 39                 TEXTR  RTS
00730
00731                          * MONITOR INITIALIZATION
00732
00733  FBAF 1F   41            MON0   TFR    S,X
00734  FBB1 C6   0C                   LDB    #12       INIT STACK REGS TO 0
00735  FBB3 6F   80            MON20  CLR    ,X+
00736  FBB5 5A                        DECB
00737  FBB6 26   FB    FBB3           BNE    MON20
00738  FBB8 C6   D0                   LDB    #$D0      INIT CC REG TO $D0 TO ALLOW
00739  FBBA E7   E4                   STB    0,S       ACCESS TO ALL REGISTERS, INT
00740  FBBC 8E   E004                 LDX    #ACIAS    SET UP PORT ADDRESS
00741  FBBF BF   F3E2                 STX    CPORT
00742  FBC2 C6   03                   LDB    #3        INIT ACIA
00743  FBC4 E7   84                   STB    0,X
00744  FBC6 8E   FF05                 LDX    #DUMRTI
00745  FBC9 BF   F3C2                 STX    VSWI3     SET VECTORS TO DEFAULTS
00746  FBCC BF   F3C4                 STX    VSWI2
```

```
00747  FBCF BF   F3C6             STX   VFIRQ
00748  FBD2 BF   F3C8             STX   VIRQ
00749  FBD5 8E   F95A             LDX   #SNMI
00750  FBD8 BF   F3BE             STX   VNMI
00751  FBDB 8E   F960             LDX   #SSWI
00752  FBDE BF   F3CA             STX   VSWI
00753  FBE1 8E   FFFF             LDX   #-1
00754  FBE4 BF   F3CC             STX   SVCO
00755  FBE7 BF   F3CE             STX   SVCL
00756  FBEA 8E   F3EE             LDX   #BPTBL    INIT BREAKPOINT TABLE
00757  FBED C6   FF               LDB   #$FF
00758  FBEF E7   80       MON25   STB   ,X+
00759  FBF1 8C   F400             CMPX  #BPTBLE
00760  FBF4 26   F9   FBEF        BNE   MON25
00761
00762                     * COLD START BANNER
00763
00764  FBF6 86   15               LDA   #ACINZ
00765  FBF8 A7   9F F3E2          STA   [CPORT]
00766  FBFC 8E   FD59             LDX   #BANNER
00767  FBFF BD   F879             JSR   PSTR      INDNTIFY SELF
00768
00769                     * TOT UP MEM
00770
00771  FC02 8E   F3DE             LDX   #MAPTBL+$E
00772  FC05 4F                    CLRA
00773  FC06 E6   82       SUM     LDB   ,-X
00774  FC08 C8   F0               EORB  #FLIPB2
00775  FC0A 27   03   FC0F        BEQ   SUM1
00776  FC0C 8B   04               ADDA  #4
00777  FC0E 19                    DAA
00778  FC0F 8C   F3D0     SUM1    CMPX  #MAPTBL
00779  FC12 26   F2   FC06        BNE   SUM
00780  FC14 34   02               PSHS  A
00781  FC16 1F   41               TFR   S,X
00782  FC18 BD   F8E2             JSR   OUT2H
00783  FC1B 86   4B               LDA   #'K
00784  FC1D BD   F868             JSR   OUTCH
00785
00786
00787                     * RESTART ENTRY
00788
00789  FC20 10CE F3B2     RESTRT  LDS   #STACK    *RESTART*
00790  FC24 10FF F3E0             STS   SP        RESET STACK
00791
00792  FC28 BD   FB1F             JSR   UNBRK     REMOVE BREAKPOINTS
00793
00794                     * COMMAND INPUT
00795
00796  FC2B 10FE F3E0     RESTR1  LDS   SP
00797  FC2F 86   15       CMD     LDA   #ACINZ    *MONITOR SOFT START*
00798  FC31 A7   9F F3E2          STA   [CPORT]   ACIA SETUP
00799  FC35 8E   FD0F             LDX   #PROMPT
00800  FC38 BD   F879             JSR   PSTR
00801  FC3B 8E   FCB0             LDX   #CTABL-3
00802  FC3E BD   F857     CMD01   JSR   INEE      GET A COMMAND LETTER
00803  FC41 30   03       CMD0    LEAX  3,X       SKIP OVER THE ENTRY
```

```
00804  FC43 6D   84              TST   0,X       FIND COMMAND IN TABLE?
00805  FC45 27   E8   FC2F      BEQ   CMD       END OF LIST
00806  FC47 A1   84              CMPA  0,X       MATCH?
00807  FC49 26   F6   FC41      BNE   CMD0      NO
00808  FC4B BD   F8EE            JSR   OUTS      ACKNOWLEDGE VALID COMMAND
00809  FC4E AD   98 01           JSR   [1,X]     DO THE ROUTINE
00810  FC51 20   DC   FC2F      BRA   CMD
00811
00812  FC53 32   62        SLOPT LEAS  2,S       WASTE RETURN
00813  FC55 8E   FCF9            LDX   #SLTBL-3  SWITCH TABLES
00814  FC58 20   E4   FC3E      BRA   CMD01     CONTINUE SEARCH
00815
00816                       * LIMIT CHECK FOR EXBLK
00817
00818  FC5A BD   F857      ONEDIG JSR  INEE
00819  FC5D BD   F8BF            JSR   HEXDIG
00820  FC60 25   BE   FC20      BCS   RESTRT
00821  FC62 81   0E              CMPA  #$E       I/O PAGE?
00822  FC64 27   02   FC68      BEQ   LIMCHX
00823  FC66 81   0F              CMPA  #$F       OUR PAGE?
00824  FC68 39             LIMCHX RTS
00825
00826                       * EXBLK - MAPPED BLOCK EXCHANGE
00827
00828  FC69 8D   EF   FC5A EXBLK  BSR   ONEDIG
00829  FC6B 27   B3   FC20      BEQ   RESTRT    BAD
00830  FC6D 1F   89              TFR   A,B
00831  FC6F BD   F8EE            JSR   OUTS
00832  FC72 8D   E6   FC5A      BSR   ONEDIG
00833  FC74 27   AA   FC20      BEQ   RESTRT    BAD
00834  FC76 34   06              PSHS  A,B
00835  FC78 A1   61              CMPA  1,S       BOTH SAME?
00836  FC7A 27   A4   FC20      BEQ   RESTRT
00837  FC7C 8E   F3D0            LDX   #MAPTBL
00838  FC7F 1F   12              TFR   X,Y
00839  FC81 30   86              LEAX  A,X
00840  FC83 31   A5              LEAY  B,Y
00841  FC85 A6   84              LDA   0,X
00842  FC87 E6   A4              LDB   0,Y
00843  FC89 A7   A4              STA   0,Y
00844  FC8B E7   84              STB   0,X
00845  FC8D EC   E1              LDD   S++       RECOVER BLOCK NUMS
00846  FC8F C1   0D              CMPB  #$D       SCRATCH?
00847  FC91 27   06   FC99      BEQ   EXBLK2
00848  FC93 1E   89              EXG   A,B
00849  FC95 C1   0D              CMPB  #$D       SCRATCH
00850  FC97 26   17   FCB0      BNE   EXBLK4
00851  FC99 C6   00        EXBLK2 LDB  #0        WAS MAPTBL*256/256
00852  FC9B 48                   ASLA            MOVE HIGH ORDER INTO POSITIO
00853  FC9C 48                   ASLA
00854  FC9D 48                   ASLA
00855  FC9E 48                   ASLA
00856  FC9F 8A   0F              ORA   #$F       MAKE $XF
00857  FCA1 1F   01              TFR   D,X
00858  FCA3 108E DF00            LDY   #$DF00    WAS STACK-10
00859  FCA7 C6   00              LDB   #0        WAS BPTBLE-(STACK-10)
00860  FCA9 A6   A0        EXBLK3 LDA  Y+
```

```
00861  FCAB A7   80              STA    X+
00862  FCAD 5A                   DECB
00863  FCAE 26   F9   FCA9       BNE    EXBLK3
00864
00865  FCB0 7E   FFC0     EXBLK4 JMP    MOVTBL   MOVE TBL & RETURN TO CMD
00866
```

```
00869                          *
00870                          *
00871                          *       COMMAND TABLE
00872                          *
00873  FCB3      2F            CTABL   FCB     '/
00874  FCB4      FC53                  FDB     SLOPT
00875  FCB6      52                    FCB     'R
00876  FCB7      F910                  FDB     REGS
00877  FCB9      53                    FCB     'S
00878  FCBA      FA35                  FDB     FINDST
00879  FCBC      4D                    FCB     'M
00880  FCBD      F9E9                  FDB     EXAM
00881  FCBF      47                    FCB     'G
00882  FCC0      F90B                  FDB     GO
00883  FCC2      4A                    FCB     'J
00884  FCC3      F907                  FDB     JUMP
00885  FCC5      49                    FCB     'I
00886  FCC6      F8F2                  FDB     INITM
00887  FCC8      51                    FCB     'Q
00888  FCC9      FE1C                  FDB     HSTART
00889  FCCB      57                    FCB     'W
00890  FCCC      D283                  FDB     ZWARMS
00891  FCCE      4B                    FCB     'K
00892  FCCF      FAF8                  FDB     BREAK
00893  FCD1      41                    FCB     'A
00894  FCD2      F991                  FDB     AREG
00895  FCD4      42                    FCB     'B
00896  FCD5      F998                  FDB     BREG
00897  FCD7      43                    FCB     'C
00898  FCD8      F9C5                  FDB     CREG
00899  FCDA      44                    FCB     'D
00900  FCDB      F99F                  FDB     DPREG
00901  FCDD      58                    FCB     'X
00902  FCDE      F9AB                  FDB     XREG
00903  FCE0      59                    FCB     'Y
00904  FCE1      F9B2                  FDB     YREG
00905  FCE3      55                    FCB     'U
00906  FCE4      F9B9                  FDB     UREG
00907  FCE6      50                    FCB     'P
00908  FCE7      F9D1                  FDB     PCREG
00909  FCE9      45                    FCB     'E
00910  FCEA      FC69                  FDB     EXBLK
00911  FCEC      5A                    FCB     'Z
00912  FCED      FAAF                  FDB     DUMPM
00913  FCEF      54                    FCB     'T
00914  FCF0      FB9D                  FDB     TEXT
00915  FCF2      2E                    FCB     '.
00916  FCF3      F9EF                  FDB     RXAM
00917  FCF5      0B                    FCB     'K-'@
00918  FCF6      FB1F                  FDB     UNBRK
00919  FCF8      46                    FCB     'F
00920  FCF9      CD03                  FDB     $CD03
00921  FCFB      00                    FCB     0
```

```
00923  FCFC  5A        SLTBL  FCB   'Z
00924  FCFD  FA83             FDB   DUMPMT
00925  FCFF  4A               FCB   'J
00926  FD00  F903             FDB   JUMPI
00927  FD02  51               FCB   'Q
00928  FD03  FB67             FDB   BOOTD
00929  FD05  52               FCB   'R
00930  FD06  FC20             FDB   RESTRT
00931  FD08  46               FCB   'F
00932  FD09  FB5B             FDB   BOOTF
00933  FD0B  48               FCB   'H
00934  FD0C  FB61             FDB   BOOTH
00935  FD0E  00               FCB   0
00937                *
00938                *      STRINGS
00939                *
00940  FD0F  2A        PROMPT FCB   '*,4
00941  FD11  43        REGTTL FCC   'CC A  B  DP X    Y    U    PC   SP'
00942  FD33  20               FCC   '    E F H I N Z V C'
00943  FD46  0D        CRLFS  FCB   $0D,$0A,4
00944  FD49  25        ERRMSG FCC   '%'
00945  FD4B  04               FCB   4
00946  FD4C  45        CCLTRS FCC   'EFHINZVC'
00947  FD54  46        FULL   FCC   'FULL'
00948  FD58  04               FCB   4
00949  FD59  4D        BANNER FCC   'MON'
00950  FD5C  35               FCB   VERSON,'9,DVRVER,$D,$A,4
00951
00952                FD62      MONE   EQU   *
00953
00954
00955                0000      SPARE  EQU   $FD62-MONE
```

```
00958                          *
00959                          * USE NON-MAPPPED MEMORY FOR THESE
00960                          *
00961
00962  FF00                           ORG     $FF00
00963
00964  FF00    01                     FCB     RELEAS
00965  FF01    09                     FCB     MONTH,DAY,YEAR
00966  FF04    05                     FCB     EDIT
00967
00968  FF05 3B                DUMRTI  RTI
00969
00970  FF06 6E  9F F3CA       ISWI    JMP     [VSWI]    *HERE ON SWI*
00971
00972  FF0A 6E  9F F3C4       ISWI2   JMP     [VSWI2]   *HERE ON SWI2*
00973
00974  FF0E 1F  43            ISWI3   TFR     S,U       *COME HERE IN SWI3*
00975  FF10 AE  6A                    LDX     10,S      GET PC FROM STACK
00976  FF12 E6  80                    LDB     0,X+      GET TABLE OFFSET
00977  FF14 AF  6A                    STX     10,S      PUT ADJUSTED PC BACK
00978  FF16 BE  F3CC                  LDX     SVCO      CHECK FOR SERVICE TABLE
00979  FF19 8C  FFFF                  CMPX    #-1
00980  FF1C 26  08   FF26             BNE     ISWI31    SERVICE HAS BEEN SET
00981  FF1E 37  1F            ISWI30  PULU    CC,A,B,DP,X RESTORE MOST
00982  FF20 EE  68                    LDU     8,S       RESTORE U
00983  FF22 6E  9F F3C2               JMP     [VSWI3]   USER SWI3
00984  FF26 4F                ISWI31  CLRA
00985  FF27 58                        ASLB              D=(B)*2
00986  FF28 49                        ROLA
00987  FF29 30  8B                    LEAX    D,X       POINT INTO SVC TABLE
00988  FF2B RC  F3CE                  CMPX    SVCL      IS IN THE TABLE ?
00989  FF2E 22  EE   FF1E             BHI     ISWI30    TOO HIGH; DEFAULT TO USER
00990  FF30 AE  84                    LDX     0,X       GET ADDR FROM TABLE
00991  FF32 34  10                    PSHS    X         SAVE POINTER
00992  FF34 EC  63                    LDD     3,S       RESTORE A,B FROM STACK
00993  FF36 AE  66                    LDX     6,S       RESTORE X FROM STACK
00994  FF38 6E  F1                    JMP     [0,S++]   JUMP @ POINTER
00995
00996  FF3A 6E  9F F3BE       INMI    JMP     [VNMI]    *HERE ON NMI*
00997
00998  FF3E 6E  9F F3C6       IFIRQ   JMP     [VFIRQ]   *HERE ON FIRQ*
00999
01000  FF42 6E  9F F3C8       IIRQ    JMP     [VIRQ]    *HERE ON IRQ*
01001
01002  FF46 6E  9F F3C0       IRESVD  JMP     [VRESVD]  *HERE ON IRESVD*
01003           FF4A           INTEND  EQU     *         END OF INTERRUPT HANDLERS
```

```
01006                          *
01007                          *        VECTORS
01008                          *
01009  FFF0                            ORG     $FFF0
01010  FFF0    FF46                    FDB     IRESVD   (RESERVED VECTOR)
01011  FFF2    FF0E                    FDB     ISWI3
01012  FFF4    FF0A                    FDB     ISWI2
01013  FFF6    FF3E                    FDB     IFIRQ
01014  FFF8    FF42                    FDB     IIRQ
01015  FFFA    FF06                    FDB     ISWI
01016  FFFC    FF3A                    FDB     INMI
01017  FFFE    FF4A                    FDB     MONIT    RESET
01018
```

```
01021  FF4A                          ORG    INTEND
01022                        *
01023                        * COLD START
01024                        * MUST BE LOCATED AT OR ABOVE $FF00
01025                        * WHEN USING SSB'S SCB=69 CPU BOARD, BECAUSE
01026                        * THE MEMORY MAPPER IS NOT INITIALIZED TO
01027                        * ANY PARTICULAR STATE ON POWER UP, BUT THE
01028                        * MAPPER ALWAYS MAPS $FFXX INTO $FFXX,
01029                        * REGARDLESS OF THE MAP RAM CONTENTS.
01030                        *
01031                        *  7-12-83 USE ONLY PARTITION 0
01032
01033  FF4A 7F   F710     MONIT  CLR    CLOCK+16 *COME HERE ON RESET*
01034  FF4D 7D   F711            TST    CLOCK+17 CLEAR ANY INTERRUPTS
01035
01036  FF50 8E   FFF0           LDX    #MAPRAM
01037  FF53 CE   F3D0           LDU    #MAPTBL
01038  FF56 108E FEFF           LDY    #$FEFF   USE $FEXX & $FFXX FOR SYSTEM
01039
01040  FF5A 4F                  CLRA
01041  FF5B A7   10             STA    -16,X    PARTITION 0
01042  FF5D 10AF 0E             STY    $0E,X    XEXXX & XFXXX TO FEXXX & FFX
01043  FF60 10AF 4E             STY    $0E,U    IN MAP TABLE TOO
01044  FF63 CC   F00D           LDD    #$F00D   A=MT CODE, B=START
01045
01046  FF66 E7   85       MONIT2 STB    B,X      SET EXTENDED ADDRESS+4K PAGE
01047  FF68 A7   C5              STA    B,U      IN MAP TABLE TOC
01048  FF6A 5A                   DECB
01049  FF6B 2A   F9   FF66       BPL    MONIT2
01050
01051  FF6D 10CE F3B2           LDS    #STACK
01052  FF71 10FF F3E0           STS    SP       RESET SYSTEM STACK POINTER
01053  FF75 108E EF00           LDY    #$EF00   START TESTING MEMORY HERE
01054  FF79 5F                  CLRB
01055
01056  FF7A 31   A9 F000  MONIT5 LEAY   -$1000,Y MOVE DOWN TO NEXT 4K BLOCK
01057  FF7E 108C FF00           CMPY   #$FF00   ALL DONE ?
01058  FF82 27   1A   FF9E      BEQ    MONIT6
01059  FF84 EE   A4             LDU    0,Y      SAVE THE CONTENTS OF MEMORY
01060  FF86 8E   A758           LDX    #$A758
01061  FF89 AF   A4             STX    0,Y      STORE THE TEST WORD HERE
01062  FF8B AC   A4             CMPX   0,Y      IS IT STILL THERE ?
01063  FF8D 26   EB   FF7A      BNE    MONIT5   IF NOT
01064  FF8F EF   A4             STU    0,Y      RESTORE THE CONTENTS
01065  FF91 34   20             PSHS   Y
01066  FF93 A6   E1             LDA    ,S++     GET RAM ADDRESS
01067  FF95 44                  LSRA            SHIFT TO GET MS-NIBBLE
01068  FF96 44                  LSRA
01069  FF97 44                  LSRA
01070  FF98 44                  LSRA
01071  FF99 34   02             PSHS   A        SAVE RAM ADDRESS
01072  FF9B 5C                  INCB            COUNT BLOCKS
01073  FF9C 20   DC   FF7A      BRA    MONIT5   GO FOR MORE RAM
```

```
01075  FF9E 8E   F3D0      MONIT6 LDX    #MAPTBL
01076  FFA1 5D                  TSTB                ANY MEMORY?
01077  FFA2 27   17   FFBB       BEQ    MONIT0      NO - WRITE BLANK TABLE
01078
01079  FFA4 C1   03             CMPB   #3
01080  FFA6 23   04   FFAC       BLS    MONIT8      LESS THAN OR EQUAL DOS
01081
01082  FFA8 C1   03        MONIT7 CMPB   #3
01083  FFAA 26   08   FFB4       BNE    MONIT9      NOT TO DOS YET
01084
01085  FFAC 8E   F3DE      MONIT8 LDX    #MAPTBL+$E -1=$D; -2=$C; -3=$B
01086  FFAF 1F   98             TFR    B,A
01087  FFB1 40                  NEGA
01088  FFB2 30   86             LEAX   A,X
01089
01090  FFB4 35   02        MONIT9 PULS   A
01091  FFB6 A7   80             STA    ,X+
01092  FFB8 5A                  DECB
01093  FFB9 26   ED   FFA8       BNE    MONIT7
01094
01095  FFBB 8D   06   FFC3 MONIT0 BSR    MOVTB2      MOVE MAP TABLE TO MAP RAM
01096  FFBD 7E   FBAF           JMP    MON0        INIT THE REST OF THE SYSTEM
01097
01098  FFC0 B7   FFE0      MOVTBL STA    LATCH       SET PARTITION #
01099  FFC3 8E   F3D0      MOVTB2 LDX    #MAPTBL      .
01100  FFC6 108E FFF0           LDY    #MAPRAM
01101  FFCA C6   10             LDB    #16
01102
01103  FFCC A6   80        MOVTB1 LDA    0,X+        GET RAM ADDRESS
01104  FFCE A7   A0             STA    0,Y+        STORE ADDRESS IN MAPPER
01105  FFD0 5A                  DECB
01106  FFD1 26   F9   FFCC       BNE    MOVTB1
01107  FFD3 39                  RTS
01108
```

```
01111                                   NAM     BFD49
01112
01113
01114                           *** 2-9-82 ADDED TRACK READ W/ SYNCH
01115                           *** ??-??-?? ADDED X SAVE IN STCHK
01116                           * THIS SECTION IS THE FIRMWARE TO DRIVE THE SSB
01117                           *  DISK CONTROLLER BOARD IN A 6809 SYSTEM
01118                           *
01119
01120                           * DISK SYSTEM EQUATES
01121
01122           D000    DSKRAM  EQU     $D000   SYSTEM AT C,D
01123
01124           F77C    PIA     EQU     $F77C   USING HIGH HARDWARE
01125
01126           F77C    DRA     EQU     PIA
01127           F77D    CRA     EQU     PIA+1
01128           F77E    DRB     EQU     PIA+2
01129           F77F    CRB     EQU     PIA+3
01130
01131           0003    DRSTEP  EQU     3       FOR 20/40 MS STEP TO START
01132
01133
01134                           * RAM TEMPORARIES FOR THE DISK FIRMWARE
01135
01136  F3E6                     ORG     RAME
01137
01138
01139  F3E6     0001    QMEM    RMB     1       LAST DRIVE CODE
01140  F3E7     0001    SIZEF   RMB     1       HARDWARE TYPE CONSTANT
01141  F3E8     0001    DRIVE   RMB     1       THIS DRIVE CODE
01142  F3E9     0001    TRACK   RMB     1
01143  F3EA     0001    SECTOR  RMB     1
01144  F3EB     0002    XR      RMB     2       SECTOR BUFFER ADDR
01145  F3ED     0001    DSK     RMB     1       SPARE
```

```
01148                          *
01149                          * DISK DRIVER JUMP TABLE
01150
01151
01152  F81E                            ORG     JUMPE
01153
01154  F81E    FE1C                    FDB     HSTART    BOOT ENTRY
01155  F820    FE1C                    FDB     HSTART    WARM BOOT ENTRY?
01156  F822    FE65                    FDB     INZPIA    INITIALIZE PIA
01157  F824    FD72                    FDB     RSCTOR    READ A SECTOR
01158  F826    FDE7                    FDB     WSCTOR    WRITE A SECTOR
01159  F828    FD6A                    FDB     RDTRK     READ A TRACK W/O SYNCH
01160  F82A    FDE3                    FDB     WRTRK     WRITE A TRACK
01161  F82C    FDCE                    FDB     TRSEEK    SEEK A TRACK
01162  F82E    FE88                    FDB     RESTR     RESTORE TO TRK Ø
01163  F830    FEA9                    FDB     WCREG     WRITE FDC CTRL REG
01164  F832    FEDC                    FDB     STCHK     WAIT FOR DONE & RETURN STATU
01165  F834    FEE4                    FDB     RDSTR     READ STATUS REGISTER
01166  F836    FE97                    FDB     WRSR      WRITE SECTOR REG
01167  F838    FEA0                    FDB     WRTR      WRITE TRACK REG
01168  F83A    FE8E                    FDB     WRDRT     WRITE (TRACK) TO DATA REG
01169  F83C    FEBA                    FDB     RDRT      READ TRACK REG
01170  F83E    FE80                    FDB     STPIN     STEP IN
01171
01172
01173
01174  FD62                            ORG     MONE
```

```
01177                                * READ TRACK POINTED TO BY (TRACK) WITH SYNCH
01178  FD62 8D   6A   FDCE RDTRK1 BSR    TRSEEK
01179  FD64 26   77   FDDD        BNE    RETN2
01180  FD66 C6   E4               LDB    #$E4
01181  FD68 20   11   FD7B        BRA    READ
01182
01183                                * READ TRACK POINTED TO BY (TRACK) W/O SYNCH
01184  FD6A 8D   62   FDCE RDTRK  BSR    TRSEEK
01185  FD6C 26   6F   FDDD        BNE    RETN2
01186  FD6E C6   E5               LDB    #$E5
01187  FD70 20   09   FD7B        BRA    READ
01188
01189                                * READ SECTOR DEFINED BY (SECTOR) & (TRACK)
01190  FD72 8D   5A   FDCE RSCTOR BSR    TRSEEK   SEEK TO TRACK
01191  FD74 26   67   FDDD        BNE    RETN2
01192  FD76 BD   FE97             JSR    WRSR     WRITE (SECTOR) TO FDC SECT R
01193  FD79 C6   8C               LDB    #$8C     FDC READ COMMAND
01194
01195  FD7B 34   04       READ    PSHS   B        SAVE FDC CMD
01196  FD7D 86   83               LDA    #$83     ACCESS FDC CTRL REG
01197  FD7F BD   FE46             JSR    FOURDR   BLEND DRIVE # & FDC CTRL BIT
01198  FD82 B7   F77C             STA    DRA
01199  FD85 86   17               LDA    #$FF-$E8
01200  FD87 B7   F77F             STA    CRB      ACCESS DDRB
01201  FD8A 7F   F77E             CLR    DRB      SET TO DRB TO OUTPUT
01202  FD8D C6   13               LDB    #$FF-$EC
01203  FD8F F7   F77F             STB    CRB      BACK TO DRB
01204  FD92 35   04               PULS   B        RECOVER FDC CMD
01205  FD94 F7   F77E             STB    DRB      SEND TO FDC
01206  FD97 C6   FF               LDB    #$FF     NOT 0 (INPUTS)
01207  FD99 B7   F77F             STA    CRB      ACCESS DDRB
01208  FD9C F7   F77E             STB    DRB      SET DDRB TO INPUTS
01209  FD9F 86   13               LDA    #$FF-$EC
01210  FDA1 B7   F77F             STA    CRB      BACK TO DRB
01211  FDA4 4F                    CLRA            FDC REG 3 & READ FF TO READ
01212  FDA5 BD   FE46             JSR    FOURDR   BLEND DRIVE # & FDC CTRL BIT
01213  FDA8 AE   04               LDX    4,X      GET (XR)
01214  FDAA B7   F77C             STA    DRA      SEND CONTROL BITS
01215  FDAD 34   08               PSHS   DP
01216  FDAF 86   F7               LDA    #PIA/$100 SET BASE PAGE TO PIA
01217  FDB1 1F   8B               TFR    A,DP
01218  FDB3 96   7D       READ1   LDA    CRA*$100/$100
01219  FDB5 2A   0B   FDC2        BPL    READ2    IF FDC INTRQ
01220  FDB7 48                    ASLA
01221  FDB8 2B   F9   FDB3        BMI    READ1    IF NO DRQ
01222  FDBA D6   7E               LDB    DRB*$100/$100 DRQ - GET THE BYTE
01223  FDBC E7   80               STB    ,X+      PUT IN MEMORY
01224  FDBE 96   7C               LDA    DRA*$100/$100 CLEAR THE FLAG
01225  FDC0 20   F1   FDB3        BRA    READ1    LOOP
01226
01227  FDC2 48            READ2   ASLA
01228  FDC3 2B   04   FDC9        BMI    READ3    IF NO DRQ
01229  FDC5 D6   7E               LDB    DRB*$100/$100 GET THE LAST BYTE
01230  FDC7 E7   84               STB    ,X
01231  FDC9 35   08       READ3   PULS   DP       RESTORE BASE PAGE
01232  FDCB 7E   FEE4             JMP    RDSTR    EXIT VIA STATUS REG READ
01233
```

```
01235                                * SEEK TO THE TRACK SPECIFIED BY (TRACK)
01236  FDCE BD   FE8E       TRSEEK  JSR    WRDRT     WRITE (TRACK) TO FDC DAT REG
01237  FDD1 C6   18                 LDB    #$18      FDC SEEK CODE
01238  FDD3 BD   FEA9               JSR    WCREG     WRITE FDC CTRL REG
01239  FDD6 BD   FEDC               JSR    STCHK     WAIT 4 DONE & GET STATUS
01240  FDD9 CA   01                 ORB    #$01      SET BUSY BIT?
01241  FDDB C5   98                 BITB   #$98      NOT RDY,NOT FND,CRC?
01242  FDDD 39              RETN2   RTS
01243
01244
01245                                * RESTORE BASE PAGE & GO TO STATUS CHECK
01246  FDDE 35   08         STCK    PULS   DP
01247  FDE0 7E   FEDC               JMP    STCHK     WAIT 4 DONE & GET STATUS
01248
01249
01250                                * WRITE A TRACK OF DATA
01251  FDE3 C6   F4         WRTRK   LDB    #$F4      WRITE TRACK CMD
01252  FDE5 20   09   FDF0          BRA    WRITE
01253
01254                                * WRITE A SECTOR OF DATA
01255  FDE7 8D   E5   FDCE  WSCTOR  BSR    TRSEEK    FIND THE TRACK
01256  FDE9 26   F2   FDDD          BNE    RETN2
01257  FDEB BD   FE97               JSR    WRSR      WRITE (SECTOR) TO FDC SECT R
01258  FDEE C6   AC                 LDB    #$AC      WRITE SECTOR CMD
01259  FDF0 86   83         WRITE   LDA    #$83      ACCESS BITS FOR FDC CRTL REG
01260  FDF2 8D   52   FE46          BSR    FOURDR    BLEND DRIVE # & CRTL BITS
01261  FDF4 B7   F77C               STA    DRA
01262  FDF7 4F                      CLRA
01263  FDF8 BD   FECA               JSR    INOUT     MAKE DRB OUTPUTS
01264  FDFB 86   80                 LDA    #$80      FDC REG 3
01265  FDFD 8D   47   FE46          BSR    FOURDR    BLEND DRIVE # & CTRL BITS
01266  FDFF AE   04                 LDX    4,X       MEMORY POINTER
01267  FE01 F7   F77E               STB    DRB       SEND WRITE CODE
01268  FE04 B7   F77C               STA    DRA       NOW TALK TO REG 3
01269  FE07 34   08                 PSHS   DP
01270  FE09 86   F7                 LDA    #PIA/$100 SET BASE PAGE TO PIA
01271  FE0B 1F   8B                 TFR    A,DP
01272  FE0D 96   7D         WRITE1  LDA    CRA*$100/$100 CHECK INTRQ
01273  FE0F 2A   CD   FDDE          BPL    STCK      IS ACTIVE
01274  FE11 48                      ASLA             CHECK DRQ
01275  FE12 2B   F9   FE0D          BMI    WRITE1    NOT ACTIVE
01276  FE14 E6   80                 LDB    ,X+       GET DATA
01277  FE16 D7   7E                 STB    DRB*$100/$100 SEND TO FDC
01278  FE18 96   7C                 LDA    DRA*$100/$100 CLEAR FLAG
01279  FE1A 20   F1   FE0D          BRA    WRITE1    LOOP
```

```
01281                              * BOOT LOADER
01282  FE1C 10CE F3B2       HSTART LDS    #STACK
01283  FE20 8E   F3E6              LDX    #QMEM
01284  FE23 6F   02                CLR    2,X       DRIVE 0 FOR BOOT
01285  FE25 6F   84                CLR    ,X        LAST DRIVE
01286  FE27 63   84                COM    ,X        FORCE HEAD LOAD DELAY
01287  FE29 C6   03                LDB    #$03      SLOW STEP
01288  FE2B E7   01                STB    1,X
01289  FE2D 1A   50                ORCC   #$50      SET INTERRUPT MASK
01290  FE2F 8D   34    FE65 STRT2  BSR    INZPIA    SET UP PIA
01291  FE31 4F                     CLRA
01292  FE32 5F                     CLRB
01293  FE33 FD   F3E9              STD    TRACK     TRACK 0, SECTOR 0
01294  FE36 8E   D000              LDX    #DSKRAM   LOAD ADDR
01295  FE39 BF   F3EB              STX    XR
01296  FE3C 8D   4A    FE88        BSR    RESTR     RESTORE THE DRIVE
01297  FE3E BD   FD72              JSR    RSCTOR    READ THE SECTOR
01298  FE41 26   EC    FE2F        BNE    STRT2     ERROR
01299  FE43 7E   D000              JMP    DSKRAM    EXECUTE CODE FROM THAT SECTO
01300
01301
01302                              * BLEND DRIVE # AND FDC CTRL BITS
01303                              * A=REG SELECT; B=FDC CTRL BITS
01304  FE46 34   02         FOURDR PSHS   A
01305  FE48 8E   F3E7              LDX    #SIZEF
01306  FE4B A6   01                LDA    1,X       GET DRIVE #
01307  FE4D A1   1F                CMPA   -1,X      SAME AS LAST?
01308  FE4F 27   07    FE58        BEQ    FOURD3
01309  FE51 A7   1F                STA    -1,X      NO - UPDATE QMEM AND ..
01310  FE53 86   FF                LDA    #$FF-$00  FORCE HEAD LOAD DELAY
01311  FE55 B7   F77C              STA    DRA
01312  FE58 A6   03         FOURD3 LDA    3,X       (SECTOR)
01313  FE5A 84   20                ANDA   #$20      GET SIDE BIT
01314  FE5C 44                     LSRA             POSITION CORRECTLY
01315  FE5D 44                     LSRA
01316  FE5E AA   1F                ORA    -1,X      ADD DRIVE BITS
01317  FE60 AA   E0                ORA    ,S+       ADD CONTROL BITS
01318  FE62 88   38                EORA   #$38      FLIP 3 BITS
01319  FE64 39                     RTS              RETURN WORD IN A
01320
```

```
01322                                * INITIALIZE PIA
01323  FE65 8E   F77C       INZPIA LDX   #PIA
01324  FE68 86   FF                 LDA   #$FF-$00
01325  FE6A A7   01                 STA   1,X       CRA
01326  FE6C A7   03                 STA   3,X       CRB
01327  FE6E A7   02                 STA   2,X       DDRB
01328  FE70 6F   84                 CLR   0,X       DDRA
01329  FE72 86   E1                 LDA   #$FF-$1E
01330  FE74 A7   01                 STA   1,X       CRA
01331  FE76 86   D3                 LDA   #$FF-$2C
01332  FE78 A7   03                 STA   3,X       CRB
01333  FE7A 20   68   FEE4         BRA   RDSTR     SET UP SEVERAL REGS
01334
01335  FE7C C6   38         STEP    LDB   #$38      STEP COMMAND
01336  FE7E 20   0A   FE8A         BRA   WCR1
01337
01338  FE80 C6   58         STPIN   LDB   #$58      STEP IN COMMAND
01339  FE82 20   06   FE8A         BRA   WCR1
01340
01341  FE84 C6   78         STPOUT  LDB   #$78      STEP OUT COMMAND
01342  FE86 20   02   FE8A         BRA   WCR1
01343
01344  FE88 C6   08         RESTR   LDB   #$08      RESOTRE TO 00 COMMAND
01345
01346  FE8A 8D   1D   FEA9 WCR1    BSR   WCREG     SEND CMD TO FDC
01347  FE8C 20   4E   FEDC         BRA   STCHK     GO WAIT FOR READY
01348
01349                                * WRITE (TRACK) TO FDC DATA REG
01350  FE8E 86   80         WRDRT   LDA   #$80      FDC CONTROL BITS
01351  FE90 F6   F3E9               LDB   TRACK
01352  FE93 C4   7F                 ANDB  #$7F      MASK TO SIZE
01353  FE95 20   17   FEAE         BRA   WTFDC
01354
01355                                * WRITE (SECTOR) TO FDC SECTOR REG
01356  FE97 86   81         WRSR    LDA   #$81      FDC CTRL BITS
01357  FE99 F6   F3EA               LDB   SECTOR
01358  FE9C C4   1F                 ANDB  #$1F      MASK TO SIZE
01359  FE9E 20   0E   FEAE         BRA   WTFDC
01360
01361                                * WRITE (TRACK) TO FDC TRACK REG
01362  FEA0 86   82         WRTR    LDA   #$82      FDC CTRL BITS
01363  FEA2 F6   F3E9               LDB   TRACK
01364  FEA5 C4   7F                 ANDB  #$7F      MASK TO SIZE
01365  FEA7 20   05   FEAE         BRA   WTFDC
01366
01367                                * WRITE (B) TO FDC CONTROL REG
01368  FEA9 86   83         WCREG   LDA   #$83      FDC CTRL BITS
01369  FEAB FA   F3E7               ORB   SIZEF     GET STEP RATE BITS
01370
01371  FEAE 8D   96   FE46 WTFDC   BSR   FOURDR    BLEND DRIVE # & FDC CTRL BIT
01372  FEB0 B7   F77C               STA   DRA
01373  FEB3 4F                      CLRA
01374  FEB4 8D   14   FECA         BSR   INOUT     SET FOR OUTPUTS
01375  FEB6 F7   F77E               STB   DRB       WRITE DATA/CMD
01376  FEB9 39                      RTS
01377
```

```
01379                              * READ FDC TRACK REG
01380  FEBA 86   FF          RDRT   LDA    #$FF
01381  FEBC 8D   0C    FECA          BSR    INOUT     MAKE INPUTS
01382  FEBE 86   C2                 LDA    #$C2      READ TRACK CMD
01383  FEC0 BD   FE46               JSR    FOURDR    BLEND DRIVE # & FDC CTRL BIT
01384  FEC3 B7   F77C               STA    DRA
01385  FEC6 12                      NOP               KILL TIME
01386  FEC7 7E   FEFC               JMP    LDDRB     USE JMP, NOT BRA
01387
01388
01389                              * SET PIA TO IN OR OUT BASED ON (A)
01390  FECA 34   02          INOUT  PSHS   A
01391  FECC 86   D7                 LDA    #$FF-$28
01392  FECE B7   F77F               STA    CRB
01393  FED1 35   02                 PULS   A         FF=INPUT; 00=OUTPUT
01394  FED3 B7   F77E               STA    DRB
01395  FED6 86   D3                 LDA    #$FF-$2C
01396  FED8 B7   F77F               STA    CRB
01397  FEDB 39                      RTS
01398
01399
01400                              * WAIT FOR READY AND READ STATUS
01401  FEDC B6   F77D        STCHK  LDA    CRA
01402  FEDF 2B   FB    FEDC          BMI    STCHK     WAIT FOR READY
01403  FEE1 B6   F77C               LDA    DRA       CLEAR FLAG
01404
01405
01406                              * READ FDC STATUS REG
01407  FEE4 86   FF          RDSTR  LDA    #$FF
01408  FEE6 8D   E2    FECA          BSR    INOUT     SET FOR INPUTS
01409  FEE8 86   C3                 LDA    #$C3      READ STATUS CODE
01410  FEEA 34   10                 PSHS   X
01411  FEEC BD   FE46               JSR    FOURDR    BLEND DRIVE # & FDC CTRL BIT
01412  FEEF 35   10                 PULS   X
01413  FEF1 B7   F77C               STA    DRA
01414  FEF4 12                      NOP               KILL TIME
01415  FEF5 8D   05    FEFC          BSR    LDDRB     READ THE REG
01416  FEF7 B6   F77C               LDA    DRA       CLEAR THE FLAG
01417  FEFA 5D                      TSTB             TEST STATUS
01418  FEFB 39                      RTS
01419
01420
01421  FEFC F6   F77E        LDDRB  LDB    DRB
01422  FEFF 39                      RTS
01423
01424                               END


TOTAL ERRORS 00000--00000
TOTAL WARNINGS 00000--00000
```

```
E004 ACIAS  00109*00740
0015 ACINZ  00110*00764 00797
F886 ACRLF  00234*00542 00595
F991 AREG   00415*00894
F8A7 BADDR  00257*00266 00269 00331 00336 00474 00480 00633 00721
FD59 BANNER 00766 00949*
FB67 BOOTD  00685*00928
FB6B BOOTE  00678 00682 00686*
FB5B BOOTF  00677*00932
FB61 BOOTH  00681*00934
F3EE BPTBL  00146*00392 00624 00643 00756
F400 BPTBLE 00147*00396 00629 00653 00759
FAF8 BREAK  00624*00892
F998 BREG   00421*00896
FAFC BRK1   00625*00630
FB11 BRK4   00627 00633*
FD4C CCLTRS 00367 00946*
F700 CLOCK  00116*01033 01034
FC2F CMD    00155 00399 00797*00805 00810
FC41 CMD0   00803*00807
FC3E CMD01  00802*00814
F3E2 CPORT  00135*00180 00189 00211 00741 00765 00798
F77D CRA    01127*01218 01272 01401
F77F CRB    01129*01200 01203 01207 01210 01392 01396
F9C5 CREG   00457*00898
FD46 CRLFS  00245 00943*
FCB3 CTABL  00801 00873*
0022 DAY    00098*00965
F816 DBOOT  00165*00685
F99F DPREG  00427*00900
F77C DRA    01126*01198 01214 01224 01261 01268 01278 01311 01372 01384
            01403 01413 01416
F77E DRB    01128*01201 01205 01208 01222 01229 01267 01277 01375 01394
            01421
F3E8 DRIVE  01141*
0003 DRSTEP 01131*
F3ED DSK    01145*
D000 DSKRAM 01122*01294 01299
FAAF DUMPM  00586*00912
FAC2 DUMPM1 00595*00618
FAD0 DUMPM2 00601*00603
FADD DUMPM3 00607*00616
FAE9 DUMPM4 00610 00613*
FAEB DUMPM5 00612 00614*
FA83 DUMPMT 00557*00924
FA87 DUMPT1 00559*00564
FA9A DUMPT2 00569*00575
FF05 DUMRTI 00744 00968*
0048 DVRVER 00102*00950
0005 EDIT   00095*00966
FD49 ERRMSG 00687 00944*
F9E9 EXAM   00480*00880
F9EC EXAM1  00481*00493 00497 00501 00503 00507 00511
FA0D EXAM4  00491 00494*
FA15 EXAM5  00495 00498*
FA1D EXAM6  00499 00502*
FC69 EXBLK  00828*00910
```

```
FC99 EXBLK2 00847 00851*
FCA9 EXBLK3 00860*00863
FCB0 EXBLK4 00850 00865*
F818 FBOOT  00166*00677
FB43 FBYTA  00502 00661*
FB40 FBYTE  00251 00522 00660*
FB59 FBYTR  00670 00674*
FB58 FBYTS  00662 00672*
FA47 FINDS1 00524*00530
FA58 FINDS2 00526 00532*00550
FA62 FINDS3 00537*00541
FA78 FINDS6 00539 00547*
FA80 FINDS9 00523 00548 00552*
FA82 FINDSR 00553*
FA35 FINDST 00516*00878
00F0 FLIPB2 00115*00774
0000 FLIPBT 00113*00572
FE58 FOURD3 01308 01312*
FE46 FOURDR 01197 01212 01260 01265 01304*01371 01383 01411
FD54 FULL   00631 00947*
F89F GBYTE  00251*00257 00260 00322 00460
F90B GO     00341*00882
F81A HBOOT  00167*00681
F8CF HEXD1  00280 00283*
F8BF HEXDIG 00275*00661 00669 00819
F8D3 HEXR   00252 00276 00278 00282 00285*
FE1C HSTART 00888 01154 01155 01282*
FF3E IFIRQ  00998*01013
FF42 IIRQ   01000*01014
F84A INCH   00156 00186*00187 00205 00240
F84E INCH1  00188*00236
F840 INCHK  00158 00179*00186 00234
F857 INEE   00195*00487 00660 00668 00802 00818
F864 INEEE  00157 00195 00205*00723
F863 INEER  00197 00199 00201*
F8F2 INITM  00319*00886
F8FA INITM1 00323*00325
FF3A INMI   00996*01016
FECA INOUT  01263 01374 01381 01390*01408
FF4A INTEND 01003*01021
FE65 INZPIA 01156 01290 01323*
E000 IO     00106*00109
FF46 IRESVD 01002*01010
FF06 ISWI   00970*01015
FF0A ISWI2  00972*01012
FF0E ISWI3  00974*01011
FF1E ISWI30 00981*00989
FF26 ISWI31 00980 00984*
FAAC JOUTS  00559 00580*
F907 JUMP   00336*00884
F81E JUMPE  00170*01152
F903 JUMPI  00331*00926
FFE0 LATCH  00108*01098
FEFC LDDRB  01386 01415 01421*
FC68 LIMCHX 00822 00824*
F8B3 LIMITS 00266*00319 00516 00586
FB76 LRA    00163 00694*
```

```
0012 LRAFIX 00118*00708
FFF0 MAPRAM 00107*01036 01100
F3D0 MAPTBL 00133*00568 00700 00771 00778 00837 01037 01075 01085 01099
F3E4 MEM    00136*00481 00483 00485 00504
FBAF MON0   00733*01096
FBB3 MON20  00735*00737
FBEF MON25  00758*00760
FD62 MONE   00952*00955 01174
FF4A MONIT  00154 01017 01033*
FFBB MONIT0 01077 01095*
FF66 MONIT2 01046*01049
FF7A MONIT5 01056*01063 01073
FF9E MONIT6 01058 01075*
FFA8 MONIT7 01082*01093
FFAC MONIT8 01080 01085*
FFB4 MONIT9 01083 01090*
0009 MONTH  00097*00965
FFCC MOVTB1 01103*01106
FFC3 MOVTB2 01095 01099*
FFC0 MOVTBL 00168 00865 01098*
FB3D NBYTE  00525 00659*
FC5A ONEDIG 00818*00828 00832
FAA9 OUT1HS 00561 00573 00579*
F8E2 OUT2H  00305*00312 00313 00782
F8EC OUT2HS 00313*00349 00357 00358 00359 00459 00486 00601
F8EA OUT4HS 00312*00360 00361 00362 00363 00365 00472 00484 00545 00598
F868 OUTCH  00159 00210*00226 00301 00375 00408 00510 00614 00784
F86D OUTCH1 00212*00214
F8E0 OUTH1  00301*00315
F8D4 OUTHL  00289*00306 00570
F8D8 OUTHR  00296*00308 00579
F8EE OUTS   00268 00271 00314*00353 00366 00376 00508 00580 00605 00659
            00808 00831
F9D1 PCREG  00466*00908
F896 PCRLF  00161 00221 00235 00244*00482 00557 00566 00722
F87F PDATA  00160 00222 00227*00246 00632
F87D PDATA0 00226*00229
F77C PIA    01124*01126 01127 01128 01129 01216 01270 01323
FD0F PROMPT 00799 00940*
F879 PSTR   00162 00221*00347 00688 00767 00800
F3E6 QMEM   01139*01283
F3E6 RAME   00138*01136
FEBA RDRT   01169 01380*
FEE4 RDSTR  01165 01232 01333 01407*
FD6A RDTRK  01159 01184*
FD62 RDTRK1 01178*
FD7B READ   01181 01187 01195*
FDB3 READ1  01218*01221 01225
FDC2 READ2  01219 01227*
FDC9 READ3  01228 01231*
F910 REGS   00346*00398 00876
F928 REGS0  00351 00357*
F948 REGS2  00371*00378
F950 REGS3  00373 00375*
F916 REGSQ  00348*
F921 REGSS  00353*00355
F934 REGST  00356 00363*
```

```
FD11 REGTTL 00346 00941*
0001 RELEAS 00094*00964
FE88 RESTR  01162 01296 01344*
FC2B RESTR1 00239 00253 00796*
FC20 RESTRT 00164 00789*00820 00829 00833 00836 00930
FDDD RETN2  01179 01185 01191 01242*01256
F9C8 RMOD   00431 00459*
F9DE RMOD2  00452 00469 00472*
F9BE RMOD2V 00438 00444 00451*
F9A4 RMODV  00417 00423 00430*
F9E8 RMODX  00476*00489
FD72 RSCTOR 01157 01190*01297
F9EF RXAM   00482*00916
F3EA SECTOR 01143*01357
F3E7 SIZEF  01140*01305 01369
FC53 SLOPT  00812*00874
FCFC SLTBL  00813 00923*
F95A SNMI   00383*00749
F3E0 SP     00134*00341 00348 00364 00369 00383 00388 00415 00421 00427
            00430 00436 00442 00448 00451 00457 00466 00468 00790 00796
            01052
0000 SPARE  00955*
F960 SSWI   00388*00751
F96F SSWI1  00393*00397
F97B SSWI2  00384 00398*00409
F980 SSWI3  00394 00402*
F3B2 STACK  00123*00789 01051 01282
FEDC STCHK  01164 01239 01247 01347 01401*01402
FDDE STCK   01246*01273
FE7C STEP   01335*
FE80 STPIN  01170 01338*
FE84 STPOUT 01341*
FE2F STRT2  01290*01298
FC06 SUM    00773*00779
FC0F SUM1   00775 00778*
F3CE SVCL   00132*00755 00988
F3CC SVCO   00131*00754 00978
FB9D TEXT   00721*00914
FBA3 TEXT1  00723*00727
FBAE TEXTR  00725 00729*
F3E9 TRACK  01142*01293 01351 01363
FDCE TRSEEK 01161 01178 01184 01190 01236*01255
FB1F UNBRK  00643*00792 00918
FB24 UNBRK1 00645*00654
FB31 UNBRK2 00647 00650*
F9P9 UREG   00448*00906
0035 VERSON 00101*00950
F3C6 VFIRQ  00128*00747 00998
F3C8 VIRQ   00129*00748 01000
F3BE VNMI   00124*00750 00996
F3C0 VRESVD 00125*01002
F3CA VSWI   00130*00752 00970
F3C4 VSWI2  00127*00746 00972
F3C2 VSWI3  00126*00745 00983
FE8A WCR1   01336 01339 01342 01346*
FEA9 WCREG  01163 01238 01346 01368*
FE8E WRDRT  01168 01236 01350*
```

```
FDF0 WRITE  01252 01259*
FE0D WRITE1 01272*01275 01279
FE97 WRSR   01166 01192 01257 01356*
FEA0 WRTR   01167 01362*
FDE3 WRTRK  01160 01251*
FDE7 WSCTOR 01158 01255*
FEAE WTFDC  01353 01359 01365 01371*
F3EB XR     01144*01295
F9AB XREG   00436*00902
0083 YEAR   00099*00965
F9B2 YREG   00442*00904
D283 ZWARMS 00119*00890
```