EPROM2 5/21/82

```
0001                    IDT    'MICROJW2'
0002                    OPTION XREF
0003                *   Jef Winsor              invert control lines
0004                *                           standrad microcode
0005                * MICROTAPE PERIPHERAL CONTROL SOFTWARE  2.5MHz
0006                *-------------------------------------------------------
0007                * SYSTEM HARDWARE IMPLEMENTATION  ,----,
0008                *                          ,------|SW  | ADDRESS  read 0 - 7
0009                *,--------,               | ,----|SW  | SELECT   add 1 for
0010                *|   70C20|               | | ,--|JMP |          device #
0011                *|        |               | | |  |____|      ,-------,
0012                *|04  6 A0|<--ADDR 0------' | |  ,-----------|VOLTAGE|
0013                *|    7 A1|<--ADDR 1--------' |  | ,---------|CONTROL|
0014                *|    8 A2|<--ADDR 2----------'  | |         |_______|
0015                *|    9 A3|<--NC                | |                 ^
0016                *|   10 A4|<--NC                | |                 |
0017                *|   16 A5|<--LOW BATTERY-------' |     ,-------,   |
0018                *|   15 A6|<--MOTOR PROHIBITED----'     |WP AND |   |
0019                *|   11 A7|<--INPUT RAW DATA--------,   |EOT/BOT|   |
0020                *|        |                  ,------+-->|SENSORS|   |
0021                *|06  3 B0|-->LED            |      |   |_______|   |
0022                *|    4 B1|-->WRITE ENABLE---+------+--,    |   ^   |
0023                *|    5 B2|-->EOT/BOT SENSOR-'      |  |    |   |   |
0024                *|   37 B3|-->MOTOR-----------------+--+-*--+---+---'
0025                *|   38 B4|-->A-LATCH-----------,   |  |  |   |   |
0026                *|    1 B5|-->R/W---------------, |  \  |   |   |   |
0027                *|   39 B6|-->ENABLE-----------, | |   |   |   |   |
0028                *|    2 B7|-->NC                V V V   |   |   |   |
0029                *|        |                ,-------,  |   |   |   |
0030                *|08 28 C0|<->DATA 0/RS---<->|       |  |   |   |   |
0031                *|   29 C1|<->DATA 1------<->|I/O BUS|  |   |   |   |
0032                *|   30 C2|<->DATA 2------<->|       |  |   |   |   |
0033                *|   31 C3|<->DATA 3------<->|SUPPORT|  |   |   |   |
0034                *|   32 C4|---NC         ,--->|_______|  |   |   |   |
0035                *|   33 C5|---NC         |      |       |   |   |   |
0036                *|   34 C6|---NC         |  ,---+-------+---+---'   |
0037                *|   35 C7|---CS---------'  |  ,-+-------+---+-------'
0038                *|        |                |  | |       |   |
0039                *|0A 27 D0|-->WRITE DATA---+--+-+---,   |   |
0040                *|   26 D1|<--EOT/BOT------'  | |   |   |   |
0041                *|   24 D2|<--WP--------------' |   V   | V V
0042                *|   23 D3|<--NC                | ,-----------,
0043                *|   22 D4|<--  --*-- +V ROM    | |TAPE DRIVE|
0044                *|   21 D5|<--  --*      OPTIONS| |__________|
0045                *|   20 D6|<--  --'             |       |
0046                *|   19 D7|-->WP SENSOR---------+-------+-------'
0047                *|        |                     |       |
0048                *| 13 INT1|<--IRQ---------------'       |
0049                *| 12 INT3|<--TAPE TRANSITION PULSE------'
0050                *|________|
```

```
0052                *-------------------------------------------------------
0053                * register file equates
0054    0000  AREG    EQU   R0                a register
0055    0001  BREG    EQU   R1                b register
0056    0002  BITCON  EQU   R2                nibble bit-counter
0057    0003  STACK   EQU   R3                soft stack location
0058    000E  STACKL  EQU   >0E               reserved stack length
0059    0011  REGS    EQU   STACK+STACKL      start of registers
0060    0012  NIBCON  EQU   REGS+1            DL nibble-counter
0061    0012  COUNT   EQU   NIBCON            16 bit counter
0062    0014  CHKSUM  EQU   NIBCON+2          checksum value (16 bits)
0063    0015  INT2    EQU   CHKSUM+1          INT2 entry point (m code
0064    0017  INT2V   EQU   CHKSUM+3          INT2 ram-vector
0065    0019  DATAP   EQU   INT2V+2           data pointer
0066    0019  DLPCNT  EQU   DATAP             DL pointer/counter
0067    001A  DATA    EQU   DATAP+1           nibble pack/unpack
0068    001B  EOTIMR  EQU   DATA+1            EOT duty cycle timer
0069    001C  FFLAG   EQU   EOTIMR+1          drive status flags
0070    001E  RECFIL  EQU   FFLAG+2           number of records
0071    0020  MAXLEN  EQU   RECFIL+2          maximum record length
0072    0022  NREC    EQU   MAXLEN+2          current record number
0073    0023  NFILE   EQU   NREC+1            current file number
0074    000F  NMLEN   EQU   >0F               file name length
0075    003E  SAB     EQU   NFILE+12+NMLEN    ms address byte of SAB
0076    0024  FNAMEL  EQU   SAB-11-NMLEN      last character of FILENA
0077    0032  FNAME1  EQU   SAB-12            first character of FILEN
0078    0033  ATTRIB  EQU   SAB-11            attributes byte
0079    0035  BLOPEN  EQU   SAB-9             BL data from OPEN
0080    0037  DLEN    EQU   SAB-7             data length
0081    0039  BLEN    EQU   SAB-5             buufer length
0082    003B  RNUM    EQU   SAB-3             record number
0083    003C  LUNO    EQU   SAB-2             logical unit number
0084    003D  CCODE   EQU   SAB-1             command code
0085    003E  DCODE   EQU   SAB               device code
0086    003E  STATUS  EQU   DCODE             status to be returned
0087    0036  TEMP8   EQU   DLEN-1            temporary use
0088    0037  TEMP7   EQU   DLEN              temporary use
0089    0038  TEMP6   EQU   BLEN-1            temporary use
0090    0039  TEMP5   EQU   BLEN              temporary use
0091    003A  TEMP4   EQU   RNUM-1            temporary use
0092    003B  TEMP3   EQU   RNUM              temporary use
0093    003C  TEMP2   EQU   LUNO              temporary use
0094    003D  TEMP1   EQU   CCODE             temporary use
0095    003E  TEMP0   EQU   DCODE             temporary use
0096    003F  DIRECT  EQU   R63               start of ram directory
0097    0040  FILE0   EQU   R64               start of FILE0 status
0098    0044  FILE1   EQU   R68               start of FILE1 status
0099    0048  FILE2   EQU   R72               start of FILE2 status
0100    004C  FILE3   EQU   R76               start of FILE3 status
0101    0050  FILE4   EQU   R80               start of FILE4 status
0102    0054  FILE5   EQU   R84               start of FILE5 status
0103    0058  FILE6   EQU   R88               start of FILE6 status
0104    005C  FILE7   EQU   R92               start of FILE7 status
0105    0060  FILE8   EQU   R96               start of FILE8 status
0106    0064  FILE9   EQU   R100              start of FILE9 status
0107    0068  FILEA   EQU   R104              start of FILEA status
0108    006C  FILEB   EQU   R108              start of FILEB status
```

```
0109    0070  FILEC  EQU   R112    start of FILEC status
0110    0074  FILED  EQU   R116    start of FILED status
0111    0078  FILEE  EQU   R120    start of FILEE status
0112    007C  FILEF  EQU   R124    start of FILEF status
0113    007F  ENDRCT EQU   R127    end of directory
```

```
0115                *-------------------------------------------------------------
0116                * peripheral file equates
0117      0000      IOCNTL EQU    P0                  i/o control
0118      0002      TIME   EQU    P2                  timer start value
0119      0003      CAPTUR EQU    P3                  timer value @ INT3
0120      0003      PSCALE EQU    P3                  prescale control
0121      0003      TIMER  EQU    P3                  timer control
0122      0004      TEST   EQU    P4                  data & system test-port
0123      0006      DRIVE  EQU    P6                  drive & system control
0124      000A      WAFER  EQU    P10                 wafer i/o
0125      000B      DDRD   EQU    P11                 ddr for PORT D
0126      0180      BDATA  EQU    >180                bus data
0127      0181      BCNTL  EQU    >181                bus control
0128      0181      BSTAT  EQU    >181                bus status
```

```
0130            *-------------------------------------------------------------
0131            * flag use descriptions
0132            *-----------------------------------
0133            * FFLAG       file management flags
0134            *-----------------------------------
0135            * Value!!   1   !   0
0136            *-----------------------
0137            * Bit 7!!dirctry!no dirc
0138            *-----------------------
0139            * Bit 6!!protect!no prot
0140            *-----------------------
0141            * Bit 5!!filname!no name
0142            *-----------------------
0143            * Bit 4!!open   !no open
0144            *-----------------------
0145            * Bit 3!!found  !not fnd
0146            *-----------------------
0147            * Bit 2!!restord!not rst
0148            *-----------------------
0149            * Bit 1!!at EOF !not EOF
0150            *-----------------------
0151            * Bit 0!!error  !no erro
0152            *-----------------------
0153            *
0154            *
0155            *
0156            *-----------------------------------
0157            * RECFIL-1    file parameter flags
0158            *-----------------------------------
0159            * Value!!   1   !   0
0160            *-----------------------
0161            * Bit 7!!active !inactiv
0162            *-----------------------
0163            * Bit 6!!last   !not lst
0164            *-----------------------
0165            * Bit 5!!at EOT !not EOT
0166            *-----------------------
0167            * Bit 4!!intrnal!display
0168            *-----------------------
0169            * Bit 3!!  ---  !  ---
0170            *-----------------------
0171            * Bit 2!!  ---  !  ---
0172            *-----------------------
0173            * Bit 1!!  ---  !  ---
0174            *-----------------------
0175            * Bit 0!!  ---  !  ---
0176            *-----------------------
```

```
0178            *-------------------------------------
0179            * ATTRIB      file attribute flags
0180            *-------------------------------------
0181            * Value!!   1   !   0
0182            *-----------------------
0183            * Bit 7!!out/upd!app/inp
0184            *-----------------------
0185            * Bit 6!!inp/upd!app/out
0186            *-----------------------
0187            * Bit 5!!relativ!sequent
0188            *-----------------------
0189            * Bit 4!!fixed  !variabl
0190            *-----------------------
0191            * Bit 3!!intrnal!display
0192            *-----------------------
0193            * Bit 2!!  res  !  res
0194            *-----------------------
0195            * Bit 1!!  res  !  res
0196            *-----------------------
0197            * Bit 0!!  res  !  res
0198            *-----------------------
0199            *
0200            *
0201            *
0202            *-------------------------------------
0203            * DIRECT      directory data
0204            *-------------------------------------
0205            * Value!!   1   !   0
0206            *-----------------------
0207            * Bit 7!!at EOT !not EOT
0208            *-----------------------
0209            * Bit 6!!  ---  !  ---
0210            *-----------------------
0211            * Bit 5!!  ---  !  ---
0212            *-----------------------
0213            * Bit 4!!  ---  !  ---
0214            *-----------------------
0215            * Bit 3!!revision # bit 3
0216            *-----------------------
0217            * Bit 2!!revision # bit 2
0218            *-----------------------
0219            * Bit 1!!revision # bit 1
0220            *-----------------------
0221            * Bit 0!!revision # bit 0
0222            *-----------------------
```

```
0224                *-------------------------------------------------------
0225                * wafer driw/opnrst/fclose/e control constant equates
0226      00FF      INITDR EQU    >FF                    initialize drive
0227      0001      INITWF EQU    >01                    initialize wafer ddr
0228      0001      INVBIT EQU    >01                    wafer-write invert
0229      00F7      MT     EQU    >F7                    turn on motor
0230      00F3      MTSN   EQU    >F3                    turn on motor & sensor
0231      00F1      MTSNWE EQU    >F1                    turn on motor, sensor &
0232      00F5      MTWE   EQU    >F5                    turn on motor & WE
0233      0004      NONSN  EQU    >04                    turn off EOT/BOT sensor
0234      00FB      SN     EQU    >FB                    turn on EOT/BOT sensor
0235      00FF      STOP   EQU    >FF                    turn off drive
0236      00FD      WE     EQU    >FD                    turn on WE
0237      0080      WPSENS EQU    >80                    turn on WP sensor
0238      007F      WPSN   EQU    >7F                    turn on WP sensor
0239                * wafer test contant equates
0240      0002      EOTTST EQU    >02                    bit-test for EOT
0241      0080      INPUT  EQU    >80                    wafer data-bit
0242      0004      SENSET EQU    >04                    test bit for sensor
0243      0004      WP     EQU    >04                    test for WP
0244                * wafer data consatnt equates
0245      0008      SETBIT EQU    >08                    store a "1" bit
0246                *-------------------------------------------------------
0247                * bus control constant equates
0248      000C      DISABL EQU    >0C                    disable IBC
0249      0001      DROP   EQU    >01                    drop HSK bit
0250      0000      HSKSET EQU    >00                    let HSK float
0251      0004      INHIB  EQU    >04                    inhibit IBC
0252      0001      RELEAS EQU    >01                    release-HSK bit
0253                * bus test constant equates
0254      0001      HSK    EQU    >01                    bus ready bit-test
0255      0008      IRQ    EQU    >08                    bus data ready bit-test
0256      0002      BAV    EQU    >02                    BAV active test
0257                *-------------------------------------------------------
0258                * timer constant equates
0259      0082      BEGINR EQU    >82                    start read timer
0260      0080      BEGINW EQU    >80                    start write timer
0261      000A      BITIME EQU    >0A   8KBaud @ 2.5MHz  data half-bit time
0262      0080      START  EQU    >80                    timer-start bit
0263                * software loop constants
0264      004B      BOSTIM EQU    >4B                    wait valid sync (150 ms)
0265      0096      BOTIME EQU    >96                    BOT past head (300 ms)
0266      004B      DWNTIM EQU    >4B                    motor downtime (150 ms)
0267      0046      EOTCNT EQU    >46                    EOT test duty cycle
0268      0023      EOTCN2 EQU    >23                    fast EOT duty cycle
0269      000A      WPDLAY EQU    BITIME                 WP sensor up-time
0270                *-------------------------------------------------------
0271                * interrupt select & clear constant equates
0272      0043      I1CS   EQU    >43                    clear and select INT1
0273      006A      I123C  EQU    >6A                    clear INT1,2&3 flags
0274      0048      I2C    EQU    >48                    clear INT2 flag
0275      004C      I2CS   EQU    >4C                    clear and select INT2
0276      0004      I2S    EQU    >04                    test INT2 select bit
0277      0068      I23C   EQU    >68                    clear INT2&3 flags
0278      007C      I23CS  EQU    >7C                    clear and select INT2&3
0279      0060      I3C    EQU    >60                    clear INT3 flag
0280      0070      I3CS   EQU    >70                    clear and select INT3
```

```
0281  0074  I3C23S EQU  >74                clear INT3 and select IN
0282  0020  I3FLAG EQU  >20                test INT3 flag-bit
0283  00A4  RDBIT1 EQU  >A4                read bit opcode
0284  0020  RDBIT2 EQU  >20                read bit parameter
0285  0000  RDBIT3 EQU  >00                read bit parameter
0286  00D6  WRBIT1 EQU  >D6                write bit opcode
0287  0002  WRBIT2 EQU  >02                write bit parameter
0288        *--------------------------------------------------------
0289        * flag set & test bit constant equates
0290  0080  ACTIVE EQU  >80                active file flag
0291  0040  DFORMA EQU  >40                initialize directory
0292  0010  DISPLY EQU  >10                display flag in director
0293  0002  EOFFLG EQU  >02                file is at EOF flag
0294  0080  EOTFLG EQU  >80                file is at EOT flag
0295  0080  FDIREC EQU  >80                directory present flag
0296  0090  FDROPN EQU  >90                directory & open active
0297  0001  FERROR EQU  >01                error flag
0298  0008  FFOUND EQU  >08                file/record found flag
0299  0020  FNAME  EQU  >20                file name/number flag
0300  0010  FOPEN  EQU  >10                file open flag
0301  0004  FRESTO EQU  >04                file restored flag
0302  0040  FWP    EQU  >40                write protect flag
0303  0060  INIFLG EQU  >60                initialize flags
0304  0008  INTDIS EQU  >08                internal/display in atrr
0305  0010  INTERN EQU  >10                internal flag in directo
0306  0040  LAST   EQU  >40                last file flag
0307  00C0  NEWFIL EQU  >C0                create new file
0308  0094  NUFLG1 EQU  >94                set up flag bits
0309  00FC  NUFLG2 EQU  >FC                set up flag bits
0310  0000  REVO   EQU  >00                format version 0
0311  0020  SEQUEN EQU  >20                sequential file
0312        * flag reset bit constant equates
0313  00F7  FMSSNG EQU  >F7                file/record not found
0314  00DF  FNUMBR EQU  >DF                file number
0315  007F  INACTV EQU  >7F                reset active flag
0316  00BF  LASTO  EQU  >BF                reset last flag
0317  00EF  OPNRST EQU  >EF                reset open flag
0318  00FD  RSTEOF EQU  >FD                reset EOF flag
0319  007F  RSTEOT EQU  >7F                reset EOT flag
0320  00FE  RSTERR EQU  >FE                reset error flag
0321  00FD  RSTWRT EQU  >FD                reset write flag
0322  00FB  USED   EQU  >FB                reset restore flag
0323  00BF  WPO    EQU  >BF                reset WP flag
0324        *--------------------------------------------------------
0325        * access mode test constant equates
0326  0080  TSTI   EQU  >80                test for input
0327  0040  TSTIU  EQU  >40                test for input or update
0328  00C0  TSTNA  EQU  >C0                test for not append
0329  00C0  TSTNI  EQU  >C0                test for not input
0330  0080  TSTO   EQU  >80                test for output
0331  0040  TSTOA  EQU  >40                test for output or appen
0332  0080  TSTU   EQU  >80                test for update
0333        *--------------------------------------------------------
0334        * miscellaneous constant equates
0335  0020  BLANK  EQU  >20                ASCII blank character
0336  008C  BROPC  EQU  >8C                branch opcode
0337  0000  FREE   EQU  >00                unused space in table
```

```
0338  000F  LSN     EQU  >0F    mask to clear msn
0339  00AA  SYNCDT  EQU  >AA    sync data
0340  0000  ZERO    EQU  >00    constant
```

```
0342                *-------------------------------------------------------
0343                * command code equates
0344      000E      CCATAL EQU    >0E                 catalog c code
0345      000D      CFORMA EQU    >0D                 format c code
0346      0001      CCLOSE EQU    >01                 close c code
0347      0002      CCLSDL EQU    >02                 close & delete c code
0348      0006      CDELET EQU    >06                 delete c code
0349      00FE      CNULL  EQU    >FE                 null c code
0350      0000      COPEN  EQU    >00                 open c code
0351      0003      CREAD  EQU    >03                 read data c code
0352      00FF      CRESET EQU    >FF                 reset c code
0353      0005      CPOSIT EQU    >05                 position record c code
0354      0007      CRSTAT EQU    >07                 return status c code
0355      000C      CVERIF EQU    >0C                 verify c code
0356      0004      CWRITE EQU    >04                 write data c code
0357                *-------------------------------------------------------
0358                * status code equates
0359      0002      RSATTR EQU    >02                 attribute error code
0360      000C      RSBLEN EQU    >0C                 buffer length error code
0361      0001      RSCHAR EQU    >01                 device/file characterist
0362      0004      RSCLOS EQU    >04                 no file open error code
0363      0019      RSDATA EQU    >19                 data error
0364      0006      RSDEVI EQU    >06                 device error code
0365      0008      RSDLEN EQU    >08                 data/file too long error
0366      0051      RSDRCT EQU    >51                 no directory error code
0367      0007      RSEOF  EQU    >07                 EOF error code
0368      0050      RSEOT  EQU    >50                 wafer full error code
0369      000B      RSFILE EQU    >0B                 too many files error cod
0370      0003      RSFIND EQU    >03                 file not found error cod
0371      0052      RSLAST EQU    >52                 not last file for append
0372      0010      RSLOWB EQU    >10                 low battery warning code
0373      0000      RSOK   EQU    >00                 normal completion status
0374      0005      RSOPEN EQU    >05                 file already open error
0375      0009      RSPROT EQU    >09                 write protect error code
0376      000F      RSREAD EQU    >0F                 read in write-only mode
0377      000D      RSSUPP EQU    >0D                 unsupported command erro
0378      00FF      RSTIME EQU    >FF                 time-out error code
0379      0018      RSVERI EQU    >18                 verify error code
0380      000E      RSWRIT EQU    >0E                 write in read-only mode
```

```
0382                *-------------------------------------------------------------
0383                *  trap equates
0384      0004  ERRWT  EQU   >04          error handler
0385      0009  EOT    EQU   >09          find EOT
0386      000C  EOTCHK EQU   >0C          test for EOT
0387      0007  SCREWD EQU   >07          ampl defaults to trap 7
0388      0008  RCVCNT EQU   >08          receive PAB/data cont.
0389      000A  TIMEX  EQU   >0A          start timer operation
0390      000B  BITEST EQU   >0B          test bit counter
0391      0005  XMIT   EQU   >05          transmit response
0392      0006  XMTCNT EQU   >06          transmit response cont.
```

```
0394                    *---------------------------------------------------------------
0395                    * origin
0396 F006               AORG   >F006          the place where it all began
0397                    *---------------------------------------------------------------
0398                    * initialization
0399 F006   A2  INIT    MOVP   %INITDR,DRIVE          initialize drive
     F007   FF
     F008   06
0400 F009   B8          PUSH   A                      R2 is BITCONtcon (u code
0401 F00A   A2          MOVP   %STOP,TIMER            reset timer
     F00B   FF
     F00C   03
0402 F00D   72          MOV    %INIFLG,FFLAG          initialize no directory
     F00E   60
     F00F   1C
0403 F010   A2          MOVP   %I123C,IOCNTL          initialize INT1,2&3
     F011   6A
     F012   00
0404 F013   A2          MOVP   %DISABL,BCNTL          enable IBC
     F014   0C
     F015   81
0405                    * find out if this operation is necessary
0406 F016   05          EINT                          enable interrupts
0407 F017   A2  EOM     MOVP   %INHIB,BCNTL           enable IBC at SOM
     F018   04
     F019   81
0408 F01A   A2          MOVP   %STOP,DRIVE            turn off LED
     F01B   FF
     F01C   06
0409 F01D   A2          MOVP   %I1CS,IOCNTL           select & clear INT1
     F01E   43
     F01F   00
0410 F020   01          IDLE                          wait start of message
0411 F021   D5  SOM     CLR    COUNT                  device code length-1
     F022   12
0412 F023   72          MOV    %SAB,DATAP             data pointer
     F024   3E
     F025   19
0413 F026   8E          CALL   @RCVPAB                receive device code
     F027 F91A
0414 F029   B0          TSTA                          test for device 0
0415 F02A   E2          JEQ    ITSME
     F02B   0B
0416 F02C   B2          DEC    A                      decrement device code
0417 F02D   26          BTJO   %>F8,A,EOM             test for device 1 to 8
     F02E   F8
     F02F   E7
0418 F030   91          MOVP   TEST,B                 get device address
     F031   04
0419 F032   53          AND    %>07,B                 mask off 5 ms bits
     F033   07
0420 F034   6D          CMP    B,A                    compare 3 ls bits
0421 F035   E6          JNE    EOM                    test device code match
     F036   E0
0422 F037   A2  ITSME   MOVP   %WPSENS,DDRD           set D7 to output
     F038   80
     F039   0B
```

```
0423 F03A  A2          MOVP  %WPSN,WAFER          turn on sensor
     F03B  7F
     F03C  0A
0424 F03D  22          MOV   %WPDLAY,A            set up delay
     F03E  0A  14
0425 F03F  73          AND   %WPO,FFLAG           reset WP flag
     F040  BF
     F041  1C
0426 F042  BA  ME2     DJNZ  A,ME2                wait-out delay
     F043  FE
0427 F044  A7          BTJZP %WP,WAFER,ME3        test WP sensor
     F045  04
     F046  0A
     F047  03
0428 F048  74          OR    %FWP,FFLAG           set WP flag
     F049  40
     F04A  1C
0429 F04B  A2  ME3     MOVP  %INITWF,DDRD         set up D1 to output
     F04C  01
     F04D  0B
0430 F04E  72          MOV   %7,COUNT             rest of PAB length-1
     F04F  07
     F050  12
0431 F051  F7          TRAP  RCVCNT               receive rest of PAB
0432 F052  32          MOV   CCODE,B              index command tables
     F053  3D
0433 F054  5D          CMP   %CNULL,B             test for NULL command
     F055  FE
0434 F056  E2          JEQ   EOM
     F057  BF
0435 F058  5D          CMP   %CRESET,B            test for RESET command
     F059  FF
0436 F05A  E2          JEQ   XRESET
     F05B  73
0437 F05C  5D          CMP   %>0F,B               test for unsupported com
     F05D  0F
0438 F05E  E3          JHS   UNSUPP
     F05F  1D
0439 F060  CF          RLC   B                    prepare table index
0440 F061  AA          LDA   @CTABLE(B)           push address of command
     F062 F9C2
0441 F064  B8          PUSH  A
0442 F065  AA          LDA   @CTABLE+1(B)         OKCRAIG
     F066 F9C3
0443 F068  B8          PUSH  A
0444 F069  0A          RETS                       sneaky branch to command
```

```
0446                    *-------------------------------------------------------------
0447 F06A   D5  RETDL   CLR     DLEN                    DL = 0
     F06B   37
0448 F06C   D5          CLR     DLEN-1
     F06D   36
0449 F06E   72          MOV     %1,COUNT                response length-1
     F06F   01
     F070   12
0450 F071   72          MOV     %DLEN,DATAP             data pointer
     F072   37
     F073   19
0451 F074   F9  RETDL2  TRAP    XMTCNT                  return zero DL
0452 F075   D5  RTSTAT  CLR     COUNT                   status length-1
     F076   12
0453 F077   72  RTSTA2  MOV     %STATUS,DATAP           data pointer
     F078   3E
     F079   19
0454 F07A   F9          TRAP    XMTCNT                  return status
0455 F07B   E0  EOM2    JMP     EOM                     wait for SOM
     F07C   9A
```

```
0457                *-------------------------------------------------------
0458                * unsupported commands
0459 F07D   FB  UNSUPP TRAP    ERRWT                   error
0460 F07E   0D          BYTE    RSSUPP                  unsupported command code
0461                *-------------------------------------------------------
0462                * error handler  nice work Craig
0463 F07F   A2  WTERR   MOVP    %STOP,DRIVE             turn off drive
     F080   FF
     F081   06
0464 F082   A2          MOVP    %STOP,TIMER             stop timer
     F083   FF
     F084   03
0465 F085   C9          POP     B                       get pc lsb
0466 F086   B9          POP     A                       get pc msb
0467 F087   9A          LDA     *B                      get error code
     F088   01
0468 F089   D0          MOV     A,STATUS                store it
     F08A   3E
0469 F08B   52          MOV     %STACK-1,B              set up stack ptr
     F08C   02
0470 F08D   0D          LDSP
0471 F08E   8E          CALL    @RCVDMY                 discard rest of message
     F08F F973
0472 F091   E0          JMP     RETDL
     F092   D7
```

```
0474                *-------------------------------------------------------
0475                * test for valid directory in ram and file open
0476 F093   76   XOPNDR BTJO   %FDIREC,FFLAG,XOPND2  test for directory in ra
     F094   80
     F095   1C
     F096   02
0477 F097   FB   EDIREC TRAP   ERRWT
0478 F098   51          BYTE   RSDRCT                 no directory err
0479                *
0480 F099   76   XOPND2 BTJO   %FOPEN,FFLAG,XOPND3   test for file open
     F09A   10
     F09B   1C
     F09C   02
0481 F09D   FB   EOPEN  TRAP   ERRWT
0482 F09E   04          BYTE   RSCLOS                 file not open
0483                *
0484 F09F   8C   XOPND3 BR     @XCLOD8                2nd pass command decode
     F0A0 F239
```

```
0486                *-------------------------------------------------------
0487                * close file and delete if not write protected
0488 F0A2   76  XCLSDL BTJO   %FWP,FFLAG,XCLSD2     test write protect
     F0A3   40
     F0A4   1C
     F0A5   24
0489 F0A6   32         MOV    NFILE,B              current file number to B
     F0A7   23
0490                * DELETE command joins CLOSE & DELETE here
0491 F0A8   73  XDELET AND    %INACTV,RECFIL-1     inactivate, find last
     F0A9   7F
     F0AA   1D
0492 F0AB   8E         CALL   @STFILD
     F0AC F8E9
0493 F0AE   77         BTJZ   %LAST,RECFIL-1,XCLOWD test for not last file
     F0AF   40
     F0B0   1D
     F0B1   2E
0494 F0B2   73         AND    %LASTO,RECFIL-1      reset last file flag
     F0B3   BF
     F0B4   1D
0495 F0B5   8E         CALL   @STFIL2
     F0B6 F8EB
0496 F0B8   C1  XDELE1 TSTB                        test file 0
0497 F0B9   E2         JZ     XDELE2
     F0BA   08
0498 F0BB   5A         SUB    %4,B                 decrement index (x4)
     F0BC   04
0499 F0BD   AA         LDA    @FILEO(B)
     F0BE 0040
0500 F0C0   27         BTJZ   %ACTIVE,A,XDELE1     test for not active
     F0C1   80
     F0C2   F5
0501 F0C3   24  XDELE2 OR     %LAST,A              set last if active or 0
     F0C4   40
0502 F0C5   AB         STA    @FILEO(B)
     F0C6 0040
0503 F0C8   E0         JMP    XCLOWD               join CLOSE
     F0C9   16
0504 F0CA   73  XCLSD2 AND    %OPNRST,FFLAG        close file
     F0CB   EF
     F0CC   1C
0505 F0CD   FB         TRAP   ERRWT                error
0506 F0CE   09         BYTE   RSPROT               WP error
0507                *-------------------------------------------------------
0508                * bus reset command, all open files are closed
0509 F0CF   77  XRESET BTJZ   %FDROPN,FFLAG,EOM2   stop if no open file
     F0D0   90
     F0D1   1C
     F0D2   A8
0510                * CLOSE joins RESET command here
0511 F0D3   77  XCLOSE BTJZ   %TSTOA,ATTRIB,XCLOST test for output/append
     F0D4   40
     F0D5   33
     F0D6   04
0512 F0D7   77         BTJZ   %TSTI,ATTRIB,XCLOFL  test for input
     F0D8   80
```

```
      F0D9   33
      F0DA   08
0513  F0DB   32  XCLOST MOV     NFILE,B                 current file number to B
      F0DC   23
0514  F0DD   8E          CALL    @STFILD                 store file parameters
      F0DE F8E9
0515              * CLOSE & DELETE may join CLOSE and RESET commands here
0516  F0E0   8E  XCLOWD CALL    @WDIREC                 write directory
      F0E1 F4FC
0517  F0E3   73  XCLOFL AND     %OPNRST,FFLAG           reset OPEN flag
      F0E4   EF
      F0E5   1C
0518  F0E6   7D          CMP     %CRESET,CCODE           test for RESET command
      F0E7   FF
      F0E8   3D
0519  F0E9   E2          JEQ     EOM2
      F0EA   90
0520  F0EB   D5          CLR     STATUS                  store OK status
      F0EC   3E
0521  F0ED   8C          BR      @RETDL
      F0EE F06A
```

```
0523                *-------------------------------------------------------------
0524                * position record
0525 F0F0   76  XPOSIT BTJO   %TSTNA,ATTRIB,XPOSIO  test for not append
     F0F1   C0
     F0F2   33
     F0F3   02
0526 F0F4   FB         TRAP   ERRWT                 error
0527 F0F5   02         BYTE   RSATTR                attribute error code
0528 F0F6   8E  XPOSIO CALL   @SEARCH               find record or EOF
     F0F7 F7C4
0529 F0F9   77         BTJZ   %FFOUND,FFLAG,XVERIO  test for record not foun
     F0FA   08
     F0FB   1C
     F0FC   0A
0530 F0FD   8C         BR     @RETDL                return DL = 0 & status
     F0FE F06A
```

```
0532                *-------------------------------------------------------------
0533                * verify record
0534 F100   8E   XVERIF CALL   @SEARCH                  position to record
     F101 F7C4
0535 F103   76          BTJO   %FFOUND,FFLAG,XVERI1     test for record found
     F104   08
     F105   1C
     F106   02
0536 F107   FB   XVERIO TRAP   ERRWT                    error
0537 F108   03          BYTE   RSFIND                   file not found error
0538 F109   8E   XVERI1 CALL   @TSTEOF                  compare nrec,recfil
     F10A F8B2
0539 F10C   E6          JNE    XVERI2                   test for EOF
     F10D   02
0540 F10E   FB          TRAP   ERRWT                    error
0541 F10F   07          BYTE   RSEOF                    EOF error
0542 F110   D3   XVERI2 INC    NREC                     increment record number
     F111   22
0543 F112   79          ADC    %0,NREC-1
     F113   00
     F114   21
0544 F115   8E          CALL   @RSYNC                   read sync & file/record
     F116 F519
0545 F118   A2          MOVP   %HSKSET,BCNTL            let HSK float
     F119   00
     F11A   81
0546 F11B   8E          CALL   @RCOMPM                  compare file/record
     F11C F5AD
0547 F11E   8E          CALL   @RDL                     read DL
     F11F F5DD
0548                * compare DL to DLEN,DLEN-1
0549 F121   8E          CALL   @RCOMPB                  compare bus data
     F122 F6EC
0550 F124   E0          JMP    XREAD4
     F125   63
```

```
0552                *-------------------------------------------------------
0553                * receive bus & write wafer
0554 F126   77  XWRITE BTJZ   %TSTOA,ATTRIB,XWRIT2  test for output/append
     F127   40
     F128   33
     F129   06
0555 F12A   76         BTJO   %TSTU,ATTRIB,XWRIT2  test for update
     F12B   80
     F12C   33
     F12D   02
0556 F12E   FB         TRAP   ERRWT                error
0557 F12F   0E         BYTE   RSWRIT               write in read only mode
0558 F130   D3  XWRIT2 INC    NREC                 increment record number
     F131   22
0559 F132   79         ADC    %0,NREC-1
     F133   00
     F134   21
0560 F135   A2         MOVP   %HSKSET,BCNTL        let HSK float
     F136   00
     F137   81
0561 F138   8E         CALL   @WSYNC               write record header
     F139 F384
0562 F13B   8E         CALL   @WDL                 write DL
     F13C F413
0563 F13E   42         MOV    NREC,RECFIL          number of records
     F13F   22
     F140   1E
0564 F141   73         AND    %>F0,RECFIL-1
     F142   F0
     F143   1D
0565 F144   44         OR     NREC-1,RECFIL-1
     F145   21
     F146   1D
0566 F147   4D         CMP    DLEN-1,MAXLEN-1      maximum record length
     F148   36
     F149   1F
0567 F14A   E7         JL     XWRIT3               test for MRL LT DL
     F14B   07
0568 F14C   E6         JNE    RTDL2                test for MRL GT DL
     F14D   0B
0569 F14E   4D         CMP    DLEN,MAXLEN
     F14F   37
     F150   20
0570 F151   E3         JHS    RTDL2                test for MRL LE DL
     F152   06
0571 F153   42  XWRIT3 MOV    DLEN,MAXLEN
     F154   37
     F155   20
0572 F156   42         MOV    DLEN-1,MAXLEN-1
     F157   36
     F158   1F
0573 F159   D5  RTDL2  CLR    STATUS               return OK status
     F15A   3E
0574 F15B   C5         CLR    B
0575 F15C   72         MOV    %1,COUNT             DL length-1(-1 in wchks!
     F15D   01
     F15E   12
```

```
0576 F15F   72          MOV     %STATUS+1,DATAP     msb DL same as status
     F160   3F
     F161   19
0577 F162   8E          CALL    @WNXPA2             return rest of DL
     F163 F941
0578 F165   8C          BR      @RTSTAT             return status
     F166 F075
```

```
0580                *-------------------------------------------------------------
0581                * read wafer & transmit bus
0582 F168   76  XREAD  BTJO   %TSTIU,ATTRIB,XREAD2  test for input/update
     F169   40
     F16A   33
     F16B   02
0583 F16C   FB         TRAP   ERRWT                 error
0584 F16D   0F         BYTE   RSREAD                read in write only mode
0585 F16E   8E  XREAD2 CALL   @TSTEOF               compare nrec,recfil
     F16F F8B2
0586 F171   E6         JNE    XREAD3                test for EOF
     F172   02
0587 F173   FB         TRAP   ERRWT                 error
0588 F174   07         BYTE   RSEOF                 EOF error
0589                * if not at EOF, implies a write was not last if in update m
0590 F175   D3  XREAD3 INC    NREC                  increment record number
     F176   22
0591 F177   79         ADC    %0,NREC-1
     F178   00
     F179   21
0592 F17A   8E         CALL   @RSYNC                read sync & file/record
     F17B F519
0593 F17D   A2         MOVP   %HSKSET,BCNTL         let HSK float
     F17E   00
     F17F   81
0594 F180   8E         CALL   @RCOMPM               compare file/record
     F181 F5AD
0595 F183   8E         CALL   @RDLX                 read DL & transmit bus
     F184 F606
0596 F186   8E         CALL   @RNIB                 read wafer & transmit bu
     F187 F679
0597 F189   32  XREAD4 MOV    STATUS,B              get status of read
     F18A   3E
0598 F18B   D5         CLR    COUNT                 status length -1
     F18C   12
0599                * DATAP wont be used
0600 F18D   8E         CALL   @WNXPA2               return rest of status
     F18E F941
0601 F190   8C         BR     @EOM                  response
     F191 F017
```

```
0603                    *--------------------------------------------------------------
0604                    * return status
0605 F193   72  XRSTAT MOV     %>07,DATA               seq,storage,r/w
     F194   07
     F195   1A
0606 F196   77         BTJZ    %FWP,FFLAG,XRSTA1       test for WP
     F197   40
     F198   1C
     F199   03
0607 F19A   74         OR      %>20,DATA               set WP flag
     F19B   20
     F19C   1A
0608 F19D   77  XRSTA1 BTJZ    %FOPEN,FFLAG,XRSTA2     test for open file
     F19E   10
     F19F   1C
     F1A0   0A
0609 F1A1   74         OR      %>10,DATA               set open flag
     F1A2   10
     F1A3   1A
0610 F1A4   77         BTJZ    %EOFFLG,FFLAG,XRSTA2    test for EOF
     F1A5   02
     F1A6   1C
     F1A7   03
0611 F1A8   74         OR      %>80,DATA               set EOF flag
     F1A9   80
     F1AA   1A
0612 F1AB   72  XRSTA2 MOV     %>01,DLEN               DL = 1
     F1AC   01
     F1AD   37
0613 F1AE   D5         CLR     DLEN-1
     F1AF   36
0614 F1B0   72         MOV     %DLEN,DATAP             set up data pointer
     F1B1   37
     F1B2   19
0615 F1B3   72         MOV     %1,COUNT                DL length-1
     F1B4   01
     F1B5   12
0616 F1B6   F9         TRAP    XMTCNT                  return DL
0617 F1B7   72         MOV     %DATA,DATAP             set up data pointer
     F1B8   1A
     F1B9   19
0618 F1BA   D5         CLR     COUNT                   device status length-1
     F1BB   12
0619 F1BC   F9         TRAP    XMTCNT                  return device status
0620 F1BD   D5         CLR     STATUS                  OK status
     F1BE   3E
0621 F1BF   8C         BR      @RTSTAT
     F1C0 F075
```

```
0623                    *-------------------------------------------------------
0624                    * read buffer length & attributes for OPEN only
0625 F1C2   7A  XCLODO SUB     %3,DLEN                  decrement DLEN by 3
     F1C3   03
     F1C4   37
0626 F1C5   E3         JHS     XCLODO                   test for DL >=3
     F1C6   02
0627 F1C7   FB  XCLODX TRAP    ERRWT                    error
0628 F1C8   01         BYTE    RSCHAR                   characteristics error
0629 F1C9   72  XCLODO MOV     %BLOPEN,DATAP            data pointer
     F1CA   35
     F1CB   19
0630 F1CC   72         MOV     %2,COUNT                 DL of BL & attributes-1
     F1CD   02
     F1CE   12
0631 F1CF   F7         TRAP    RCVCNT                   receive BL & attributes
0632                    * DELETE joins OPEN at this point
0633 F1D0   42  XCLODD MOV     DLEN,COUNT               move DL to count
     F1D1   37
     F1D2   12
0634 F1D3   E2         JZ      XCLODX                   test for no filename
     F1D4   F2
0635 F1D5   76         BTJO    %>FF,DLEN-1,XCLODX       test for DL too big
     F1D6   FF
     F1D7   36
     F1D8   EE
0636 F1D9   52         MOV     %>0F,B                   blank filename
     F1DA   0F
0637 F1DB   22         MOV     %BLANK,A
     F1DC   20
0638 F1DD   AB  XCLOD1 STA     @FNAME1->0F(B)           store blank
     F1DE 0023
0639 F1E0   CA         DJNZ    B,XCLOD1
     F1E1   FB
0640 F1E2   76         BTJO    %>F0,COUNT,XCLODY        test for filename too lo
     F1E3   F0
     F1E4   12
     F1E5   05
0641 F1E6   D2         DEC     COUNT                    decrement DL (entry poin
     F1E7   12
0642 F1E8   F7         TRAP    RCVCNT                   receive filename
0643 F1E9   E0         JMP     XCLOD2
     F1EA   04
0644 F1EB   72  XCLODY MOV     %>0E,COUNT               DL too long
     F1EC   0E
     F1ED   12
0645 F1EE   F7         TRAP    RCVCNT                   receive 15 char filename
0646                    * call rcvdmy elsewhere, or begin response here (HSK release
0647                    *       CALL   @RCVDMY            truncate filename
0648 F1EF   74  XCLOD2 OR      %FNAME,FFLAG             set filename flag
     F1F0   20
     F1F1   1C
0649 F1F2   76         BTJO    %>E0,FNAME1,XCLODR       test for file name/numbe
     F1F3   E0
     F1F4   32
     F1F5   03
0650                    * CATALOG joins OPEN and DELETE at this point
```

```
0651 F1F6   73  XCLODC AND    %FNUMBR,FFLAG          reset file number flag
     F1F7   DF
     F1F8   1C
0652            * test valid directory; if not read it, test for no open
0653 F1F9   76  XCLODR BTJO    %FDIREC,FFLAG,XCLODF   test for directory in ra
     F1FA   80
     F1FB   1C
     F1FC   03
0654 F1FD   8E         CALL    @RDIREC                read directory
     F1FE F76D
0655            * FORMAT joins OPEN, DELETE, and CATALOG at this point
0656 F200   77  XCLODF BTJZ    %FDROPN,FFLAG,XCLOD3   test for no file open
     F201   90
     F202   1C
     F203   02
0657 F204   FB         TRAP    ERRWT                  error
0658 F205   05         BYTE    RSOPEN                 file open error
0659 F206   32  XCLOD3 MOV     CCODE,B                move ccode to B
     F207   3D
0660 F208   5D         CMP     %CCATAL,B              test for CATALOG
     F209   0E
0661 F20A   E2         JEQ     XCLOD6
     F20B   18
0662 F20C   5D         CMP     %COPEN,B               test for OPEN command
     F20D   00
0663 F20E   E6         JNE     XCLOD5
     F20F   0E
0664 F210   77         BTJZ    %SEQUEN,ATTRIB,XCLOD4  test for sequential file
     F211   20
     F212   33
     F213   02
0665 F214   FB         TRAP    ERRWT
0666 F215   02         BYTE    RSATTR                 attribute error
0667 F216   77  XCLOD4 BTJZ    %TSTOA,ATTRIB,XCLOD5   test for output/append
     F217   40
     F218   33
     F219   04
0668 F21A   77         BTJZ    %TSTI,ATTRIB,XCLOD6    test for input
     F21B   80
     F21C   33
     F21D   06
0669 F21E   77  XCLOD5 BTJZ    %FWP,FFLAG,XCLOD6      test for WP
     F21F   40
     F220   1C
     F221   02
0670 F222   FB         TRAP    ERRWT
0671 F223   09         BYTE    RSPROT                 write protect err
0672 F224   5D  XCLOD6 CMP     %CFORMA,B              test for FORMAT command
     F225   0D
0673 F226   E2         JEQ     XCLOD7
     F227   0F
0674 F228   8E         CALL    @POSITN                search for file (no erro
     F229 F805
0675 F22B   32         MOV     CCODE,B                move command code to B
     F22C   3D
0676 F22D   5D         CMP     %COPEN,B               test for OPEN command
     F22E   00
```

```
0677 F22F   E2          JEQ    XCLOD7
     F230   06
0678 F231   76          BTJO   %FFOUND,FFLAG,XCLOD7   test for file found
     F232   08
     F233   1C
     F234   02
0679 F235   FB          TRAP   ERRWT
0680 F236   03          BYTE   RSFIND                  file not found err
0681 F237   B0  XCLOD7  CLRC                           clear carry for shift
0682 F238   CF          RLC    B                       prepare index
0683 F239   AA  XCLOD8  LDA    @CTABLX(B)              push address of command
     F23A F9E0
0684 F23C   B8          PUSH   A
0685 F23D   AA          LDA    @CTABLX+1(B)
     F23E F9E1
0686 F240   B8          PUSH   A
0687 F241   0A          RETS                           sneaky branch
```

```
0689                    *-------------------------------------------------------
0690                    * open command
0691 F242   77  XOPEN   BTJZ    %FFOUND,FFLAG,XOPNEW   test for file not found
     F243   08
     F244   1C
     F245   03
0692 F246   8C          BR      @XOPEN4
     F247 F2D5
0693                    * if file not found
0694 F249   77  XOPNEW  BTJZ    %TSTOA,ATTRIB,XOPNOO   test for append/output
     F24A   40
     F24B   33
     F24C   06
0695 F24D   FB  XOPENO  TRAP    ERRWT                  error
0696 F24E   03          BYTE    RSFIND                 file not found
0697                    * attempt to OPEN in output mode
0698 F24F   76  XOPENO  BTJO    %LAST,RECFIL-1,XOPNO4  to OPEN last for output
     F250   40
     F251   1D
     F252   44
0699 F253   77  XOPNOO  BTJZ    %LAST,FILEF,XOPNO1     test file capacity
     F254   40
     F255   7C
     F256   02
0700 F257   FB          TRAP    ERRWT                  error
0701 F258   0B          BYTE    RSFILE                 too many files
0702 F259   42  XOPNO1  MOV     NFILE,TEMP4            remember present file #
     F25A   23
     F25B   3A
0703 F25C   8E          CALL    @FNDEOD                find end of data
     F25D F865
0704 F25F   77          BTJZ    %FFOUND,FFLAG,XOPNO5   test for no active files
     F260   08
     F261   1C
     F262   02
0705 F263   D3          INC     NFILE                  next file number
     F264   23
0706                    * write file header (if EOT, no return)
0707 F265   72  XOPNO5  MOV     %>0F,DLEN              filename length
     F266   0F
     F267   37
0708 F268   D5          CLR     NREC                   file header record #
     F269   22
0709 F26A   D5          CLR     NREC-1
     F26B   21
0710 F26C   8E          CALL    @WSYNC                 write record header
     F26D F384
0711 F26F   72          MOV     %FNAME1,DATAP          data pointer
     F270   32
     F271   19
0712 F272   8E          CALL    @WMEM                  write filename & checksu
     F273 F513
0713                    * reset old file active/last flag if neccessary
0714 F275   32          MOV     TEMP4,B                recall past file
     F276   3A
0715 F277   8E          CALL    @LDFILD                recall file data
     F278 F900
```

```
0716 F27A   77          BTJZ   %LAST,RECFIL-1,XOPNO2 test for not last file
     F27B   40
     F27C   1D
     F27D   05
0717 F27E   73          AND    %LASTO,RECFIL-1       reset last flag
     F27F   BF
     F280   1D
0718 F281   E0          JMP    XOPNO3
     F282   03
0719 F283   73  XOPNO2  AND    %INACTV,RECFIL-1      reset active flag
     F284   7F
     F285   1D
0720 F286   8E  XOPNO3  CALL   @STFIL2               save past file
     F287 F8EB
0721 F289   32          MOV    NFILE,B               get current file number
     F28A   23
0722 F28B   C2          DEC    B                     get previous file #
0723 F28C   E7          JNC    XOPNO4                test for B not >FF
     F28D   09
0724 F28E   8E          CALL   @LDFILD               get file data
     F28F F900
0725 F291   73          AND    %LASTO,RECFIL-1       reset last flag
     F292   BF
     F293   1D
0726 F294   8E          CALL   @STFIL2               put file data
     F295 F8EB
0727                * OPEN existing last file for output joins here
0728 F297   72  XOPNO4 MOV     %NEWFIL,RECFIL-1      new file (active/last)
     F298   C0
     F299   1D
0729 F29A   D5          CLR    RECFIL
     F29B   1E
0730 F29C   D5          CLR    MAXLEN-1
     F29D   1F
0731 F29E   D5          CLR    MAXLEN
     F29F   20
0732 F2A0   77          BTJZ   %INTDIS,ATTRIB,XOPEN1 test for display format
     F2A1   08
     F2A2   33
     F2A3   03
0733 F2A4   74          OR     %INTERN,RECFIL-1      set internal file type
     F2A5   10
     F2A6   1D
0734 F2A7   76  XOPEN1 BTJO    %>FF,BLOPEN-1,XOPEN2  test for BL > 0
     F2A8   FF
     F2A9   34
     F2AA   06
0735 F2AB   76          BTJO   %>FF,BLOPEN,XOPEN2    test for BL > 0
     F2AC   FF
     F2AD   35
     F2AE   02
0736 F2AF   D3          INC    BLOPEN-1              256 -> BL
     F2B0   34
0737 F2B1   74  XOPEN2 OR      %NUFLG1,FFLAG         set up open flags
     F2B2   94
     F2B3   1C
0738 F2B4   73          AND    %NUFLG2,FFLAG         WP flag may be set
```

```
      F2B5   FC
      F2B6   1C
0739  F2B7   D5          CLR     STATUS
      F2B8   3E
0740  F2B9   72  XOPEN3  MOV     %4,DLEN                 prepare to send DL
      F2BA   04
      F2BB   37
0741  F2BC   D5          CLR     DLEN-1
      F2BD   36
0742  F2BE   72          MOV     %1,COUNT                DL length-1
      F2BF   01
      F2C0   12
0743  F2C1   72          MOV     %DLEN,DATAP
      F2C2   37
      F2C3   19
0744  F2C4   F9          TRAP    XMTCNT                  return DL
0745  F2C5   72          MOV     %1,COUNT                BL length -1
      F2C6   01
      F2C7   12
0746  F2C8   72          MOV     %BLOPEN,DATAP           (ms byte already 0)
      F2C9   35
      F2CA   19
0747  F2CB   F9          TRAP    XMTCNT                  return BL
0748  F2CC   72  RTNREC  MOV     %1,COUNT                RN length -1
      F2CD   01
      F2CE   12
0749  F2CF   72          MOV     %NREC,DATAP             (ms byte already 0)
      F2D0   22
      F2D1   19
0750  F2D2   8C          BR      @RETDL2                 return BL & status
      F2D3 F074
0751                  * if file found
0752  F2D5   76  XOPEN4  BTJO    %TSTIU,ATTRIB,XOPEN9    test for input/update
      F2D6   40
      F2D7   33
      F2D8   07
0753  F2D9   77          BTJZ    %TSTO,ATTRIB,XOPEN9     test for not output
      F2DA   80
      F2DB   33
      F2DC   03
0754  F2DD   8C          BR      @XOPEN0                 open for output
      F2DE F24F
0755  F2E0   12  XOPEN9  MOV     ATTRIB,A                move attributes to A
      F2E1   33
0756  F2E2   BE          RL      A                       shift file type to bit 4
0757  F2E3   15          XOR     RECFIL-1,A              compare file type
      F2E4   1D
0758  F2E5   27          BTJZ    %DISPLY,A,XOPEN8        test for same value
      F2E6   10
      F2E7   02
0759  F2E8   FB          TRAP    ERRWT                   error
0760  F2E9   02          BYTE    RSATTR                  attribute error code
0761  F2EA   76  XOPEN8  BTJO    %TSTIU,ATTRIB,XOPEN7    test for input/update
      F2EB   40
      F2EC   33
      F2ED   0F
0762                  * if append or update
```

```
0763 F2EE   76  XOPEN5 BTJO   %LAST,RECFIL-1,XOPEN6 test for last file
     F2EF   40
     F2F0   1D
     F2F1   02
0764 F2F2   FB         TRAP   ERRWT                  error
0765 F2F3   52         BYTE   RSLAST                 not last file for append
0766 F2F4   76  XOPEN6 BTJO   %TSTU,ATTRIB,XOPNIU    test for update
     F2F5   80
     F2F6   33
     F2F7   09
0767                   * OPEN for append
0768 F2F8   8E         CALL   @FNDEOD                append at EOF
     F2F9 F865
0769 F2FB   E0         JMP    XOPEN1
     F2FC   AA
0770                   * OPEN for input or update
0771 F2FD   76  XOPEN7 BTJO   %TSTU,ATTRIB,XOPEN5    test for update
     F2FE   80
     F2FF   33
     F300   ED
0772 F301   76  XOPNIU BTJO   %>FF,BLOPEN-1,XOPNIO   test for BL > 0
     F302   FF
     F303   34
     F304   0C
0773 F305   76         BTJO   %>FF,BLOPEN,XOPNIO     test for BL > 0
     F306   FF
     F307   35
     F308   08
0774 F309   42         MOV    MAXLEN,BLOPEN          return MRL
     F30A   20
     F30B   35
0775 F30C   42         MOV    MAXLEN-1,BLOPEN-1
     F30D   1F
     F30E   34
0776 F30F   E0         JMP    XOPEN1                 go test for BL = 0
     F310   96
0777 F311   4D  XOPNIO CMP    BLOPEN-1,MAXLEN-1
     F312   34
     F313   1F
0778 F314   E7         JL     XOPEN2                 test for MRL LT BL
     F315   9B
0779 F316   E6         JNE    XOPNI1                 test for MRL GT BL
     F317   05
0780 F318   4D         CMP    MAXLEN,BLOPEN
     F319   20
     F31A   35
0781 F31B   E3         JHS    XOPEN2                 test for MRL GE BL
     F31C   94
0782 F31D   72  XOPNI1 MOV    %RSBLEN,STATUS
     F31E   0C
     F31F   3E
0783 F320   42         MOV    MAXLEN,BLOPEN          return MRL with error
     F321   20
     F322   35
0784 F323   42         MOV    MAXLEN-1,BLOPEN-1
     F324   1F
     F325   34
```

```
0785 F326   E0          JMP     XOPEN3
     F327   91
```

```
0787                *-------------------------------------------------------
0788                * delete file command is a subset of CLOSE & DELETE
0789                *-------------------------------------------------------
0790                * format wafer
0791 F328   B5   XFORMA CLR    A                    initialize directory
0792 F329   52         MOV    %>41,B               set up index
     F32A   41
0793 F32B   AB   XFORM1 STA    @>3E(B)              no last/active files,REV
     F32C 003E
0794 F32E   CA         DJNZ   B,XFORM1
     F32F   FB
0795 F330   72         MOV    %DFORMA,FILEO        inactive/last
     F331   40
     F332   40
0796 F333   8E         CALL   @WDIREC              write directory
     F334 F4FC
0797                * add format pattern write and read
0798 F336   72         MOV    %RSOK,STATUS
     F337   00
     F338   3E
0799 F339   8C         BR     @RETDL
     F33A F06A
```

```
0801                *-------------------------------------------------------
0802                * catalog command, return current file parameters
0803 F33C   76   XCATAL BTJO   %>FF,BLEN-1,XCATA2     test for adequate BLEN
     F33D   FF
     F33E   38
     F33F   07
0804 F340   7D          CMP    %22,BLEN
     F341   16
     F342   39
0805 F343   E3          JHS    XCATA2
     F344   02
0806 F345   FB          TRAP   ERRWT
0807 F346   0C          BYTE   RSBLEN                BLEN error
0808                *
0809 F347   72   XCATA2 MOV    %22,DLEN              prepare response DL
     F348   16
     F349   37
0810 F34A   D5          CLR    DLEN-1
     F34B   36
0811 F34C   72          MOV    %DLEN,DATAP           set up DL pointer
     F34D   37
     F34E   19
0812 F34F   D5          CLR    DATAP-1
     F350   18
0813 F351   72          MOV    %2,COUNT              length of DL
     F352   02
     F353   12
0814 F354   FA          TRAP   XMIT                  return DL
0815                * check on the order in which data appears in data buffer
0816 F355   72          MOV    %NFILE,DATAP          prepare to send file num
     F356   23
     F357   19
0817 F358   D5          CLR    COUNT                 file number length-1
     F359   12
0818 F35A   F9          TRAP   XMTCNT                return file number
0819                *
0820 F35B   72          MOV    %FNAME1,DATAP         prepare to send file nam
     F35C   32
     F35D   19
0821 F35E   72          MOV    %NMLEN-1,COUNT        file name length-1
     F35F   0E
     F360   12
0822 F361   F9          TRAP   XMTCNT                return file name
0823                *
0824 F362   42          MOV    RECFIL-1,TEMP1        prepare to send file sta
     F363   1D
     F364   3D
0825 F365   73          AND    %LSN,RECFIL-1
     F366   0F
     F367   1D
0826 F368   72          MOV    %MAXLEN,DATAP
     F369   20
     F36A   19
0827 F36B   72          MOV    %3,COUNT              file status length-1
     F36C   03
     F36D   12
0828 F36E   F9          TRAP   XMTCNT                return number of records
```

```
0829                    *
0830 F36F   42          MOV     TEMP1,RECFIL-1
     F370   3D
     F371   1D
0831 F372   73          AND     %>E0,TEMP1          prepare to send flags
     F373   E0
     F374   3D
0832 F375   D5          CLR     TEMP2
     F376   3C
0833 F377   72          MOV     %1,COUNT            flags length-1
     F378   01
     F379   12
0834 F37A   72          MOV     %TEMP1,DATAP
     F37B   3D
     F37C   19
0835 F37D   F9          TRAP    XMTCNT              return flags
0836                    *
0837 F37E   72          MOV     %RSOK,STATUS        store OK status
     F37F   00
     F380   3E
0838 F381   8C  XCATA3  BR      @RTSTAT             return status
     F382 F075
```

```
0840                    *--------------------------------------------------------------
0841                    * write sync pattern
0842                    * XWRITE and OPEN enter here
0843 F384   F5  WSYNC   TRAP    TIMEX                  start motor & timer
0844 F385   72          MOV     %>01,EOTIMR            immeadiate EOT test
     F386   01
     F387   1B
0845 F388   72          MOV     %>02,NIBCON-1          512 nibbles of sync
     F389   02
     F38A   11
0846                    * (256 ms minimum delay inherent)
0847 F38B   76          BTJO    %>FF,NFILE,WSYNCO      test for file # > 0
     F38C   FF
     F38D   23
     F38E   0A
0848 F38F   76          BTJO    %>FF,NREC,WSYNCO       test for record # > 0
     F390   FF
     F391   22
     F392   06
0849 F393   76          BTJO    %>FF,NREC-1,WSYNCO     test for record # > 0
     F394   FF
     F395   21
     F396   02
0850 F397   DE          RL      NIBCON-1               extra length sync (0,0)
     F398   11
0851                    * 1024 nibbles of sync (512 ms delay inherent)
0852                    * WDIREC enters here
0853 F399   D5  WSYNCO CLR     NIBCON                 ls byte of NIBCON
     F39A   12
0854 F39B   73          AND     %FMSSNG,FFLAG          reset file-found flag
     F39C   F7
     F39D   1C
0855 F39E   72          MOV     %NFILE,DATAP           file & record numbers
     F39F   23
     F3A0   19
0856 F3A1   D5          CLR     CHKSUM-1               initialize checksum
     F3A2   13
0857 F3A3   D5          CLR     CHKSUM
     F3A4   14
0858 F3A5   76  WSYNC1 BTJO    %>FF,BITCON,WSYNC1     wait end of nibble
     F3A6   FF
     F3A7   02
     F3A8   FC
0859 F3A9   72          MOV     %>08,BITCON            restart bit-count
     F3AA   08
     F3AB   02
0860 F3AC   52          MOV     %SYNCDT,B              sync data
     F3AD   AA
0861 F3AE   DA          DJNZ    EOTIMR,WSYNC3          check to test EOT
     F3AF   1B
     F3B0   12
0862 F3B1   A7          BTJZP   %SENSET,DRIVE,WSYNC2   check to test EOT
     F3B2   04
     F3B3   06
     F3B4   07
0863 F3B5   D3          INC     EOTIMR
     F3B6   1B
```

```
0864 F3B7  A2          MOVP   %MTSNWE,DRIVE          turn on sensor
     F3B8  F1
     F3B9  06
0865 F3BA  E0          JMP    WSYNC4
     F3BB  0A
0866 F3BC  A6  WSYNC2  BTJOP  %EOTTST,WAFER,WEOT     test for EOT
     F3BD  02
     F3BE  0A
     F3BF  4F
0867 F3C0  72          MOV    %EOTCNT,EOTIMR
     F3C1  46
     F3C2  1B
0868 F3C3  A2  WSYNC3  MOVP   %MTWE,DRIVE           turn on motor & WE
     F3C4  F5
     F3C5  06
0869 F3C6  D2  WSYNC4  DEC    NIBCON                dec nibble counter
     F3C7  12
0870 F3C8  7B          SBB    %0,NIBCON-1
     F3C9  00
     F3CA  11
0871 F3CB  E3          JC     WSYNC1                test for end data
     F3CC  D8
0872 F3CD  B5          CLR    A                     clear A (start nibble/sp
0873                 * start nibble and spacer will be written in WBYTE
0874 F3CE  72          MOV    %>03,COUNT            file/record length
     F3CF  03
     F3D0  12
```

```
0876                *-------------------------------------------------------
0877                * load and write wafer
0878 F3D1   D5  WBYTE  CLR   DATAP-1                  keep in reg file
     F3D2   18
0879 F3D3   76  WBYTEO BTJO  %>FF,BITCON,WBYTEO       wait end of byte
     F3D4   FF
     F3D5   02
     F3D6   FC
0880 F3D7   72         MOV   %>10,BITCON              restart bit-count
     F3D8   10
     F3D9   02
0881 F3DA   C0         MOV   A,B                      move data in queue
0882                * first time : writes preset value of A (start/spacer)
0883 F3DB   9A         LDA   *DATAP                   load data
     F3DC   19
0884 F3DD   D2         DEC   DATAP                    dec data pointer
     F3DE   19
0885 F3DF   48         ADD   A,CHKSUM                 add to checksum (byte)
     F3E0   00
     F3E1   14
0886 F3E2   79         ADC   %0,CHKSUM-1
     F3E3   00
     F3E4   13
0887 F3E5   DA         DJNZ  EOTIMR,WBYTE2            check to test EOT
     F3E6   1B
     F3E7   12
0888 F3E8   A7         BTJZP %SENSET,DRIVE,WBYTE1     check to test EOT
     F3E9   04
     F3EA   06
     F3EB   07
0889 F3EC   D3         INC   EOTIMR
     F3ED   1B
0890 F3EE   A2         MOVP  %MTSNWE,DRIVE            turn on sensor
     F3EF   F1
     F3F0   06
0891 F3F1   E0         JMP   WBYTE3
     F3F2   0A
0892 F3F3   A6  WBYTE1 BTJOP %EOTTST,WAFER,WEOT       test for EOT
     F3F4   02
     F3F5   0A
     F3F6   18
0893 F3F7   72         MOV   %EOTCN2,EOTIMR
     F3F8   23
     F3F9   1B
0894 F3FA   A2  WBYTE2 MOVP  %MTWE,DRIVE              turn on motor & WE
     F3FB   F5
     F3FC   06
0895 F3FD   DA  WBYTE3 DJNZ  COUNT,WBYTEO              dec byte counter
     F3FE   12
     F3FF   D3
0896 F400   76  WBYTE4 BTJO  %>FF,BITCON,WBYTE4       wait end of byte
     F401   FF
     F402   02
     F403   FC
0897 F404   72         MOV   %>10,BITCON              restart bit-count
     F405   10
     F406   02
```

```
0898 F407   C0          MOV     A,B                     move data in queue
0899                * start writing last data byte, spacer started elsewhere
0900 F408   42          MOV     DLEN,NIBCON             move DL to nibcon
     F409   37
     F40A   12
0901 F40B   42          MOV     DLEN-1,NIBCON-1
     F40C   36
     F40D   11
0902 F40E   0A          RETS
0903                *-------------------------------------------------------
0904 F40F   FB  WEOT    TRAP    ERRWT                   error
0905 F410   50          BYTE    RSEOT                   wafer full
0906                *-------------------------------------------------------
0907 F411   FB  WHELP   TRAP    ERRWT                   message terminated/slow
0908 F412   FF          BYTE    RSTIME                  bus error
```

```
0910                *-------------------------------------------------------
0911 F413  76  WDL     BTJO    %>FF,BITCON,WDL       wait for end of byte
     F414  FF
     F415  02
     F416  FC
0912 F417  72          MOV     %>10,BITCON           restart byte bit-counter
     F418  10
     F419  02
0913                * writes byte spacer to be picked up at end of RBYTE
0914 F41A  72          MOV     %8,DLPCNT             non-linear dl counter
     F41B  08
     F41C  19
0915 F41D  48          ADD     DLEN,CHKSUM           add to checksum
     F41E  37
     F41F  14
0916 F420  79          ADC     %0,CHKSUM-1
     F421  00
     F422  13
0917 F423  48          ADD     DLEN-1,CHKSUM
     F424  36
     F425  14
0918 F426  79          ADC     %0,CHKSUM-1
     F427  00
     F428  13
0919 F429  E0          JMP     WDL1
     F42A  08
0920 F42B  76  WDL0    BTJO    %>FF,BITCON,WDL0      wait for end of nibble
     F42C  FF
     F42D  02
     F42E  FC
0921 F42F  72          MOV     %>8,BITCON            restart nibble bit-count
     F430  08
     F431  02
0922 F432  C0          MOV     A,B                   move data through queue
0923 F433  76  WDL1    BTJO    %>C,DLPCNT,WDL2       test for next dl byte
     F434  0C
     F435  19
     F436  04
0924 F437  12          MOV     DLEN-1,A              get ms byte
     F438  36
0925 F439  E0          JMP     WDL3
     F43A  05
0926 F43B  12  WDL2    MOV     DLEN,A                get ls byte
     F43C  37
0927 F43D  7A          SUB     %2,DLPCNT             non-linear adjust
     F43E  02
     F43F  19
0928 F440  77  WDL3    BTJZ    %1,DLPCNT,WDL4        test for ls/ms nibble
     F441  01
     F442  19
     F443  01
0929 F444  B7          SWAP    A                     ms nibble
0930 F445  DA  WDL4    DJNZ    EOTIMR,WDL6           check to test EOT
     F446  1B
     F447  12
0931 F448  A7          BTJZP   %SENSET,DRIVE,WDL5    check to test EOT
     F449  04
```

```
      F44A  06
      F44B  07
0932  F44C  D3          INC     EOTIMR
      F44D  1B
0933  F44E  A2          MOVP    %MTSNWE,DRIVE           turn on sensor
      F44F  F1
      F450  06
0934  F451  E0          JMP     WDL7
      F452  0A
0935  F453  A6  WDL5    BTJOP   %EOTTST,WAFER,WEOT      test for EOT
      F454  02
      F455  0A
      F456  B8
0936  F457  72          MOV     %EOTCNT,EOTIMR          restart EOT test counter
      F458  46
      F459  1B
0937  F45A  A2  WDL6    MOVP    %MTWE,DRIVE             turn on motor & WE
      F45B  F5
      F45C  06
0938  F45D  DA  WDL7    DJNZ    DLPCNT,WDL0             test for end of dl
      F45E  19
      F45F  CB
0939  F460  76  WDL8    BTJO    %>FF,BITCON,WDL8        wait end of nibble
      F461  FF
      F462  02
      F463  FC
0940  F464  72          MOV     %>10,BITCON             restart bit counter
      F465  10
      F466  02
0941                    * double count for DL ms nibble & 1 spacer nibble
0942                    * 2nd spacer will be written in WNIB
0943  F467  C0          MOV     A,B                     move data through queue
```

```
0945                *-------------------------------------------------------
0946                * receive bus & write wafer
0947 F468  DF  WNIB   RLC    NIBCON                  multiply nibcon by 2
     F469  12
0948 F46A  DF         RLC    NIBCON-1                (byte count -> nibble co
     F46B  11
0949 F46C  E0         JMP    WNIB3
     F46D  38
0950 F46E  76  WNIBO  BTJO   %>FF,BITCON,WNIBO       wait end of nibble
     F46F  FF
     F470  02
     F471  FC
0951 F472  72         MOV    %>08,BITCON             restart bit-count
     F473  08
     F474  02
0952 F475  C0         MOV    A,B                     move data in queue
0953                * first time : writes 2nd spacer after DL
0954 F476  A7         BTJZP  %BAV,BSTAT,WHELP        test for BAV active
     F477  02
     F478  81
     F479  97
0955 F47A  A7         BTJZP  %IRQ,BSTAT,WHELP        test for bus data ready
     F47B  08
     F47C  81
     F47D  93
0956 F47E  A2         MOVP   %RELEAS,BCNTL           reset IRQ
     F47F  01
     F480  81
0957 F481  80         MOVP   BDATA,A                 read data from bus
     F482  80
0958 F483  A2         MOVP   %HSKSET,BCNTL           let hsk float
     F484  00
     F485  81
0959 F486  23         AND    %LSN,A                  add to checksum
     F487  0F
0960 F488  48         ADD    A,CHKSUM
     F489  00
     F48A  14
0961 F48B  79         ADC    %0,CHKSUM-1
     F48C  00
     F48D  13
0962 F48E  DA         DJNZ   EOTIMR,WNIB2            check to test EOT
     F48F  1B
     F490  12
0963 F491  A7         BTJZP  %SENSET,DRIVE,WNIB1     check to test EOT
     F492  04
     F493  06
     F494  07
0964 F495  D3         INC    EOTIMR
     F496  1B
0965 F497  A2         MOVP   %MTSNWE,DRIVE           turn on sensor
     F498  F1
     F499  06
0966 F49A  E0         JMP    WNIB3
     F49B  0A
0967 F49C  A6  WNIB1  BTJOP  %EOTTST,WAFER,WEOT2     test for EOT
     F49D  02
```

```
        F49E   0A
        F49F   5A
0968    F4A0   72          MOV     %EOTCNT,EOTIMR
        F4A1   46
        F4A2   1B
0969    F4A3   A2  WNIB2   MOVP    %MTWE,DRIVE              turn on motor & WE
        F4A4   F5
        F4A5   06
0970    F4A6   D2  WNIB3   DEC     NIBCON                   dec nibble counter
        F4A7   12
0971    F4A8   7B          SBB     %0,NIBCON-1
        F4A9   00
        F4AA   11
0972    F4AB   E3          JC      WNIBO                    test for end data
        F4AC   C1
0973               * last data byte and spacer will be written in WCHKSM
```

```
0975                    *--------------------------------------------------------
0976                    * write checksum, turn off drive and wait for stop
0977 F4AD   72  WCHKSM MOV     %CHKSUM,DATAP          set up data pointer
     F4AE   14
     F4AF   19
0978 F4B0   D5          CLR     DATAP-1
     F4B1   18
0979 F4B2   72          MOV     %2,COUNT               set up length
     F4B3   02
     F4B4   12
0980 F4B5   76  WCHKS0 BTJO    %>FF,BITCON,WCHKS0     wait end of byte
     F4B6   FF
     F4B7   02
     F4B8   FC
0981 F4B9   72          MOV     %>10,BITCON            restart bit-counter
     F4BA   10
     F4BB   02
0982 F4BC   C0          MOV     A,B                    move data in queue
0983                    * first time : writes preset value of A (bus datat/spacer)
0984 F4BD   9A          LDA     *DATAP                 load lsb checksum
     F4BE   19
0985 F4BF   D2          DEC     DATAP                  dec checksum pointer
     F4C0   19
0986 F4C1   F3          TRAP    EOTCHK
0987                    *        DJNZ   EOTIMR,WCHKS2          check to test EOT
0988                    *        BTJZP  %SENSET,DRIVE,WCHKS1   check to test EOT
0989                    *        INC    EOTIMR
0990                    *        MOVP   %MTSNWE,DRIVE          turn on sensor
0991                    *        JMP    WCHKS3
0992                    *WCHKS1  BTJZP  %EOTTST,WAFER,WEOT3   test for no EOT
0993                    *        OR     %EOTFLG,RECFIL-1       EOT found
0994                    *WEOT3   MOV    %EOTCNT,EOTIMR
0995                    *WCHKS2  MOVP   %MTWE,DRIVE            turn on motor & WE
0996 F4C2   DA  WCHKS3 DJNZ    COUNT,WCHKS0           test end of chksum
     F4C3   12
     F4C4   F0
0997 F4C5   76  WCHKS4 BTJO    %>FF,BITCON,WCHKS4     wait end of byte
     F4C6   FF
     F4C7   02
     F4C8   FC
0998 F4C9   72          MOV     %>18,BITCON            restart bit-counter
     F4CA   18
     F4CB   02
0999 F4CC   C0          MOV     A,B                    move data in queue
1000                    * write out last checksum byte & extra data for RCHKSM
1001 F4CD   7D          CMP     %CWRITE,CCODE          test for XWRITE
     F4CE   04
     F4CF   3D
1002 F4D0   E6          JNE     WCHKS5                 and
     F4D1   06
1003 F4D2   A2          MOVP    %>00,BDATA             begin to return DL
     F4D3   00
     F4D4   80
1004 F4D5   A2          MOVP    %DROP,BCNTL
     F4D6   01
     F4D7   81
1005                    * wait end of extra data so RCHKSM wont hang up
```

```
1006 F4D8   76  WCHKS5 BTJO   %>FF,BITCON,WCHKS5    wait end of byte
     F4D9   FF
     F4DA   02
     F4DB   FC
```

```
1008                *-------------------------------------------------------
1009                * turn off drive & wait for stop, keep looking for EOT
1010 F4DC   72          MOV     %>FA,NIBCON             250 byte stop
     F4DD   FA
     F4DE   12
1011 F4DF   A2          MOVP    %WE,DRIVE               WE only
     F4E0   FD
     F4E1   06
1012 F4E2   76  WLAST   BTJO    %>FF,BITCON,WLAST       wait end of nibble
     F4E3   FF
     F4E4   02
     F4E5   FC
1013 F4E6   72          MOV     %>20,BITCON             restart bit-counter
     F4E7   20
     F4E8   02
1014 F4E9   F3          TRAP    EOTCHK
1015                *        DJNZ    EOTIMR,WLAST2           check to test EOT
1016                *        BTJZP   %SENSET,DRIVE,WLAST1    check to test EOT
1017                *        INC     EOTIMR
1018                *        MOVP    %SN,DRIVE               turn on sensor
1019                *        JMP     WLAST3
1020                *WLAST1  BTJOP   %EOTTST,WAFER,WEOT4    test for EOT
1021                *        MOV     %EOTCNT,EOTIMR
1022                *WLAST2  MOVP    %WE,DRIVE               WE only
1023 F4EA   DA  WLAST3 DJNZ    NIBCON,WLAST            test for end
     F4EB   12
     F4EC   F5
1024 F4ED   A2          MOVP    %STOP,DRIVE             turn drive off
     F4EE   FF
     F4EF   06
1025 F4F0   A2          MOVP    %STOP,TIMER             stop timer
     F4F1   FF
     F4F2   03
1026 F4F3   A2          MOVP    %I123C,IOCNTL           clear INT1,2&3
     F4F4   6A
     F4F5   00
1027 F4F6   74          OR      %FDIREC,FFLAG           set directory flag
     F4F7   80
     F4F8   1C
1028 F4F9   0A          RETS
1029                *-------------------------------------------------------
1030 F4FA   FB  WEOT2   TRAP    ERRWT                   error
1031 F4FB   50          BYTE    RSEOT                   wafer full
```

```
1033                *-------------------------------------------------------
1034                * write directory
1035 F4FC   F6  WDIREC TRAP   EOT                  find EOT
1036 F4FD   72         MOV    %>05,NIBCON-1        1280 sync nibbles
     F4FE   05
     F4FF   11
1037                * (640 ms delay inherent)
1038 F500   72         MOV    %>FF,NFILE           file >FF
     F501   FF
     F502   23
1039 F503   D5         CLR    NREC                 record 0
     F504   22
1040 F505   D5         CLR    NREC-1
     F506   21
1041 F507   72         MOV    %>FF,EOTIMR          directory EOT delay
     F508   FF
     F509   1B
1042 F50A   72         MOV    %>41,DLEN            directory DL
     F50B   41
     F50C   37
1043 F50D   8E         CALL   @WSYNCO              write sync pattern
     F50E F399
1044 F510   72         MOV    %ENDRCT,DATAP        directory data pointer
     F511   7F
     F512   19
1045 F513   8E  WMEM   CALL   @WBYTE               write RAM data
     F514 F3D1
1046 F516   8C         BR     @WCHKSM              write checksum
     F517 F4AD
```

```
1048                    *---------------------------------------------------------------
1049                    * read sync pattern
1050                    * blank tape noise could be high and/or low frequency,
1051                    * short and/or long bit times may be read if noise
1052                    * before sync is read as sync
1053 F519   F5  RSYNC   TRAP    TIMEX                   start motor & timer
1054 F51A   72          MOV     %BOSTIM,BITCON          macro count (150 ms)
     F51B   4B
     F51C   02
1055 F51D   A2  RSYNC0  MOVP    %>9F,TIMER              start timer
     F51E   9F
     F51F   03
1056 F520   76  RSYNC1  BTJO    %>FF,BITCON,RSYNC1      wait for safe area
     F521   FF
     F522   02
     F523   FC
1057 F524   72  RESYNC  MOV     %ISYNC2/256,INT2V-1     INT2 vector
     F525   F9
     F526   16
1058 F527   72          MOV     %ISYNC2-(256*(ISYNC2/256)),INT2V
     F528   BE
     F529   17
1059 F52A   72          MOV     %BROPC,INT2
     F52B   8C
     F52C   15
1060 F52D   A2          MOVP    %>FF,TIME               set up max time
     F52E   FF
     F52F   02
1061 F530   A2          MOVP    %I23CS,IOCNTL           clear & select INT2&3
     F531   7C
     F532   00
1062 F533   72  RSYNC2  MOV     %>20,BITCON             valid-bit counter
     F534   20
     F535   02
1063 F536   D5          CLR     CHKSUM                  clear chksum
     F537   14
1064 F538   D5          CLR     CHKSUM-1
     F539   13
1065 F53A   01  RSYNC3  IDLE                            wait transition
1066 F53B   91          MOVP    CAPTUR,B                read bit time
     F53C   03
1067 F53D   C4          INV     B                       get true count
1068 F53E   56          BTJO    %>F0,B,RSYNC2           data rate too slow
     F53F   F0
     F540   F2
1069 F541   56          BTJO    %>0C,B,RSYNC4           data rate OK
     F542   0C
     F543   02
1070 F544   E0          JMP     RSYNC2                  data rate too fast
     F545   ED
1071 F546   48  RSYNC4  ADD     B,CHKSUM                add to cumulative
     F547   01
     F548   14
1072 F549   79          ADC     %0,CHKSUM-1
     F54A   00
     F54B   13
1073 F54C   05          EINT                            after possible INT2
```

```
1074 F54D  DA          DJNZ   BITCON,RSYNC3          test end sync sample
     F54E  02
     F54F  EA
1075             * test for EOT
1076             * add check for bit time in above loop (length consistency)
1077 F550  A2          MOVP   %I3CS,IOCNTL           select & clear INT3
     F551  70
     F552  00
1078 F553  72          MOV    %4,BITCON              cumulative/ >20
     F554  04
     F555  02
1079 F556  76          BTJO   %>07,CHKSUM,RSYNC5     test rounding
     F557  07
     F558  14
     F559  02
1080 F55A  E0          JMP    RSYNC6                 no round
     F55B  03
1081 F55C  78  RSYNC5  ADD    %>08,CHKSUM            round
     F55D  08
     F55E  14
1082 F55F  DD  RSYNC6  RRC    CHKSUM-1               shift msb right
     F560  13
1083 F561  DD          RRC    CHKSUM
     F562  14
1084 F563  DA          DJNZ   BITCON,RSYNC6          test end of divide
     F564  02
     F565  F9
1085 F566  32          MOV    CHKSUM,B
     F567  14
1086 F568  92          MOVP   B,TIME                 2/3 bit timer
     F569  02
1087 F56A  D5          CLR    CHKSUM                 set chksum to 0
     F56B  14
1088 F56C  D5          CLR    CHKSUM-1
     F56D  13
1089 F56E  72          MOV    %RBITDT/256,INT2V-1    set up INT2 vector
     F56F  F9
     F570  16
1090 F571  72          MOV    %RBITDT-(256*(RBITDT/256)),INT2V
     F572  A7
     F573  17
1091 F574  72          MOV    %BROPC,INT2            set up vector branch
     F575  8C
     F576  15
1092             *       MOV    %RDBIT1,INT2           set up INT2 opcode
1093             *       MOV    %RDBIT2,INT2+1         set up INT2 parameter
1094             *       MOV    %RDBIT3,INT2+2         set up INT2 parameter
1095 F577  A2          MOVP   %I2CS,IOCNTL           select & clear INT2
     F578  4C
     F579  00
1096 F57A  72          MOV    %>02,BITCON            keep on top
     F57B  02
     F57C  02
1097 F57D  C5          CLR    B                      reset input bit
1098 F57E  F4          TRAP   BITEST                 wait end of bit
1099 F57F  56          BTJO   %SETBIT,B,RSYNC8       test for 1
     F580  08
```

```
      F581  07
1100  F582  D3  RSYNC7 INC    BITCON                set bit counter
      F583  02
1101  F584  C5         CLR    B                     reset input bit
1102  F585  F4         TRAP   BITEST                wait end of bit
1103  F586  57         BTJZ   %SETBIT,B,RSYNC9      test for 0
      F587  08
      F588  07
1104  F589  D3  RSYNC8 INC    BITCON                set bit counter
      F58A  02
1105  F58B  C5         CLR    B                     reset input bit
1106  F58C  F4         TRAP   BITEST                wait end of bit
1107  F58D  57         BTJZ   %SETBIT,B,RSYNC7      test for 0
      F58E  08
      F58F  F2
1108                   * test for 2 more 0s
1109  F590  D3  RSYNC9 INC    BITCON                set bit counter
      F591  02
1110  F592  C5         CLR    B                     reset input bit
1111  F593  F4         TRAP   BITEST                wait end of bit
1112  F594  56         BTJO   %SETBIT,B,RESYNC      test for 1
      F595  08
      F596  8D
1113  F597  D3         INC    BITCON                set bit counter
      F598  02
1114  F599  C5         CLR    B                     reset input bit
1115  F59A  F4         TRAP   BITEST                wait end of bit
1116  F59B  56         BTJO   %SETBIT,B,RESYNC      test for 1
      F59C  08
      F59D  86
1117  F59E  72         MOV    %>04,BITCON           restart nibble bit-count
      F59F  04
      F5A0  02
1118  F5A1  F4         TRAP   BITEST                wait end of nibble (spac
1119  F5A2  72         MOV    %>08,BITCON           restart nibble bit-count
      F5A3  08
      F5A4  02
1120  F5A5  C5         CLR    B                     clear all input bits
1121                   * at this point the first byte for RCOMPM is being picked up
1122  F5A6  72         MOV    %>03,COUNT             length of file/record
      F5A7  03
      F5A8  12
1123  F5A9  72         MOV    %NFILE,DATAP          file & record numbers
      F5AA  23
      F5AB  19
1124  F5AC  0A         RETS
```

```
1126                *-------------------------------------------------------
1127                * read wafer and compare to memory
1128 F5AD   73  RCOMPM AND   %RSTERR,FFLAG          reset error flag
     F5AE   FE
     F5AF   1C
1129 F5B0   D5         CLR   DATAP-1
     F5B1   18
1130 F5B2   76  RCMPM2 BTJO  %>FF,BITCON,RCMPM2     wait end of byte
     F5B3   FF
     F5B4   02
     F5B5   FC
1131 F5B6   72         MOV   %>08,BITCON            restart bit counter
     F5B7   08
     F5B8   02
1132 F5B9   62         MOV   B,A                    move data in queue
1133 F5BA   C5         CLR   B                      clear input bits
1134 F5BB   B7         SWAP  A                      position nibbles
1135 F5BC   9D         CMPA  *DATAP                 compare data
     F5BD   19
1136 F5BE   E2         JEQ   RCMPM3                 test for match
     F5BF   03
1137 F5C0   74         OR    %FERROR,FFLAG          set error flag
     F5C1   01
     F5C2   1C
1138 F5C3   D2  RCMPM3 DEC   DATAP                  dec data pointer
     F5C4   19
1139 F5C5   48         ADD   A,CHKSUM               add data to checksum
     F5C6   00
     F5C7   14
1140 F5C8   79         ADC   %0,CHKSUM-1
     F5C9   00
     F5CA   13
1141 F5CB   DA         DJNZ  COUNT,RCMPM2           dec byte counter
     F5CC   12
     F5CD   E4
1142 F5CE   76  RCMPM4 BTJO  %>FF,BITCON,RCMPM4     wait end of byte
     F5CF   FF
     F5D0   02
     F5D1   FC
1143 F5D2   72         MOV   %>04,BITCON            restart bit counter
     F5D3   04
     F5D4   02
1144 F5D5   C5         CLR   B                      clear input bits
1145                * 1st nibble of RCOMPM/RBYTE/RDL/RDLX/RCHKSM is being picked
1146 F5D6   42         MOV   DLEN,NIBCON            move DL to nibcon
     F5D7   37
     F5D8   12
1147 F5D9   42         MOV   DLEN-1,NIBCON-1
     F5DA   36
     F5DB   11
1148 F5DC   0A         RETS
```

```
1150                *-------------------------------------------------------
1151                * read DL & compare to memory
1152 F5DD   D5   RDL    CLR    NIBCON                  set nibble count to 0
     F5DE   12
1153 F5DF   D5         CLR    NIBCON-1
     F5E0   11
1154 F5E1   72         MOV    %8,DLPCNT               set up dl pointer/counte
     F5E2   08
     F5E3   19
1155 F5E4   76   RDL1   BTJO   %>FF,BITCON,RDL1       wait end of nibble
     F5E5   FF
     F5E6   02
     F5E7   FC
1156 F5E8   72         MOV    %4,BITCON               restart nibble bit-count
     F5E9   04
     F5EA   02
1157 F5EB   62         MOV    B,A                     move data through queue
1158 F5EC   C5         CLR    B                       set all input bits to 0
1159 F5ED   77         BTJZ   %1,DLPCNT,RDL2          test for ms/ls nibble
     F5EE   01
     F5EF   19
     F5F0   01
1160 F5F1   B7         SWAP   A                       ms nibble
1161 F5F2   76   RDL2   BTJO   %>C,DLPCNT,RDL3        test for next byte of dl
     F5F3   0C
     F5F4   19
     F5F5   05
1162 F5F6   44         OR     A,NIBCON-1              store ms byte
     F5F7   00
     F5F8   11
1163 F5F9   E0         JMP    RDL4
     F5FA   06
1164 F5FB   44   RDL3   OR     A,NIBCON                store ls byte
     F5FC   00
     F5FD   12
1165 F5FE   7A         SUB    %2,DLPCNT               non-linear adjust
     F5FF   02
     F600   19
1166 F601   DA   RDL4   DJNZ   DLPCNT,RDL1            test for end of dl
     F602   19
     F603   E0
1167                * this DL is compared to DL in SAB in main loop
1168 F604   E0         JMP    RDLX5
     F605   37
```

```
1170                *-------------------------------------------------------
1171                * read DL & transmit bus
1172 F606  D5  RDLX   CLR    NIBCON                 set nibble count to 0
     F607  12
1173 F608  D5         CLR    NIBCON-1
     F609  11
1174 F60A  72         MOV    %8,DLPCNT              set up dl pointer/counte
     F60B  08
     F60C  19
1175                * HSK released in XREAD
1176 F60D  76  RDLX1  BTJO   %>FF,BITCON,RDLX1      wait end of nibble
     F60E  FF
     F60F  02
     F610  FC
1177 F611  72         MOV    %4,BITCON              restart nibble bit-count
     F612  04
     F613  02
1178 F614  62         MOV    B,A                    move data through queue
1179 F615  C5         CLR    B                      set all input bits to 0
1180 F616  A7         BTJZP  %BAV,BSTAT,RHELP       test for BAV active
     F617  02
     F618  81
     F619  3C
1181 F61A  A6         BTJOP  %HSK,BSTAT,RHELP       test for bus ready
     F61B  01
     F61C  81
     F61D  38
1182 F61E  82         MOVP   A,BDATA                send data over bus
     F61F  80
1183 F620  A2         MOVP   %DROP,BCNTL            drop HSK
     F621  01
     F622  81
1184 F623  A2         MOVP   %HSKSET,BCNTL          let HSK float
     F624  00
     F625  81
1185 F626  77         BTJZ   %1,DLPCNT,RDLX2        test for ms/ls nibble
     F627  01
     F628  19
     F629  01
1186 F62A  B7         SWAP   A                      ms nibble
1187 F62B  76  RDLX2  BTJO   %>C,DLPCNT,RDLX3       test for next byte of dl
     F62C  0C
     F62D  19
     F62E  05
1188 F62F  44         OR     A,NIBCON-1             store ms byte
     F630  00
     F631  11
1189 F632  E0         JMP    RDLX4
     F633  06
1190 F634  44  RDLX3  OR     A,NIBCON               store ls byte
     F635  00
     F636  12
1191 F637  7A         SUB    %2,DLPCNT              non-linear adjust
     F638  02
     F639  19
1192 F63A  DA  RDLX4  DJNZ   DLPCNT,RDLX1           test for end of dl
     F63B  19
```

```
      F63C   D0
1193  F63D   76  RDLX5  BTJO   %>FF,BITCON,RDLX5     test for end of nibble
      F63E   FF
      F63F   02
      F640   FC
1194  F641   72         MOV    %>08,BITCON           restart bit counter
      F642   08
      F643   02
1195  F644   C5         CLR    B                     reset all input bits
1196                  * non-data spacer followed by 1st nibble of data is picked u
1197  F645   48         ADD    NIBCON,CHKSUM         add dl to checksum
      F646   12
      F647   14
1198  F648   79         ADC    %0,CHKSUM-1
      F649   00
      F64A   13
1199  F64B   48         ADD    NIBCON-1,CHKSUM
      F64C   11
      F64D   14
1200  F64E   79         ADC    %0,CHKSUM-1
      F64F   00
      F650   13
1201  F651   DF         RLC    NIBCON                multiply nibcon by 2
      F652   12
1202  F653   DF         RLC    NIBCON-1              (byte count -> nibble co
      F654   11
1203  F655   0A         RETS
```

```
1205                *-------------------------------------------------------------
1206 F656   FB  RHELP  TRAP   ERRWT                  message terminated/slow
1207 F657   FF         BYTE   RSTIME                 bus error
```

```
1209                *-------------------------------------------------------------
1210                * read wafer & transmit bus
1211 F658   76  RNIB2  BTJO   %>FF,BITCON,RNIB2     wait end of nibble
     F659   FF
     F65A   02
     F65B   FC
1212 F65C   72         MOV    %>04,BITCON          restart bit counter
     F65D   04
     F65E   02
1213 F65F   62         MOV    B,A                  move data in queue
1214 F660   C5         CLR    B                    clear input bits
1215 F661   A7         BTJZP  %BAV,BSTAT,RHELP     test for BAV active
     F662   02
     F663   81
     F664   F1
1216 F665   A6         BTJOP  %HSK,BSTAT,RHELP     test for bus ready
     F666   01
     F667   81
     F668   ED
1217 F669   82         MOVP   A,BDATA              send data over bus
     F66A   80
1218 F66B   A2         MOVP   %DROP,BCNTL          drop HSK
     F66C   01
     F66D   81
1219 F66E   A2         MOVP   %HSKSET,BCNTL        let HSK float
     F66F   00
     F670   81
1220 F671   23         AND    %LSN,A               clear msn for checksum
     F672   0F
1221 F673   48         ADD    A,CHKSUM             add data to checksum
     F674   00
     F675   14
1222 F676   79         ADC    %0,CHKSUM-1
     F677   00
     F678   13
1223 F679   D2  RNIB   DEC    NIBCON               dec nibble counter
     F67A   12
1224 F67B   7B         SBB    %0,NIBCON-1
     F67C   00
     F67D   11
1225 F67E   E3         JC     RNIB2                test end of data
     F67F   D8
1226 F680   E0         JMP    RBYTE3
     F681   33
```

```
1228                *-------------------------------------------------------
1229                * read wafer & forget
1230 F682   76   FORGT2 BTJO   %>FF,BITCON,FORGT2    wait end of nibble
     F683   FF
     F684   02
     F685   FC
1231 F686   72         MOV    %>04,BITCON           restart bit counter
     F687   04
     F688   02
1232 F689   62         MOV    B,A                   move data in queue
1233 F68A   C5         CLR    B                     clear input bits
1234 F68B   23         AND    %LSN,A                clear msn for checksum
     F68C   0F
1235 F68D   48         ADD    A,CHKSUM              add data to checksum
     F68E   00
     F68F   14
1236 F690   79         ADC    %0,CHKSUM-1
     F691   00
     F692   13
1237 F693   D2   FORGET DEC    NIBCON                dec nibble counter
     F694   12
1238 F695   7B         SBB    %0,NIBCON-1
     F696   00
     F697   11
1239 F698   E3         JC     FORGT2                test end of data
     F699   E8
1240 F69A   E0         JMP    RBYTE3
     F69B   19
```

```
1242                *-------------------------------------------------------
1243                * read wafer and store
1244 F69C   D5   RBYTE  CLR    DATAP-1                 keep in reg file
     F69D   18
1245 F69E   76   RBYTE2 BTJO   %>FF,BITCON,RBYTE2     wait end of byte
     F69F   FF
     F6A0   02
     F6A1   FC
1246 F6A2   72          MOV    %>08,BITCON             restart bit counter
     F6A3   08
     F6A4   02
1247 F6A5   62          MOV    B,A                     move data in queue
1248 F6A6   C5          CLR    B                       clear input bits
1249 F6A7   B7          SWAP   A                       position nibbles
1250 F6A8   9B          STA    *DATAP                  store data
     F6A9   19
1251 F6AA   D2          DEC    DATAP                   dec data pointer
     F6AB   19
1252 F6AC   48          ADD    A,CHKSUM                add data to checksum
     F6AD   00
     F6AE   14
1253 F6AF   79          ADC    %0,CHKSUM-1
     F6B0   00
     F6B1   13
1254 F6B2   DA          DJNZ   NIBCON,RBYTE2           dec byte counter
     F6B3   12
     F6B4   E9
1255 F6B5   76   RBYTE3 BTJO   %>FF,BITCON,RBYTE3     wait end of nibble
     F6B6   FF
     F6B7   02
     F6B8   FC
1256 F6B9   72          MOV    %>04,BITCON             restart bit counter
     F6BA   04
     F6BB   02
1257 F6BC   C5          CLR    B                       clear input bits
1258                * at this point the 1st nibble of RCHKSM is being picked up
1259 F6BD   8C          BR     @RCHKSM
     F6BE F6F5
```

```
1261                *-------------------------------------------------------
1262                * read wafer and compare to bus data
1263 F6C0   76  RCMPB2 BTJO   %>FF,BITCON,RCMPB2    wait end of nibble
     F6C1   FF
     F6C2   02
     F6C3   FC
1264 F6C4   72         MOV    %>04,BITCON          restart bit counter
     F6C5   04
     F6C6   02
1265 F6C7   D1         MOV    B,DATA               save data for compare
     F6C8   1A
1266 F6C9   C5         CLR    B                    clear input bits
1267 F6CA   A7         BTJZP  %BAV,BSTAT,RHELP     test for BAV active
     F6CB   02
     F6CC   81
     F6CD   88
1268 F6CE   A7         BTJZP  %IRQ,BSTAT,RHELP     test for bus data ready
     F6CF   08
     F6D0   81
     F6D1   84
1269 F6D2   A2         MOVP   %RELEAS,BCNTL        reset IRQ
     F6D3   01
     F6D4   81
1270 F6D5   80         MOVP   BDATA,A              read data from bus
     F6D6   80
1271 F6D7   A2         MOVP   %HSKSET,BCNTL        let hsk float
     F6D8   00
     F6D9   81
1272 F6DA   23         AND    %LSN,A               clear msn of bus data
     F6DB   0F
1273 F6DC   73         AND    %LSN,DATA            clear msn of wafer data
     F6DD   0F
     F6DE   1A
1274 F6DF   1D         CMP    DATA,A               compare data
     F6E0   1A
1275 F6E1   E2         JEQ    RCMPB3               test for match
     F6E2   03
1276 F6E3   74         OR     %FERROR,FFLAG        store compare-error
     F6E4   01
     F6E5   1C
1277 F6E6   48  RCMPB3 ADD    A,CHKSUM             add data to checksum
     F6E7   00
     F6E8   14
1278 F6E9   79         ADC    %0,CHKSUM-1
     F6EA   00
     F6EB   13
1279 F6EC   D2  RCOMPB DEC    NIBCON               dec nibble counter
     F6ED   12
1280 F6EE   7B         SBB    %0,NIBCON-1
     F6EF   00
     F6F0   11
1281 F6F1   E3         JC     RCMPB2               test end of data
     F6F2   CD
1282 F6F3   E0         JMP    RBYTE3
     F6F4   C0
```

```
1284                *-------------------------------------------------------
1285                * read chekcsum
1286 F6F5  76   RCHKSM BTJO   %>FF,BITCON,RCHKSM    wait end of nibble
     F6F6  FF
     F6F7  02
     F6F8  FC
1287 F6F9  72         MOV    %4,BITCON             restart bit count
     F6FA  04
     F6FB  02
1288                * 2nd half of first byte of checksum is being picked up
1289 F6FC  72         MOV    %DLEN,DATAP           set up pointer
     F6FD  37
     F6FE  19
1290 F6FF  D5         CLR    DATAP-1
     F700  18
1291 F701  72         MOV    %2,COUNT              set up length
     F702  02
     F703  12
1292 F704  76   RCHKS1 BTJO   %>FF,BITCON,RCHKS1    wait end of byte
     F705  FF
     F706  02
     F707  FC
1293 F708  72         MOV    %8,BITCON             restart bit count
     F709  08
     F70A  02
1294 F70B  62         MOV    B,A                   move data through queue
1295 F70C  C5         CLR    B                     mask off msn
1296 F70D  B7         SWAP   A                     position nibbles
1297 F70E  9B         STA    *DATAP                store checksum
     F70F  19
1298 F710  D2         DEC    DATAP                 dec pointer
     F711  19
1299 F712  DA         DJNZ   COUNT,RCHKS1          test for end of chksum
     F713  12
     F714  EF
1300                * turn off drive
1301 F715  F5         TRAP   TIMEX                 set up interrupt (time-o
1302 F716  A2         MOVP   %STOP,DRIVE           drive off
     F717  FF
     F718  06
1303                * verify checksum
1304 F719  32         MOV    CCODE,B               move ccode to B
     F71A  3D
1305 F71B  4D         CMP    DLEN,CHKSUM           compare chksum
     F71C  37
     F71D  14
1306 F71E  E6         JNE    RCHKS3                test chksum
     F71F  23
1307 F720  4D         CMP    DLEN-1,CHKSUM-1       compare chksum
     F721  36
     F722  13
1308 F723  E6         JNE    RCHKS3                test chksum
     F724  1E
1309 F725  74         OR     %FDIREC,FFLAG         set directory flag
     F726  80
     F727  1C
1310 F728  5D         CMP    %CREAD,B              test for read code
```

```
      F729  03
1311  F72A  E2          JEQ     RCHKS2
      F72B  10
1312  F72C  5D          CMP     %CVERIF,B               test for verify code
      F72D  0C
1313  F72E  E6          JNE     RLASTO
      F72F  24
1314  F730  77          BTJZ    %FERROR,FFLAG,RCHKS2    test for no data error
      F731  01
      F732  1C
      F733  08
1315  F734  A2          MOVP    %RSVERI,BDATA           begin to return status
      F735  18
      F736  80
1316  F737  72          MOV     %RSVERI,STATUS          set up end of status
      F738  18
      F739  3E
1317  F73A  E0          JMP     RCHKS5
      F73B  15
1318  F73C  A2  RCHKS2  MOVP    %RSOK,BDATA             begin to return status
      F73D  00
      F73E  80
1319  F73F  D5          CLR     STATUS                  set up end of status
      F740  3E
1320  F741  E0          JMP     RCHKS5
      F742  0E
1321  F743  5D  RCHKS3  CMP     %CREAD,B                test for read code
      F744  03
1322  F745  E2          JEQ     RCHKS4
      F746  04
1323  F747  5D          CMP     %CVERIF,B               test for verify code
      F748  0C
1324  F749  E6          JNE     RCHKSE
      F74A  1D
1325  F74B  A2  RCHKS4  MOVP    %RSDATA,BDATA           begin to return status
      F74C  19
      F74D  80
1326  F74E  72          MOV     %RSDATA,STATUS          set up end of status
      F74F  19
      F750  3E
1327  F751  A2  RCHKS5  MOVP    %DROP,BCNTL
      F752  01
      F753  81
1328              *-------------------------------------------------------
1329              * turn off drive & wait for stop, verify checksum
1330  F754  72  RLASTO  MOV     %>FA,NIBCON             250 byte stop
      F755  FA
      F756  12
1331  F757  76  RLAST   BTJO    %>FF,BITCON,RLAST       test for end
      F758  FF
      F759  02
      F75A  FC
1332  F75B  72          MOV     %>20,BITCON             restart bit counter
      F75C  20
      F75D  02
1333  F75E  DA          DJNZ    NIBCON,RLAST            dec byte counter
      F75F  12
```

```
      F760   F6
1334  F761   A2          MOVP   %STOP,TIMER          stop timer
      F762   FF
      F763   03
1335  F764   A2          MOVP   %I123C,IOCNTL        clear INT1,2&3
      F765   6A
      F766   00
1336  F767   0A          RETS
1337  F768   8E   RCHKSE CALL   @RLASTO              delay
      F769 F754
1338  F76B   FB   RCHKE2 TRAP   ERRWT                checksum error
1339  F76C   06          BYTE   RSDEVI               device error code
```

```
1341                *-------------------------------------------------------
1342                * read directory
1343 F76D   F6   RDIREC TRAP   EOT                       find EOT
1344 F76E   A2         MOVP   %>9F,TIMER                start timer
     F76F   9F
     F770   03
1345 F771   72         MOV    %BOTIME,BITCON            macro count
     F772   96
     F773   02
1346 F774   72         MOV    %>FF,NFILE                file >FF
     F775   FF
     F776   23
1347 F777   D5         CLR    NREC                      record 0
     F778   22
1348 F779   D5         CLR    NREC-1
     F77A   21
1349 F77B   72         MOV    %>41,DLEN                 directory DL
     F77C   41
     F77D   37
1350 F77E   8E         CALL   @RSYNCO                   read sync & file/record
     F77F F51D
1351 F781   8E         CALL   @RCOMPM                   compare file/record
     F782 F5AD
1352 F784   76         BTJO   %FERROR,FFLAG,RCHKE2      test for wrong file
     F785   01
     F786   1C
     F787   E3
1353 F788   72         MOV    %ENDRCT,DATAP             set up data pointer
     F789   7F
     F78A   19
1354 F78B   76   RMEM  BTJO   %>FF,BITCON,RMEM          wait end of nibble
     F78C   FF
     F78D   02
     F78E   FC
1355 F78F   72         MOV    %4,BITCON                 restart bit count
     F790   04
     F791   02
1356                * 2nd half of 1st nibble of RBYTE is being picked up
1357 F792   8C         BR     @RBYTE                    read directory data
     F793 F69C
```

```
1359                *-------------------------------------------------------
1360                * find EOT
1361 F795   F5   FNDEOT TRAP   TIMEX                    start timer operation
1362 F796   72          MOV    %>01,EOTIMR          set up duty cycle
     F797   01
     F798   1B
1363 F799   73          AND    %FMSSNG,FFLAG          reset found flag
     F79A   F7
     F79B   1C
1364 F79C   76   FEOT1  BTJO   %>FF,BITCON,FEOT1      wait end of nibble
     F79D   FF
     F79E   02
     F79F   FC
1365 F7A0   72          MOV    %>08,BITCON            restart bit-count
     F7A1   08
     F7A2   02
1366 F7A3   F3          TRAP   EOTCHK                 test for EOT
1367 F7A4   77          BTJZ   %FFOUND,FFLAG,FEOT1    test for not EOT
     F7A5   08
     F7A6   1C
     F7A7   F4
1368 F7A8   0A          RETS                          EOT found
1369                *-------------------------------------------------------
1370                * start motor & timer operation
1371 F7A9   06   ROLLEM DINT                          disable interrupts
1372                *       MOV    %WRBIT1,INT2         set up INT2 opcode
1373                *       MOV    %WRBIT2,INT2+1       set up INT2 parameter
1374 F7AA   72          MOV    %WBITDT/256,INT2V-1    set up INT2 vector
     F7AB   F9
     F7AC   16
1375 F7AD   72          MOV    %WBITDT-(256*(WBITDT/256)),INT2V
     F7AE   99
     F7AF   17
1376 F7B0   72          MOV    %BROPC,INT2            set up vector branch
     F7B1   8C
     F7B2   15
1377 F7B3   05          EINT                          re-enable ints
1378 F7B4   A2          MOVP   %BITIME,TIME           half-bit time
     F7B5   0A
     F7B6   02
1379 F7B7   72          MOV    %>08,BITCON            start bit counter
     F7B8   08
     F7B9   02
1380 F7BA   A2          MOVP   %I2CS,IOCNTL           select & clear INT2
     F7BB   4C
     F7BC   00
1381 F7BD   A2          MOVP   %BEGINW,TIMER          start timer
     F7BE   80
     F7BF   03
1382 F7C0   A2          MOVP   %MT,DRIVE              turn on motor
     F7C1   F7
     F7C2   06
1383 F7C3   0A          RETS
```

```
1385                *-------------------------------------------------------
1386                * search for record in current file
1387 F7C4   4D  SEARCH CMP   NREC-1,RNUM-1          compare record numbers
     F7C5   21
     F7C6   3A
1388 F7C7   E2        JEQ   SEARC5                  test low order record #
     F7C8   28
1389 F7C9   E7        JL    SEARC2                  record follows
     F7CA   03
1390 F7CB   8E  SEARC1 CALL  @PREPOS                 find start of file
     F7CC F7FA
1391 F7CE   8E  SEARC2 CALL  @TSTEOF                 test for EOF
     F7CF F8B2
1392 F7D1   72        MOV   %RSEOF,STATUS           EOF error status
     F7D2   07
     F7D3   3E
1393 F7D4   E2        JEQ   SEARC4
     F7D5   1A
1394 F7D6   8E        CALL  @TSTREC                 test for record
     F7D7 F8A8
1395 F7D9   E2        JEQ   SEARC3
     F7DA   13
1396 F7DB   D3        INC   NREC                    increment record number
     F7DC   22
1397 F7DD   79        ADC   %0,NREC-1
     F7DE   00
     F7DF   21
1398 F7E0   8E        CALL  @RSYNC                  read sync & get bit time
     F7E1 F519
1399 F7E3   8E        CALL  @RCOMPM                 compare file/record
     F7E4 F5AD
1400 F7E6   8E        CALL  @RDL                    read DL
     F7E7 F5DD
1401 F7E9   8E        CALL  @FORGET                 read wafer & forget
     F7EA F693
1402 F7EC   E0        JMP   SEARC2
     F7ED   E0
1403 F7EE   D5  SEARC3 CLR   STATUS                 store OK status
     F7EF   3E
1404 F7F0   0A  SEARC4 RETS
1405 F7F1   4D  SEARC5 CMP   NREC,RNUM              compare record numbers
     F7F2   22
     F7F3   3B
1406 F7F4   E2        JEQ   SEARC3                  record ready
     F7F5   F8
1407 F7F6   E7        JL    SEARC2                  record follows
     F7F7   D6
1408 F7F8   E0        JMP   SEARC1                  record precedes
     F7F9   D1
```

```
1410                    *-------------------------------------------------------
1411 F7FA   32  PREPOS  MOV     NFILE,B                 set up index
     F7FB   23
1412 F7FC   8E          CALL    @STFILD                 save file parameters
     F7FD F8E9
1413 F7FF   8E          CALL    @WDIREC                 write directory
     F800 F4FC
1414 F802   74          OR      %FNAME,FFLAG            file name in RAM
     F803   20
     F804   1C
1415                    * unless find by number is supported
1416                    *-------------------------------------------------------
1417                    * position wafer to file (assume valid file number present)
1418 F805   74  POSITN  OR      %FFOUND,FFLAG           set found flag
     F806   08
     F807   1C
1419 F808   32          MOV     NFILE,B                 set up index
     F809   23
1420 F80A   D1          MOV     B,TEMP2                 save file #
     F80B   3C
1421 F80C   77          BTJZ    %FNAME,FFLAG,BYNMBR     test file name/number
     F80D   20
     F80E   1C
     F80F   3A
1422 F810   57  BYNAME  BTJZ    %>FF,B,POSIT5           test for direct not last
     F811   FF
     F812   2B
1423 F813   77          BTJZ    %LAST,FILEO,POSIT3      test for 0 not last file
     F814   40
     F815   40
     F816   0E
1424 F817   76          BTJO    %ACTIVE,FILEO,POSIT3    test for 0 active
     F818   80
     F819   40
     F81A   0A
1425 F81B   D3          INC     NFILE                   no files present
     F81C   23
1426 F81D   73  POSIT1  AND     %FMSSNG,FFLAG           reset file-not-found
     F81E   F7
     F81F   1C
1427 F820   E0          JMP     FEOD3                   file not found, find EOD
     F821   46
1428 F822   8E  POSIT2  CALL    @RDIREC                 read directory
     F823 F76D
1429 F825   8E  POSIT3  CALL    @CMPFIL                 compare file header
     F826 F884
1430 F828   76          BTJO    %FERROR,FFLAG,POSIT4    test for file not found
     F829   01
     F82A   1C
     F82B   05
1431 F82C   77          BTJZ    %ACTIVE,RECFIL-1,POSIT4 test for not active
     F82D   80
     F82E   1D
     F82F   01
1432 F830   0A          RETS
1433 F831   32  POSIT4  MOV     NFILE,B                 move file number to B
     F832   23
```

```
1434 F833  4D          CMP   B,TEMP2                compare present/starting
     F834  01
     F835  3C
1435 F836  E2          JEQ   POSIT1                 test for complete search
     F837  E5
1436 F838  77          BTJZ  %>F0,TEMP2,POSIT5      if any are 0, all are 0
     F839  F0
     F83A  3C
     F83B  02
1437 F83C  D5          CLR   TEMP2                  clear TEMP2 if it was >F
     F83D  3C
1438 F83E  8E  POSIT5  CALL  @LDFILD                load file parameters
     F83F  F900
1439 F841  8E          CALL  @FNDEOF                find EOF
     F842  F84C
1440 F844  76          BTJO  %LAST,RECFIL-1,POSIT2  test for last file
     F845  40
     F846  1D
     F847  DA
1441 F848  E0          JMP   POSIT3
     F849  DB
1442 F84A  E0  BYNMBR  JMP   BYNMBR                 not implemented
     F84B  FE
```

```
1444                *-------------------------------------------------------------
1445                * find EOF
1446 F84C   8E   FNDEOF CALL   @TSTEOF              compare current/last rec
     F84D F8B2
1447 F84F   E6          JNE    FEOF2                test for not last record
     F850   01
1448 F851   0A          RETS                        EOF found
1449 F852   D3   FEOF2  INC    NREC                 increment record number
     F853   22
1450 F854   79          ADC    %0,NREC-1
     F855   00
     F856   21
1451 F857   8E          CALL   @RSYNC               read sync & get bit time
     F858 F519
1452 F85A   8E          CALL   @RCOMPM              compare file/record
     F85B F5AD
1453 F85D   8E          CALL   @RDL                 read DL
     F85E F5DD
1454 F860   8E          CALL   @FORGET              read wafer & forget
     F861 F693
1455 F863   E0          JMP    FNDEOF
     F864   E7
```

```
1457                *-------------------------------------------------------------
1458                * find EOF of last file (assume valid file number present)
1459 F865   74  FNDEOD OR      %FFOUND,FFLAG          set found flag
     F866   08
     F867   1C
1460 F868   32  FEOD3  MOV     NFILE,B                set up index
     F869   23
1461 F86A   8E          CALL    @LDFILD                load file parameters
     F86B F900
1462 F86D   77          BTJZ    %LAST,RECFIL-1,FEOD2   test for not last file
     F86E   40
     F86F   1D
     F870   0B
1463 F871   76          BTJO    %ACTIVE,RECFIL-1,FEOD2 test for active
     F872   80
     F873   1D
     F874   07
1464                * if last file is inactive it must be file 0
1465 F875   73          AND     %FMSSNG,FFLAG          reset file not found
     F876   F7
     F877   1C
1466 F878   0A          RETS
1467 F879   8E  FEOD1   CALL    @CMPFIL                read next file header
     F87A F884
1468 F87C   8E  FEOD2   CALL    @FNDEOF                find EOF
     F87D F84C
1469 F87F   77          BTJZ    %LAST,RECFIL-1,FEOD1   test for not last file
     F880   40
     F881   1D
     F882   F6
1470 F883   0A          RETS
```

```
1472                *-------------------------------------------------------
1473                * compare file header (tests file/record & filename)
1474 F884   D3  CMPFIL INC     NFILE                    next file
     F885   23
1475 F886   32  CMPFLO MOV     NFILE,B                  load file parameters
     F887   23
1476 F888   8E          CALL    @LDFILD                  load file parameters
     F889 F900
1477 F88B   72  CMPFL1 MOV     %>0F,DLEN                filename DL
     F88C   0F
     F88D   37
1478 F88E   D5          CLR     NREC                     record 0
     F88F   22
1479 F890   D5          CLR     NREC-1
     F891   21
1480 F892   8E          CALL    @RSYNC                   read sync
     F893 F519
1481 F895   8E          CALL    @RCOMPM                  read file/record
     F896 F5AD
1482 F898   72          MOV     %FNAME1,DATAP            set up data pointer
     F899   32
     F89A   19
1483 F89B   76  CMPFL2 BTJO    %>FF,BITCON,CMPFL2       restart bit count
     F89C   FF
     F89D   02
     F89E   FC
1484 F89F   72          MOV     %4,BITCON
     F8A0   04
     F8A1   02
1485                * 2nd half of 1st filename byte is being picked up
1486 F8A2   8E          CALL    @RCOMPM                  compare filename
     F8A3 F5AD
1487 F8A5   8C          BR      @RCHKSM                  read checksum
     F8A6 F6F5
```

```
1489                *-------------------------------------------------------------
1490 F8A8   4D   TSTREC CMP     NREC,RNUM              compare current/target r
     F8A9   22
     F8AA   3B
1491 F8AB   E6           JNE     TREC2
     F8AC   04
1492 F8AD   32           MOV     RNUM-1,B
     F8AE   3A
1493 F8AF   3D           CMP     NREC-1,B
     F8B0   21
1494 F8B1   0A   TREC2   RETS
```

```
1496                *----------------------------------------------------------
1497 F8B2  4D  TSTEOF CMP     NREC,RECFIL           compare current/last rec
     F8B3  22
     F8B4  1E
1498 F8B5  E6         JNE     TEOF2
     F8B6  06
1499 F8B7  32         MOV     RECFIL-1,B
     F8B8  1D
1500 F8B9  53         AND     %>0F,B
     F8BA  0F
1501 F8BB  3D         CMP     NREC-1,B
     F8BC  21
1502 F8BD  0A  TEOF2  RETS
```

```
1504                    *---------------------------------------------------------------
1505 F8BE   76  TSTBIT BTJO    %>FF,BITCON,TSTBIT      wait end of nibble/byte
     F8BF   FF
     F8C0   02
     F8C1   FC
1506 F8C2   0A          RETS
1507                    *---------------------------------------------------------------
1508 F8C3   D0  TSTEOT MOV     A,DATA                  save A
     F8C4   1A
1509 F8C5   DA          DJNZ    EOTIMR,TSTE03           check to test EOT
     F8C6   1B
     F8C7   18
1510 F8C8   A7          BTJZP   %SENSET,DRIVE,TSTE01    check to test EOT
     F8C9   04
     F8CA   06
     F8CB   0A
1511 F8CC   D3          INC     EOTIMR
     F8CD   1B
1512 F8CE   80          MOVP    DRIVE,A                 turn on sensor
     F8CF   06
1513 F8D0   23          AND     %SN,A
     F8D1   FB
1514 F8D2   82          MOVP    A,DRIVE
     F8D3   06
1515 F8D4   E0          JMP     TSTE04
     F8D5   10
1516 F8D6   A7  TSTE01 BTJZP   %EOTTST,WAFER,TSTE02   test for not EOT
     F8D7   02
     F8D8   0A
     F8D9   03
1517 F8DA   74          OR      %FFOUND,FFLAG           EOT has been found
     F8DB   08
     F8DC   1C
1518 F8DD   72  TSTE02 MOV     %EOTCN2,EOTIMR          set up duty cycle
     F8DE   23
     F8DF   1B
1519 F8E0   80  TSTE03 MOVP    DRIVE,A                 turn off sensor
     F8E1   06
1520 F8E2   24          OR      %NONSN,A
     F8E3   04
1521 F8E4   82          MOVP    A,DRIVE
     F8E5   06
1522 F8E6   12  TSTE04 MOV     DATA,A                  restore A
     F8E7   1A
1523 F8E8   0A          RETS                            EOT found
```

```
1525                    *---------------------------------------------------------------
1526                    * store file parameters
1527 F8E9   CE  STFILD RL      B                       adjust index
1528 F8EA   CE         RL      B
1529 F8EB   12  STFIL2 MOV     MAXLEN,A
     F8EC   20
1530 F8ED   AB         STA     @FILE0+3(B)             store ls byte of MRL
     F8EE 0043
1531 F8F0   12         MOV     MAXLEN-1,A
     F8F1   1F
1532 F8F2   AB         STA     @FILE0+2(B)             store ms byte of MRL
     F8F3 0042
1533 F8F5   12         MOV     RECFIL,A
     F8F6   1E
1534 F8F7   AB         STA     @FILE0+1(B)             store ls byte of # recor
     F8F8 0041
1535 F8FA   12         MOV     RECFIL-1,A
     F8FB   1D
1536 F8FC   AB         STA     @FILE0(B)               store ms byte of # recor
     F8FD 0040
1537 F8FF   0A         RETS
1538                    *---------------------------------------------------------------
1539                    * load file parameters
1540 F900   CE  LDFILD RL      B                       adjust index
1541 F901   CE         RL      B
1542 F902   AA  LDFIL2 LDA     @FILE0+3(B)             load ls byte of MRL
     F903 0043
1543 F905   D0         MOV     A,MAXLEN
     F906   20
1544 F907   AA         LDA     @FILE0+2(B)             load ms byte of MRL
     F908 0042
1545 F90A   D0         MOV     A,MAXLEN-1
     F90B   1F
1546 F90C   AA         LDA     @FILE0+1(B)             load ls byte of # record
     F90D 0041
1547 F90F   D0         MOV     A,RECFIL
     F910   1E
1548 F911   AA         LDA     @FILE0(B)               load ms byte of # record
     F912 0040
1549 F914   D0         MOV     A,RECFIL-1
     F915   1D
1550 F916   0A         RETS
```

```
1552                    *---------------------------------------------------------
1553                    * receive and store data from bus
1554 F917   A2  RNXHSK MOVP   %HSKSET,BCNTL          release HSK
     F918   00
     F919   81
1555 F91A   8E  RCVPAB CALL   @RCVNIB                receive lsd
     F91B F94F
1556 F91D   A2         MOVP   %HSKSET,BCNTL          let HSK float
     F91E   00
     F91F   81
1557 F920   62         MOV    B,A                    save lsd
1558 F921   8E         CALL   @RCVNIB                receive msd, hold HSK
     F922 F94F
1559 F924   C7         SWAP   B                      justify byte
1560 F925   64         OR     B,A                    combine nibbles
1561 F926   D5         CLR    DATAP-1                keep pointer in reg file
     F927   18
1562 F928   9B         STA    *DATAP                 store byte
     F929   19
1563 F92A   D2         DEC    DATAP                  decrement pointer
     F92B   19
1564 F92C   D2         DEC    COUNT                  decrement counter
     F92D   12
1565 F92E   E3         JC     RNXHSK                 test for end of data
     F92F   E7
1566 F930   0A         RETS
1567                    *---------------------------------------------------------
1568                    * load and transmit data to bus
1569 F931   D2  XMTPAB DEC    COUNT                  decrement counter
     F932   12
1570 F933   E3         JC     WNXPAB                 test for non-zero DL
     F934   04
1571 F935   0A         RETS
1572 F936   A2  WNXHSK MOVP   %HSKSET,BCNTL          release HSK
     F937   00
     F938   81
1573 F939   D5  WNXPAB CLR    DATAP-1                keep pointer in reg file
     F93A   18
1574 F93B   9A         LDA    *DATAP                 load data into a
     F93C   19
1575 F93D   C0         MOV    A,B                    move nibble into b
1576 F93E   8E         CALL   @XMTNIB                transmit lsd
     F93F F961
1577 F941   A2  WNXPA2 MOVP   %HSKSET,BCNTL          let HSK float
     F942   00
     F943   81
1578 F944   C7         SWAP   B                      position data
1579 F945   8E         CALL   @XMTNIB                transmit msd, hold HSK
     F946 F961
1580 F948   D2         DEC    DATAP                  decrement pointer
     F949   19
1581 F94A   D2         DEC    COUNT                  decrement counter
     F94B   12
1582 F94C   E3         JC     WNXHSK                 test for end of data
     F94D   E8
1583 F94E   0A         RETS
```

```
1585                    *---------------------------------------------------------------
1586                    * receive nibble from bus
1587 F94F   A7  RCVNIB BTJZP  %BAV,BSTAT,GONE        test for BAV active
     F950   02
     F951   81
     F952   0C
1588 F953   A7         BTJZP  %IRQ,BSTAT,RCVNIB     test for bus data ready
     F954   08
     F955   81
     F956   F8
1589 F957   A2         MOVP   %RELEAS,BCNTL         reset IRQ
     F958   01
     F959   81
1590 F95A   91         MOVP   BDATA,B               read data from bus
     F95B   80
1591 F95C   53         AND    %LSN,B                clear msn
     F95D   0F
1592 F95E   0A         RETS
1593                    *---------------------------------------------------------------
1594                    * disappearing-message-frame handler
1595 F95F   FB  GONE   TRAP   ERRWT                 message terminated
1596 F960   FF         BYTE   RSTIME                bus error
1597                    *---------------------------------------------------------------
1598                    * transmit nibble to bus
1599 F961   A7  XMTNIB BTJZP  %BAV,BSTAT,GONE        test for BAV active
     F962   02
     F963   81
     F964   FA
1600 F965   A6         BTJOP  %HSK,BSTAT,XMTNIB     test for bus ready
     F966   01
     F967   81
     F968   F8
1601 F969   92         MOVP   B,BDATA               send data over bus
     F96A   80
1602 F96B   A2         MOVP   %DROP,BCNTL           drop HSK
     F96C   01
     F96D   81
1603 F96E   0A         RETS
```

```
1605                *-------------------------------------------------------
1606                * receive and discard rest of message
1607 F96F   06   RCVDMO DINT                            no interrupt interferenc
1608 F970   A2          MOVP   %RELEAS,BCNTL            reset IRQ
     F971   01
     F972   81
1609 F973   A2   RCVDMY MOVP   %HSKSET,BCNTL            let hsk float
     F974   00
     F975   81
1610 F976   72          MOV    %WAKEUP/256,INT2V-1      set up INT2 vector
     F977   F9
     F978   16
1611 F979   72          MOV    %WAKEUP-(256*(WAKEUP/256)),INT2V
     F97A   92
     F97B   17
1612 F97C   72          MOV    %BROPC,INT2              set up vector branch
     F97D   8C
     F97E   15
1613 F97F   A2          MOVP   %>80,TIME
     F980   80
     F981   02
1614 F982   A2          MOVP   %>9F,TIMER               start timer
     F983   9F
     F984   03
1615 F985   A2          MOVP   %I2CS,IOCNTL             select & clear INT2
     F986   4C
     F987   00
1616 F988   05          EINT
1617 F989   A6   RCVDM1 BTJOP  %IRQ,BSTAT,RCVDMO        test for active bus
     F98A   08
     F98B   81
     F98C   E2
1618 F98D   A6          BTJOP  %I2S,IOCNTL,RCVDM1       test for INT2 selected
     F98E   04
     F98F   00
     F990   F8
1619 F991   0A          RETS
```

```
1621                    *-------------------------------------------------------
1622                    * interrupt 1 routine for wake-up
1623 F992   A2  WAKEUP MOVP   %I123C,IOCNTL          clear INT1,2&3
     F993   6A
     F994   00
1624 F995   A2         MOVP   %STOP,TIMER            stop timer
     F996   FF
     F997   03
1625 F998   0B         RETI
1626                    *-------------------------------------------------------
1627                    * interrupt 2 routine for wafer write data
1628 F999   D2  WBITDT DEC    BITCON                 decrement bit counter
     F99A   02
1629 F99B   77         BTJZ   %1,BITCON,WINVRS       test for 2nd bit-half
     F99C   01
     F99D   02
     F99E   04
1630 F99F   92         MOVP   B,WAFER                output true bit-half
     F9A0   0A
1631 F9A1   CC         RR     B                      rotate data right
1632 F9A2   0B         RETI
1633 F9A3   A5  WINVRS XORP   %INVBIT,WAFER          inverted output
     F9A4   01
     F9A5   0A
1634 F9A6   0B         RETI
1635                    *-------------------------------------------------------
1636                    * interrupt 2 routine for wafer read data
1637 F9A7   A4  RBITDT ORP    %I23C,IOCNTL           clear interrupt 3 flag
     F9A8   68
     F9A9   00
1638 F9AA   D2         DEC    BITCON                 decrement bit counter
     F9AB   02
1639 F9AC   CC         RR     B                      rotate data right
1640 F9AD   A7  RBITD2 BTJZP  %I3FLAG,IOCNTL,RBITD2  wait for int3 flag
     F9AE   20
     F9AF   00
     F9B0   FC
1641 F9B1   A2         MOVP   %START,TIMER           restart timer
     F9B2   80
     F9B3   03
1642 F9B4   A7         BTJZP  %INPUT,TEST,RBITD3     test for 0 input bit
     F9B5   80
     F9B6   04
     F9B7   02
1643 F9B8   54         OR     %SETBIT,B              if not zero, set bit
     F9B9   08
1644 F9BA   A4  RBITD3 ORP    %I2C,IOCNTL            clear INT2
     F9BB   48
     F9BC   00
1645 F9BD   0B         RETI
1646                    *-------------------------------------------------------
1647                    * interrupt 2 routine for restart bit time sync
1648 F9BE   A2  ISYNC2 MOVP   %STOP,TIMER
     F9BF   FF
     F9C0   03
1649 F9C1   0B         RETI
1650                    *-------------------------------------------------------
```

```
1651            * branches using ram vectors
1652            * delete RAM label INT2 and remove comments
1653            *INT2   BR      *INT2V                  indirect branch
```

```
1655                    *---------------------------------------------------------------
1656                    * first-pass command-decode table
1657 F9C2 F1C2  CTABLE DATA   XCLODO                    00 open: no open;read BL
1658 F9C4 F093         DATA   XOPNDR                    01 close: open
1659 F9C6 F093         DATA   XOPNDR                    02 close/delete: open
1660 F9C8 F093         DATA   XOPNDR                    03 read: open
1661 F9CA F093         DATA   XOPNDR                    04 write: open
1662 F9CC F093         DATA   XOPNDR                    05 position record: open
1663 F9CE F1D0         DATA   XCLODD                    06 delete: no open;read
1664 F9D0 F193         DATA   XRSTAT                    07 return status
1665 F9D2 F07D         DATA   UNSUPP                    08 unsupported srvreq en
1666 F9D4 F07D         DATA   UNSUPP                    09 unsupported srvreq di
1667 F9D6 F07D         DATA   UNSUPP                    0A unsupported was it yo
1668 F9D8 F07D         DATA   UNSUPP                    0B unsupported you are m
1669 F9DA F093         DATA   XOPNDR                    0C verify: open
1670 F9DC F200         DATA   XCLODF                    0D format: no open;skip
1671 F9DE F1F6         DATA   XCLODC                    0E catalog: no open
1672                    *---------------------------------------------------------------
1673                    * second-pass command-decode table
1674 F9E0 F242  CTABLX DATA   XOPEN                     00 open address
1675 F9E2 F0D3         DATA   XCLOSE                    01 close address
1676 F9E4 F0A2         DATA   XCLSDL                    02 close/delete address
1677 F9E6 F168         DATA   XREAD                     03 read data address
1678 F9E8 F126         DATA   XWRITE                    04 write data address
1679 F9EA F0F0         DATA   XPOSIT                    05 position record addre
1680 F9EC F0A8         DATA   XDELET                    06 delete address
1681                    *===============================================================
1682                    * unused space in table
1683 F9EE   20  BLANKS BYTE   BLANK                       free table space
1684 F9EF   20         BYTE   BLANK
1685 F9F0   20         BYTE   BLANK
1686                    *---------------------------------------------------------------
1687                    * interrupt 3 routine for bit time sync
1688 F9F1   A2  ISYNC3 MOVP   %BEGINR,TIMER             restart timer for sync
     F9F2   82
     F9F3   03
1689 F9F4   A2         MOVP   %I3C23S,IOCNTL            clear INT3,select 2&3
     F9F5   74
     F9F6   00
1690 F9F7   0B         RETI
1691                    *===============================================================
1692 F9F8 F100         DATA   XVERIF                    0C verify address
1693 F9FA F328         DATA   XFORMA                    0D format address
1694 F9FC F33C         DATA   XCATAL                    0E catalog address
```

```
1696              *---------------------------------------------------------
1697              * vectors
1698 FFD0                 AORG   >FFD0
1699 FFD0 0000           DATA   ZERO                trap 23
1700 FFD2 0000           DATA   ZERO                trap 22
1701 FFD4 0000           DATA   ZERO                trap 21
1702 FFD6 0000           DATA   ZERO                trap 20
1703 FFD8 0000           DATA   ZERO                trap 19
1704 FFDA 0000           DATA   ZERO                trap 18
1705 FFDC 0000           DATA   ZERO                trap 17
1706 FFDE 0000           DATA   ZERO                trap 16
1707 FFE0 0000           DATA   ZERO                trap 15
1708 FFE2 0000           DATA   ZERO                trap 14
1709 FFE4 0000           DATA   ZERO                trap 13
1710 FFE6 F8C3           DATA   TSTEOT              trap 12 test for EOT
1711 FFE8 F8BE           DATA   TSTBIT              trap 11 test bit counter
1712 FFEA F7A9           DATA   ROLLEM              trap 10 start wafer oper
1713 FFEC F795           DATA   FNDEOT              trap 09 find EOT
1714 FFEE F917           DATA   RNXHSK              trap 08 receive PAB/data
1715 FFF0 F9F1           DATA   ISYNC3              trap 07 ampl defaults to
1716 FFF2 F936           DATA   WNXHSK              trap 06 transmit respons
1717 FFF4 F931           DATA   XMTPAB              trap 05 transmit respons
1718 FFF6 F07F           DATA   WTERR               trap 04 error handler
1719 FFF8 F9F1           DATA   ISYNC3              interrupt 3
1720 FFFA 0015           DATA   INT2                interrupt 2
1721 FFFC F992           DATA   WAKEUP              interrupt 1
1722 FFFE F006           DATA   INIT                reset
NO ERRORS, NO WARNINGS
```

```
ACTIVE  0080  0290  0500 1424 1431 1463
AREG    0000  0054
ATTRIB  0033  0078  0511 0512 0525 0554 0555 0582 0664 0667 0668
                    0694 0732 0752 0753 0755 0761 0766 0771
BAV     0002  0256  0954 1180 1215 1267 1587 1599
BCNTL   0181  0127  0404 0407 0545 0560 0593 0956 0958 1004 1183
                    1184 1218 1219 1269 1271 1327 1554 1556 1572
                    1577 1589 1602 1608 1609
BDATA   0180  0126  0957 1003 1182 1217 1270 1315 1318 1325 1590
                    1601
BEGINR  0082  0259  1688
BEGINW  0080  0260  1381
BITCON  0002  0056  0858 0859 0879 0880 0896 0897 0911 0912 0920
                    0921 0939 0940 0950 0951 0980 0981 0997 0998
                    1006 1012 1013 1054 1056 1062 1074 1078 1084
                    1096 1100 1104 1109 1113 1117 1119 1130 1131
                    1142 1143 1155 1156 1176 1177 1193 1194 1211
                    1212 1230 1231 1245 1246 1255 1256 1263 1264
                    1286 1287 1292 1293 1331 1332 1345 1354 1355
                    1364 1365 1379 1483 1484 1505 1628 1629 1638
BITEST  000B  0390  1098 1102 1106 1111 1115 1118
BITIME  000A  0261  0269 1378
BLANK   0020  0335  0637 1683 1684 1685
BLANKS  F9EE  1683
BLEN    0039  0081  0089 0090 0803 0804
BLOPEN  0035  0079  0629 0734 0735 0736 0746 0772 0773 0774 0775
                    0777 0780 0783 0784
BOSTIM  004B  0264  1054
BOTIME  0096  0265  1345
BREG    0001  0055
BROPC   008C  0336  1059 1091 1376 1612
BSTAT   0181  0128  0954 0955 1180 1181 1215 1216 1267 1268 1587
                    1588 1599 1600 1617
BYNAME  F810  1422
BYNMBR  F84A  1442  1421 1442
CAPTUR  0003  0119  1066
CCATAL  000E  0344  0660
CCLOSE  0001  0346
CCLSDL  0002  0347
CCODE   003D  0084  0094 0432 0518 0659 0675 1001 1304
CDELET  0006  0348
CFORMA  000D  0345  0672
CHKSUM  0014  0062  0063 0064 0856 0857 0885 0886 0915 0916 0917
                    0918 0960 0961 0977 1063 1064 1071 1072 1079
                    1081 1082 1083 1085 1087 1088 1139 1140 1197
                    1198 1199 1200 1221 1222 1235 1236 1252 1253
                    1277 1278 1305 1307
CMPFIL  F884  1474  1429 1467
CMPFLO  F886  1475
CMPFL1  F88B  1477
CMPFL2  F89B  1483  1483
CNULL   00FE  0349  0433
COPEN   0000  0350  0662 0676
COUNT   0012  0061  0411 0430 0449 0452 0575 0598 0615 0618 0630
                    0633 0640 0641 0644 0742 0745 0748 0813 0817
                    0821 0827 0833 0874 0895 0979 0996 1122 1141
                    1291 1299 1564 1569 1581
```

LABEL	VALUE	DEFN	REFERENCES
CPOSIT	0005	0353	
CREAD	0003	0351	1310 1321
CRESET	00FF	0352	0435 0518
CRSTAT	0007	0354	
CTABLE	F9C2	1657	0440 0442
CTABLX	F9E0	1674	0683 0685
CVERIF	000C	0355	1312 1323
CWRITE	0004	0356	1001
DATA	001A	0067	0068 0605 0607 0609 0611 0617 1265 1273 1274 1508 1522
DATAP	0019	0065	0066 0067 0412 0450 0453 0576 0614 0617 0629 0711 0743 0746 0749 0811 0812 0816 0820 0826 0834 0855 0878 0883 0884 0977 0978 0984 0985 1044 1123 1129 1135 1138 1244 1250 1251 1289 1290 1297 1298 1353 1482 1561 1562 1563 1573 1574 1580
DCODE	003E	0085	0086 0095
DDRD	000B	0125	0422 0429
DFORMA	0040	0291	0795
DIRECT	003F	0096	
DISABL	000C	0248	0404
DISPLY	0010	0292	0758
DLEN	0037	0080	0087 0088 0447 0448 0450 0566 0569 0571 0572 0612 0613 0614 0625 0633 0635 0707 0740 0741 0743 0809 0810 0811 0900 0901 0915 0917 0924 0926 1042 1146 1147 1289 1305 1307 1349 1477
DLPCNT	0019	0066	0914 0923 0927 0928 0938 1154 1159 1161 1165 1166 1174 1185 1187 1191 1192
DRIVE	0006	0123	0399 0408 0463 0862 0864 0868 0888 0890 0894 0931 0933 0937 0963 0965 0969 1011 1024 1302 1382 1510 1512 1514 1519 1521
DROP	0001	0249	1004 1183 1218 1327 1602
DWNTIM	004B	0266	
EDIREC	F097	0477	
ENDRCT	007F	0113	1044 1353
EOFFLG	0002	0293	0610
EOM	F017	0407	0417 0421 0434 0455 0601
EOM2	F07B	0455	0509 0519
EOPEN	F09D	0481	
EOT	0009	0385	1035 1343
EOTCHK	000C	0386	0986 1014 1366
EOTCN2	0023	0268	0893 1518
EOTCNT	0046	0267	0867 0936 0968
EOTFLG	0080	0294	
EOTIMR	001B	0068	0069 0844 0861 0863 0867 0887 0889 0893 0930 0932 0936 0962 0964 0968 1041 1362 1509 1511 1518
EOTTST	0002	0240	0866 0892 0935 0967 1516
ERRWT	0004	0384	0459 0477 0481 0505 0526 0536 0540 0556 0583 0587 0627 0657 0665 0670 0679 0695 0700 0759 0764 0806 0904 0907 1030 1206 1338 1595
FDIREC	0080	0295	0476 0653 1027 1309
FDROPN	0090	0296	0509 0656
FEOD1	F879	1467	1469
FEOD2	F87C	1468	1462 1463
FEOD3	F868	1460	1427
FEOF2	F852	1449	1447

FEOT1	F79C	1364	1364	1367						
FERROR	0001	0297	1137	1276	1314	1352	1430			
FFLAG	001C	0069	0070	0402	0425	0428	0476	0480	0488	0504	0509
			0517	0529	0535	0606	0608	0610	0648	0651	0653
			0656	0669	0678	0691	0704	0737	0738	0854	1027
			1128	1137	1276	1309	1314	1352	1363	1367	1414
			1418	1421	1426	1430	1459	1465	1517		
FFOUND	0008	0298	0529	0535	0678	0691	0704	1367	1418	1459	1517
FILE0	0040	0097	0499	0502	0795	1423	1424	1530	1532	1534	1536
			1542	1544	1546	1548					
FILE1	0044	0098									
FILE2	0048	0099									
FILE3	004C	0100									
FILE4	0050	0101									
FILE5	0054	0102									
FILE6	0058	0103									
FILE7	005C	0104									
FILE8	0060	0105									
FILE9	0064	0106									
FILEA	0068	0107									
FILEB	006C	0108									
FILEC	0070	0109									
FILED	0074	0110									
FILEE	0078	0111									
FILEF	007C	0112	0699								
FMSSNG	00F7	0313	0854	1363	1426	1465					
FNAME	0020	0299	0648	1414	1421						
FNAME1	0032	0077	0638	0649	0711	0820	1482				
FNAMEL	0024	0076									
FNDEOD	F865	1459	0703	0768							
FNDEOF	F84C	1446	1439	1455	1468						
FNDEOT	F795	1361	1713								
FNUMBR	00DF	0314	0651								
FOPEN	0010	0300	0480	0608							
FORGET	F693	1237	1401	1454							
FORGT2	F682	1230	1230	1239							
FREE	0000	0337									
FRESTO	0004	0301									
FWP	0040	0302	0428	0488	0606	0669					
GONE	F95F	1595	1587	1599							
HSK	0001	0254	1181	1216	1600						
HSKSET	0000	0250	0545	0560	0593	0958	1184	1219	1271	1554	1556
			1572	1577	1609						
I123C	006A	0273	0403	1026	1335	1623					
I1CS	0043	0272	0409								
I23C	0068	0277	1637								
I23CS	007C	0278	1061								
I2C	0048	0274	1644								
I2CS	004C	0275	1095	1380	1615						
I2S	0004	0276	1618								
I3C	0060	0279									
I3C23S	0074	0281	1689								
I3CS	0070	0280	1077								
I3FLAG	0020	0282	1640								
INACTV	007F	0315	0491	0719							
INHIB	0004	0251	0407								
INIFLG	0060	0303	0402								

INIT	F006	0399	1722								
INITDR	00FF	0226	0399								
INITWF	0001	0227	0429								
INPUT	0080	0241	1642								
INT2	0015	0063	1059	1091	1376	1612	1720				
INT2V	0017	0064	0065	1057	1058	1089	1090	1374	1375	1610	1611
INTDIS	0008	0304	0732								
INTERN	0010	0305	0733								
INVBIT	0001	0228	1633								
IOCNTL	0000	0117	0403	0409	1026	1061	1077	1095	1335	1380	1615
			1618	1623	1637	1640	1644	1689			
IRQ	0008	0255	0955	1268	1588	1617					
ISYNC2	F9BE	1648	1057	1058	1058						
ISYNC3	F9F1	1688	1715	1719							
ITSME	F037	0422	0415								
LAST	0040	0306	0493	0501	0698	0699	0716	0763	1423	1440	1462
			1469								
LASTO	00BF	0316	0494	0717	0725						
LDFIL2	F902	1542									
LDFILD	F900	1540	0715	0724	1438	1461	1476				
LSN	000F	0338	0825	0959	1220	1234	1272	1273	1591		
LUNO	003C	0083	0093								
MAXLEN	0020	0071	0072	0566	0569	0571	0572	0730	0731	0774	0775
			0777	0780	0783	0784	0826	1529	1531	1543	1545
ME2	F042	0426	0426								
ME3	F04B	0429	0427								
MT	00F7	0229	1382								
MTSN	00F3	0230									
MTSNWE	00F1	0231	0864	0890	0933	0965					
MTWE	00F5	0232	0868	0894	0937	0969					
NEWFIL	00C0	0307	0728								
NFILE	0023	0073	0075	0489	0513	0702	0705	0721	0816	0847	0855
			1038	1123	1346	1411	1419	1425	1433	1460	1474
			1475								
NIBCON	0012	0060	0061	0062	0845	0850	0853	0869	0870	0900	0901
			0947	0948	0970	0971	1010	1023	1036	1146	1147
			1152	1153	1162	1164	1172	1173	1188	1190	1197
			1199	1201	1202	1223	1224	1237	1238	1254	1279
			1280	1330	1333						
NMLEN	000F	0074	0075	0076	0821						
NONSN	0004	0233	1520								
NREC	0022	0072	0073	0542	0543	0558	0559	0563	0565	0590	0591
			0708	0709	0749	0848	0849	1039	1040	1347	1348
			1387	1396	1397	1405	1449	1450	1478	1479	1490
			1493	1497	1501						
NUFLG1	0094	0308	0737								
NUFLG2	00FC	0309	0738								
OPNRST	00EF	0317	0504	0517							
POSIT1	F81D	1426	1435								
POSIT2	F822	1428	1440								
POSIT3	F825	1429	1423	1424	1441						
POSIT4	F831	1433	1430	1431							
POSIT5	F83E	1438	1422	1436							
POSITN	F805	1418	0674								
PREPOS	F7FA	1411	1390								
PSCALE	0003	0120									
RBITD2	F9AD	1640	1640								

```
RBITD3  F9BA  1644  1642
RBITDT  F9A7  1637  1089  1090  1090
RBYTE   F69C  1244  1357
RBYTE2  F69E  1245  1245  1254
RBYTE3  F6B5  1255  1226  1240  1255  1282
RCHKE2  F76B  1338  1352
RCHKS1  F704  1292  1292  1299
RCHKS2  F73C  1318  1311  1314
RCHKS3  F743  1321  1306  1308
RCHKS4  F74B  1325  1322
RCHKS5  F751  1327  1317  1320
RCHKSE  F768  1337  1324
RCHKSM  F6F5  1286  1259  1286  1487
RCMPB2  F6C0  1263  1263  1281
RCMPB3  F6E6  1277  1275
RCMPM2  F5B2  1130  1130  1141
RCMPM3  F5C3  1138  1136
RCMPM4  F5CE  1142  1142
RCOMPB  F6EC  1279  0549
RCOMPM  F5AD  1128  0546  0594  1351  1399  1452  1481  1486
RCVCNT  0008  0388  0431  0631  0642  0645
RCVDM0  F96F  1607  1617
RCVDM1  F989  1617  1618
RCVDMY  F973  1609  0471
RCVNIB  F94F  1587  1555  1558  1588
RCVPAB  F91A  1555  0413
RDBIT1  00A4  0283
RDBIT2  0020  0284
RDBIT3  0000  0285
RDIREC  F76D  1343  0654  1428
RDL     F5DD  1152  0547  1400  1453
RDL1    F5E4  1155  1155  1166
RDL2    F5F2  1161  1159
RDL3    F5FB  1164  1161
RDL4    F601  1166  1163
RDLX    F606  1172  0595
RDLX1   F60D  1176  1176  1192
RDLX2   F62B  1187  1185
RDLX3   F634  1190  1187
RDLX4   F63A  1192  1189
RDLX5   F63D  1193  1168  1193
RECFIL  001E  0070  0071  0491  0493  0494  0563  0564  0565  0698  0716
                    0717  0719  0725  0728  0729  0733  0757  0763  0824
                    0825  0830  1431  1440  1462  1463  1469  1497  1499
                    1533  1535  1547  1549
REGS    0011  0059  0060
RELEAS  0001  0252  0956  1269  1589  1608
RESYNC  F524  1057  1112  1116
RETDL   F06A  0447  0472  0521  0530  0799
RETDL2  F074  0451  0750
REV0    0000  0310
RHELP   F656  1206  1180  1181  1215  1216  1267  1268
RLAST   F757  1331  1331  1333
RLAST0  F754  1330  1313  1337
RMEM    F78B  1354  1354
RNIB    F679  1223  0596
RNIB2   F658  1211  1211  1225
```

RNUM	003B	0082	0091	0092	1387	1405	1490	1492			
RNXHSK	F917	1554	1565	1714							
ROLLEM	F7A9	1371	1712								
RSATTR	0002	0359	0527	0666	0760						
RSBLEN	000C	0360	0782	0807							
RSCHAR	0001	0361	0628								
RSCLOS	0004	0362	0482								
RSDATA	0019	0363	1325	1326							
RSDEVI	0006	0364	1339								
RSDLEN	0008	0365									
RSDRCT	0051	0366	0478								
RSEOF	0007	0367	0541	0588	1392						
RSEOT	0050	0368	0905	1031							
RSFILE	000B	0369	0701								
RSFIND	0003	0370	0537	0680	0696						
RSLAST	0052	0371	0765								
RSLOWB	0010	0372									
RSOK	0000	0373	0798	0837	1318						
RSOPEN	0005	0374	0658								
RSPROT	0009	0375	0506	0671							
RSREAD	000F	0376	0584								
RSSUPP	000D	0377	0460								
RSTEOF	00FD	0318									
RSTEOT	007F	0319									
RSTERR	00FE	0320	1128								
RSTIME	00FF	0378	0908	1207	1596						
RSTWRT	00FD	0321									
RSVERI	0018	0379	1315	1316							
RSWRIT	000E	0380	0557								
RSYNC	F519	1053	0544	0592	1398	1451	1480				
RSYNC0	F51D	1055	1350								
RSYNC1	F520	1056	1056								
RSYNC2	F533	1062	1068	1070							
RSYNC3	F53A	1065	1074								
RSYNC4	F546	1071	1069								
RSYNC5	F55C	1081	1079								
RSYNC6	F55F	1082	1080	1084							
RSYNC7	F582	1100	1107								
RSYNC8	F589	1104	1099								
RSYNC9	F590	1109	1103								
RTDL2	F159	0573	0568	0570							
RTNREC	F2CC	0748									
RTSTA2	F077	0453									
RTSTAT	F075	0452	0578	0621	0838						
SAB	003E	0075	0076	0077	0078	0079	0080	0081	0082	0083	0084
			0085	0412							
SCREWD	0007	0387									
SEARC1	F7CB	1390	1408								
SEARC2	F7CE	1391	1389	1402	1407						
SEARC3	F7EE	1403	1395	1406							
SEARC4	F7F0	1404	1393								
SEARC5	F7F1	1405	1388								
SEARCH	F7C4	1387	0528	0534							
SENSET	0004	0242	0862	0888	0931	0963	1510				
SEQUEN	0020	0311	0664								
SETBIT	0008	0245	1099	1103	1107	1112	1116	1643			
SN	00FB	0234	1513								

SDM	F021	0411									
STACK	0003	0057	0059	0469							
STACKL	000E	0058	0059								
START	0080	0262	1641								
STATUS	003E	0086	0453	0468	0520	0573	0576	0597	0620	0739	0782
			0798	0837	1316	1319	1326	1392	1403		
STFIL2	F8EB	1529	0495	0720	0726						
STFILD	F8E9	1527	0492	0514	1412						
STOP	00FF	0235	0401	0408	0463	0464	1024	1025	1302	1334	1624
			1648								
SYNCDT	00AA	0339	0860								
TEMP0	003E	0095									
TEMP1	003D	0094	0824	0830	0831	0834					
TEMP2	003C	0093	0832	1420	1434	1436	1437				
TEMP3	003B	0092									
TEMP4	003A	0091	0702	0714							
TEMP5	0039	0090									
TEMP6	0038	0089									
TEMP7	0037	0088									
TEMP8	0036	0087									
TEOF2	F8BD	1502	1498								
TEST	0004	0122	0418	1642							
TIME	0002	0118	1060	1086	1378	1613					
TIMER	0003	0121	0401	0464	1025	1055	1334	1344	1381	1614	1624
			1641	1648	1688						
TIMEX	000A	0389	0843	1053	1301	1361					
TREC2	F8B1	1494	1491								
TSTBIT	F8BE	1505	1505	1711							
TSTE01	F8D6	1516	1510								
TSTE02	F8DD	1518	1516								
TSTE03	F8E0	1519	1509								
TSTE04	F8E6	1522	1515								
TSTEOF	F8B2	1497	0538	0585	1391	1446					
TSTEOT	F8C3	1508	1710								
TSTI	0080	0326	0512	0668							
TSTIU	0040	0327	0582	0752	0761						
TSTNA	00C0	0328	0525								
TSTNI	00C0	0329									
TSTO	0080	0330	0753								
TSTOA	0040	0331	0511	0554	0667	0694					
TSTREC	F8A8	1490	1394								
TSTU	0080	0332	0555	0766	0771						
UNSUPP	F07D	0459	0438	1665	1666	1667	1668				
USED	00FB	0322									
WAFER	000A	0124	0423	0427	0866	0892	0935	0967	1516	1630	1633
WAKEUP	F992	1623	1610	1611	1611	1721					
WBITDT	F999	1628	1374	1375	1375						
WBYTE	F3D1	0878	1045								
WBYTE0	F3D3	0879	0879	0895							
WBYTE1	F3F3	0892	0888								
WBYTE2	F3FA	0894	0887								
WBYTE3	F3FD	0895	0891								
WBYTE4	F400	0896	0896								
WCHKS0	F4B5	0980	0980	0996							
WCHKS3	F4C2	0996									
WCHKS4	F4C5	0997	0997								
WCHKS5	F4D8	1006	1002	1006							

LABEL	VALUE	DEFN	REFERENCES			
WCHKSM	F4AD	0977	1046			
WDIREC	F4FC	1035	0516	0796	1413	
WDL	F413	0911	0562	0911		
WDL0	F42B	0920	0920	0938		
WDL1	F433	0923	0919			
WDL2	F43B	0926	0923			
WDL3	F440	0928	0925			
WDL4	F445	0930	0928			
WDL5	F453	0935	0931			
WDL6	F45A	0937	0930			
WDL7	F45D	0938	0934			
WDL8	F460	0939	0939			
WE	00FD	0236	1011			
WEOT	F40F	0904	0866	0892	0935	
WEOT2	F4FA	1030	0967			
WHELP	F411	0907	0954	0955		
WINVRS	F9A3	1633	1629			
WLAST	F4E2	1012	1012	1023		
WLAST3	F4EA	1023				
WMEM	F513	1045	0712			
WNIB	F468	0947				
WNIB0	F46E	0950	0950	0972		
WNIB1	F49C	0967	0963			
WNIB2	F4A3	0969	0962			
WNIB3	F4A6	0970	0949	0966		
WNXHSK	F936	1572	1582	1716		
WNXPA2	F941	1577	0577	0600		
WNXPAB	F939	1573	1570			
WP	0004	0243	0427			
WP0	00BF	0323	0425			
WPDLAY	000A	0269	0424			
WPSENS	0080	0237	0422			
WPSN	007F	0238	0423			
WRBIT1	00D6	0286				
WRBIT2	0002	0287				
WSYNC	F384	0843	0561	0710		
WSYNC0	F399	0853	0847	0848	0849	1043
WSYNC1	F3A5	0858	0858	0871		
WSYNC2	F3BC	0866	0862			
WSYNC3	F3C3	0868	0861			
WSYNC4	F3C6	0869	0865			
WTERR	F07F	0463	1718			
XCATA2	F347	0809	0803	0805		
XCATA3	F381	0838				
XCATAL	F33C	0803	1694			
XCLOD0	F1C9	0629	0626			
XCLOD1	F1DD	0638	0639			
XCLOD2	F1EF	0648	0643			
XCLOD3	F206	0659	0656			
XCLOD4	F216	0667	0664			
XCLOD5	F21E	0669	0663	0667		
XCLOD6	F224	0672	0661	0668	0669	
XCLOD7	F237	0681	0673	0677	0678	
XCLOD8	F239	0683	0484			
XCLODC	F1F6	0651	1671			
XCLODD	F1D0	0633	1663			
XCLODF	F200	0656	0653	1670		

XCLODO	F1C2	0625	1657								
XCLODR	F1F9	0653	0649								
XCLODX	F1C7	0627	0634	0635							
XCLODY	F1EB	0644	0640								
XCLOFL	F0E3	0517	0512								
XCLOSE	F0D3	0511	1675								
XCLOST	F0DB	0513	0511								
XCLOWD	F0E0	0516	0493	0503							
XCLSD2	F0CA	0504	0488								
XCLSDL	F0A2	0488	1676								
XDELE1	F0B8	0496	0500								
XDELE2	F0C3	0501	0497								
XDELET	F0A8	0491	1680								
XFORM1	F32B	0793	0794								
XFORMA	F328	0791	1693								
XMIT	0005	0391	0814								
XMTCNT	0006	0392	0451	0454	0616	0619	0744	0747	0818	0822	0828
			0835								
XMTNIB	F961	1599	1576	1579	1600						
XMTPAB	F931	1569	1717								
XOPEN	F242	0691	1674								
XOPEN0	F24D	0695									
XOPEN1	F2A7	0734	0732	0769	0776						
XOPEN2	F2B1	0737	0734	0735	0778	0781					
XOPEN3	F2B9	0740	0785								
XOPEN4	F2D5	0752	0692								
XOPEN5	F2EE	0763	0771								
XOPEN6	F2F4	0766	0763								
XOPEN7	F2FD	0771	0761								
XOPEN8	F2EA	0761	0758								
XOPEN9	F2E0	0755	0752	0753							
XOPENO	F24F	0698	0754								
XOPND2	F099	0480	0476								
XOPND3	F09F	0484	0480								
XOPNDR	F093	0476	1658	1659	1660	1661	1662	1669			
XOPNEW	F249	0694	0691								
XOPNI0	F311	0777	0772	0773							
XOPNI1	F31D	0782	0779								
XOPNIU	F301	0772	0766								
XOPNO0	F253	0699	0694								
XOPNO1	F259	0702	0699								
XOPNO2	F283	0719	0716								
XOPNO3	F286	0720	0718								
XOPNO4	F297	0728	0698	0723							
XOPNO5	F265	0707	0704								
XPOSI0	F0F6	0528	0525								
XPOSIT	F0F0	0525	1679								
XREAD	F168	0582	1677								
XREAD2	F16E	0585	0582								
XREAD3	F175	0590	0586								
XREAD4	F189	0597	0550								
XRESET	F0CF	0509	0436								
XRSTA1	F19D	0608	0606								
XRSTA2	F1AB	0612	0608	0610							
XRSTAT	F193	0605	1664								
XVERI0	F107	0536	0529								
XVERI1	F109	0538	0535								

XVERI2	F110	0542	0539								
XVERIF	F100	0534	1692								
XWRIT2	F130	0558	0554	0555							
XWRIT3	F153	0571	0567								
XWRITE	F126	0554	1678								
ZERO	0000	0340	1699	1700	1701	1702	1703	1704	1705	1706	1707
			1708	1709							