```
MICROJW1     MLP FAMILY ASSEMBLER    1.0              16:14:  0    3/23/82
 SOURCE ACCESS NAME:          CD3.JJW.SRC.MICROJW1
 OBJECT ACCESS NAME:          CD3.JJW.OBJ.MICROJW1
 LISTING ACCESS NAME:         CD3.JJW.LST.MICROJW1
```

```
0001                    IDT    'MICROJW1'
0002                    OPTION XREF
0003            *       JEF WINSOR
0004            *
0005            * MICROTAPE PERIPHERAL CONTROL SOFTWARE
0006            *-------------------------------------------------------------
0007            * SYSTEM HARDWARE IMPLEMENTATION  ,-------,
0008            *                          ,------¦ADDRESS¦
0009            *,--------,                ¦ ,----¦SELECT ¦
0010            *¦   70C20¦                ¦ ¦ ,--¦       ¦
0011            *¦        ¦                ¦ ¦ ¦ ,-¦_______¦   ,-------,
0012            *¦04  6 A0¦<--ADDR 0------' ¦ ¦ ¦ ,----------¦VOLTAGE¦
0013            *¦    7 A1¦<--ADDR 1-------' ¦ ¦ ¦ ,---------¦CONTROL¦
0014            *¦    8 A2¦<--ADDR 2---------' ¦ ¦ ¦ ,------>¦_______¦
0015            *¦    9 A3¦<--ADDR 3-----------' ¦ ¦ ¦
0016            *¦   10 A4¦<--NC                 ¦ ¦ ¦
0017            *¦   16 A5¦<--LOW BATTERY--------' ¦ ¦   ,-------,
0018            *¦   15 A6¦<--MOTOR PROHIBITED-----' ¦   ¦WP AND ¦
0019            *¦   11 A7¦<--NC                    ¦   ¦EOT/BOT¦
0020            *¦        ¦                  ,------+-->¦SENSORS¦
0021            *¦06  3 B0¦-->LED            ¦      ¦   ¦_______¦
0022            *¦    4 B1¦-->WRITE ENABLE---+------+--,    ¦ ¦
0023            *¦    5 B2¦-->SENSORS ON-----'      ¦  ¦    ¦ ¦
0024            *¦   37 B3¦-->MOTOR ON--------------*--+-,  ¦ ¦
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¦<->CS-----------<->¦_________¦  ¦ ¦ ¦ ¦
0035            *¦   33 C5¦---NC                  ¦  ,----+-+-' ¦
0036            *¦   34 C6¦---NC             ,-----+--' ,--+-+---'
0037            *¦   35 C7¦---NC             ¦  ,--+----'  V V
0038            *¦        ¦                  ¦  ¦  ¦     ,-----,
0039            *¦0A 27 D0¦---WRITE DATA---+--+--+--+--->¦TAPE ¦
0040            *¦   26 D1¦<--EOT/BOT------'  ¦  ¦  ,--¦DRIVE¦
0041            *¦   24 D2¦<--WP--------------'  ¦  ¦  ¦_____¦
0042            *¦   23 D3¦<--INPUT RAW DATA-----+--'     ¦      *input raw d
0043            *¦   22 D4¦<--  --*-- +V ROM     ¦        ¦       may go back
0044            *¦   21 D5¦<--  --*      OPTIONS ¦        ¦       port a to h
0045            *¦   20 D6¦<--  --*              ¦        ¦       micro-code
0046            *¦   19 D7¦<--  --'              ¦        ¦
0047            *¦        ¦                     ¦        ¦
0048            *¦ 13 INT1¦<--IRQ-----------------'       ¦
0049            *¦ 12 INT3¦<--TAPE TRANSITION PULSE-------'
0050            *¦_________¦
```

```
0052            *--------------------------------------------------------
0053            * register file equates
0054    0010    STACKL  EQU     >10                     reserved stack area
0055    0002    BITCON  EQU     R2                      nibble bit-counter
0056    0003    STACK   EQU     R3                      soft stack location
0057    0013    REGS    EQU     STACK+STACKL            start of registers
0058    0014    NIBCON  EQU     REGS+1                  data length nibble-
0059    0014    COUNT   EQU     NIBCON                  16 bit counter
0060    0016    CHKSUM  EQU     REGS+3                  checksum value (16
0061    0018    INT1V   EQU     REGS+5                  int1 ram-vector
0062    001A    INT2V   EQU     REGS+7                  int2 ram-vector
0063    001C    INT3V   EQU     REGS+9                  int3 ram-vector
0064    001E    DATAP   EQU     REGS+11                 data pointer
0065    001E    DLPCNT  EQU     DATAP                   data length pointer
0066    001F    DATA    EQU     REGS+12                 nibble pack/unpack
0067    0029    SAB     EQU     DATA+10                 ms address byte of
0068    0020    STATUS  EQU     SAB-9                   status to be return
0069    0022    DLEN    EQU     SAB-7                   data length
0070    0024    BLEN    EQU     SAB-5                   buufer length
0071    0026    RNUM    EQU     SAB-3                   record number
0072    0027    DQUAL   EQU     SAB-2                   device qualifier
0073    0028    CCODE   EQU     SAB-1                   command code
0074    0029    DCODE   EQU     SAB                     device code
```

```
0076                *-------------------------------------------------------
0077                * peripheral file equates
0078    0000    IOCNTL  EQU     P0                      i/o control
0079    0002    TIME    EQU     P2                      timer start value
0080    0003    CAPTUR  EQU     P3                      timer value @ int3
0081    0003    PSCALE  EQU     P3                      prescale control
0082    0003    TIMER   EQU     P3                      timer control
0083    0004    TEST    EQU     P4                      data & system test-
0084    0006    DRIVE   EQU     P6                      drive & system cont
0085    000A    WAFER   EQU     P10                     wafer i/o
0086    000B    DDRD    EQU     P11                     ddr for port d
0087    0110    BDATA   EQU     >110                    bus data
0088    0111    BCNTL   EQU     >111                    bus control
0089    0111    BSTAT   EQU     >111                    bus status
```

```
0091                *-------------------------------------------------------------
0092                * constant equates
0093  0000  BITIME EQU    >00                          data half-bit time
0094  0003  CINPUT EQU    >03                          input command code
0095  0004  CPRINT EQU    >04                          print command code
0096  0000  DLTIME EQU    >00                          dl half-bit time
0097  0000  ENABLE EQU    >00                          enable ibc
0098  0002  EOT    EQU    >02                          bit-test for EOT
0099  0001  DROP   EQU    >01                          drop hsk bit
0100  0001  HSK    EQU    >01                          bus ready bit-test
0101  0000  HSKSET EQU    >00                          let hsk float
0102  0004  INHIB  EQU    >04                          inhibit bus communi
0103  0000  INITDR EQU    >00                          initialize drive
0104  0001  INITWF EQU    >01                          initialize wafer dd
0105  0008  INPUT  EQU    >08                          wafer data-bit
0106                *                          change to >10 for micro-code
0107                *                          change RBITDT to read from TEST
0108  0001  INVBIT EQU    >01                          wafer-write invert
0109  00AA  I123C  EQU    >AA                          clear int1, 2 & 3 f
0110  0088  I2C    EQU    >88                          clear int2 flag
0111  008C  I2CS   EQU    >8C                          clear and select in
0112  00A8  I23C   EQU    >A8                          clear int2 & 3 flag
0113  00BC  I23CS  EQU    >BC                          clear and select in
0114  00A0  I3C    EQU    >A0                          clear int3 flag
0115  00B0  I3CS   EQU    >B0                          clear and select in
0116  00A0  I3FLAG EQU    >A0                          test int3 flag-bit
0117  0001  IRQ    EQU    >01                          bus data ready bit-
0118  000F  LSN    EQU    >0F                          mask to clear msn
0119  0008  MT     EQU    >08                          turn on motor
0120  000C  MTSN   EQU    >0C                          turn on motor & sen
0121  000A  MTWE   EQU    >0A                          turn on motor & WE
0122  0000  OK     EQU    >00                          ok status-code
0123  0001  RELEAS EQU    >01                          release-hsk bit
0124  0001  SETBIT EQU    >01                          store 1 bit
0125  0080  START  EQU    >80                          timer-start bit
0126  0000  SYNCTM EQU    >00                          sync half-bit time
0127  00FF  TIMOUT EQU    >FF                          time-out error code
0128  0000  ZERO   EQU    >00                          constant
```

```
0130                    *---------------------------------------------------------
0131                    * initialization
0132 0000   B8  INIT    PUSH    A                       r2 is for bitcon
0133 0001   A2          MOVP    %INITDR,DRIVE           initialize drive
     0002   00
     0003   06
0134 0004   A2          MOVP    %INITWF,DDRD            initialize wafer dd
     0005   01
     0006   0B
0135 0007   A2          MOVP    %ZERO,TIMER             reset timer
     0008   00
     0009   03
0136 000A   A2          MOVP    %I123C,IOCNTL           initialize int1,2&3
     000B   AA
     000C   00
0137 000D   A2          MOVP    %ENABLE,BCNTL           enable ibc
     000E   00
     000F   11
0138 0010   05          EINT                            enable interrupts
0139 0011   00  SOM     NOP                             wait start of messa
0140 0012   88          MOVD    %9,COUNT                pab length
     0013 0009
     0015   14
0141 0016   88          MOVD    %SAB,DATAP              data pointer
     0017 0029
     0019   1E
0142 001A   8E          CALL    @RCVPAB                 receive pab
     001B 026E'
0143 001D   7D          CMP     %CINPUT,CCODE           test for input comm
     001E   03
     001F   28
0144 0020   E6          JNE     GTPAB2
     0021   03
0145 0022   8C          BR      @XINPUT                 execute input
     0023 0121'
0146 0025   7D  GTPAB2  CMP     %CPRINT,CCODE           test for print comm
     0026   04
     0027   28
0147 0028   E2          JEQ     XPRINT
     0029   20
0148 002A   E0          JMP     SOM                     handle reset comman
     002B   E5
0149                    *---------------------------------------------------------
0150 002C   88  RETDL   MOVD    %2,COUNT                response length
     002D 0002
     002F   14
0151 0030   88          MOVD    %DLEN,DATAP             data pointer
     0031 0022
     0033   1E
0152 0034   8E          CALL    @XMTPAB                 return zero dl
     0035 0288'
0153 0037   A2          MOVP    %HSKSET,BCNTL           release hsk
     0038   00
     0039   11
0154 003A   88  RTSTAT  MOVD    %1,COUNT                status length
     003B 0001
     003D   14
```

```
0155 003E   88          MOVD    %STATUS,DATAP               data pointer
     003F 0020
     0041   1E
0156 0042   8E          CALL    @XMTPAB                     return status
     0043 0288'
0157 0045   A2          MOVP    %HSKSET,BCNTL               release hsk
     0046   00
     0047   11
0158 0048   E0          JMP     SOM                         wait for som
     0049   C7
```

```
0160                *-------------------------------------------------------
0161                * receive bus & write wafer
0162 004A   88  XPRINT MOVD   %WBITDT,INT2V              set up int2 vector
     004B 0110'
     004D   1A
0163 004E   A2         MOVP   %>FF,TIME                  set up max time
     004F   FF
     0050   02
0164 0051   72         MOV    %5,BITCON                  set up macro loop
     0052   05
     0053   02
0165 0054   A2         MOVP   %I2CS,IOCNTL               select & clear int2
     0055   8C
     0056   00
0166 0057   A2         MOVP   %MTSN,DRIVE                turn on motor & sen
     0058   0C
     0059   06
0167 005A   A7  EOTW   BTJZP  %EOT,WAFER,EOTW            wait for EOT
     005B   02
     005C   0A
     005D   FC
0168 005E   A2         MOVP   %MTWE,DRIVE                turn on motor & WE
     005F   0A
     0060   06
0169 0061   A2         MOVP   %>9F,TIMER                 start BOT timer
     0062   9F
     0063   03
0170 0064   76  BOTW   BTJO   %>FF,BITCON,BOTW           wait for safe area
     0065   FF
     0066   02
     0067   FC
0171 0068   A2         MOVP   %BITIME,TIME               set up sync bit tim
     0069   00
     006A   02
0172 006B   88         MOVD   %99,NIBCON                 set up for 99 nibbl
     006C 0063
     006E   14
0173 006F   72         MOV    %8,BITCON                  set up nibble bit-c
     0070   08
     0071   02
0174 0072   52         MOV    %>5,B                      set up sync data
     0073   05
0175 0074   D5         CLR    CHKSUM                     zero checksum
     0075   16
0176 0076   D5         CLR    CHKSUM-1
     0077   15
0177 0078   48         ADD    DLEN,CHKSUM                add dl to chksum
     0079   22
     007A   16
0178 007B   49         ADC    DLEN-1,CHKSUM-1
     007C   21
     007D   15
0179 007E   A2         MOVP   %>80,PSCALE                start data timer
     007F   80
     0080   03
0180 0081   A2         MOVP   %HSKSET,BCNTL              let hsk float
     0082   00
```

```
      0083  11
0181  0084  76  WSYNC   BTJO  %>0F,BITCON,WSYNC        test for end of nib
      0085  0F
      0086  02
      0087  FC
0182  0088  72          MOV   %>8,BITCON               restart nibble bit-
      0089  08
      008A  02
0183  008B  DA          DJNZ  COUNT,WSYNC              test for end of syn
      008C  14
      008D  F6
0184  008E  98          MOVD  DLEN,NIBCON              move dl to nibcon
      008F  22
      0090  14
0185  0091  B0          CLRC                         ls bit must be zero
0186  0092  DF          RLC   NIBCON                 multiply nibcon by
      0093  14
0187  0094  DF          RLC   NIBCON-1               (byte count -> nibb
      0095  13
0188  0096  B5          CLR   A                      clear A (start nibb
0189  0097  72          MOV   %8,DLPCNT              non-linear dl count
      0098  08
      0099  1E
0190  009A  76  WTSNDL  BTJO  %>0F,BITCON,WTSNDL      wait for end of nib
      009B  0F
      009C  02
      009D  FC
0191  009E  72          MOV   %>8,BITCON               restart nibble bit-
      009F  08
      00A0  02
0192  00A1  C0          MOV   A,B                    move data through q
0193  00A2  76          BTJO  %>C,DLPCNT,WLSB        test for next dl by
      00A3  0C
      00A4  1E
      00A5  04
0194  00A6  12          MOV   DLEN-1,A               get ms byte
      00A7  21
0195  00A8  E0          JMP   WRTDL
      00A9  05
0196  00AA  12  WLSB    MOV   DLEN,A                 get ls byte
      00AB  22
0197  00AC  7A          SUB   %2,DLPCNT              non-linear adjust
      00AD  02
      00AE  1E
0198  00AF  77  WRTDL   BTJZ  %1,DLPCNT,WRTDL2       test for ls/ms nibb
      00B0  01
      00B1  1E
      00B2  01
0199  00B3  B7          SWAP  A                      ms nibble
0200  00B4  DA  WRTDL2  DJNZ  DLPCNT,WTSNDL          test for end of dl
      00B5  1E
      00B6  E3
0201  00B7  76  WRTDL3  BTJO  %>F,BITCON,WRTDL3      wait for end of nib
      00B8  0F
      00B9  02
      00BA  FC
0202  00BB  72          MOV   %8,BITCON              restart bit counter
```

```
      00BC  08
      00BD  02
0203  00BE  C0          MOV     A,B                         move data through q
0204              * above causes last dl nibble to be written twice (need both
0205              * receive bus & write wafer
0206              * critical path for writing to microtape
0207  00BF  76  WNIBDT BTJO    %>0F,BITCON,WNIBDT          wait for end of nib
      00C0  0F
      00C1  02
      00C2  FC
0208  00C3  72          MOV     %>08,BITCON                 restart bit counter
      00C4  08
      00C5  02
0209  00C6  C0          MOV     A,B                         move data through q
0210  00C7  A6          BTJOP   %IRQ,BSTAT,BUSERR           test for bus data r
      00C8  01
      00C9  11
      00CA  53
0211  00CB  A2          MOVP    %RELEAS,BCNTL               reset irq
      00CC  01
      00CD  11
0212  00CE  80          MOVP    BDATA,A                     read data from bus
      00CF  10
0213  00D0  A2          MOVP    %HSKSET,BCNTL               let hsk float
      00D1  00
      00D2  11
0214  00D3  23          AND     %LSN,A                      clear msn for check
      00D4  0F
0215  00D5  48          ADD     A,CHKSUM                    add data to checksu
      00D6  00
      00D7  16
0216  00D8  79          ADC     %0,CHKSUM-1
      00D9  00
      00DA  15
0217              * eot test must be added
0218  00DB  DB  WNIBD2 DECD    NIBCON                      decrement nibble co
      00DC  14
0219  00DD  E3          JC      WNIBDT                      test for end of dat
      00DE  E0
0220              *
0221  00DF  72          MOV     %8,DLPCNT                   non-linear counter
      00E0  08
      00E1  1E
0222  00E2  76  WCHKSM BTJO    %>0F,BITCON,WCHKSM          wait for end of nib
      00E3  0F
      00E4  02
      00E5  FC
0223  00E6  72          MOV     %>8,BITCON                  restart nibble coun
      00E7  08
      00E8  02
0224  00E9  C0          MOV     A,B                         move data through q
0225  00EA  76          BTJO    %>C,DLPCNT,WCLSB            test for ms/ls byte
      00EB  0C
      00EC  1E
      00ED  04
0226  00EE  12          MOV     CHKSUM-1,A                  get ms byte
      00EF  15
```

```
0227 00F0   E0          JMP     WRTCSM
     00F1   05
0228 00F2   12  WCLSB   MOV     CHKSUM,A                get ls byte
     00F3   16
0229 00F4   7A          SUB     %2,DLPCNT               non-linear adjust
     00F5   02
     00F6   1E
0230 00F7   77  WRTCSM BTJZ    %1,DLPCNT,WRTCS2        test for ms/ls nibb
     00F8   01
     00F9   1E
     00FA   01
0231 00FB   B7          SWAP    A                       ms nibble
0232 00FC   DA  WRTCS2 DJNZ    DLPCNT,WTSNDL           test for end of chk
     00FD   1E
     00FE   9B
0233 00FF   76  WLAST   BTJO    %>0F,BITCON,WLAST       wait for end of nib
     0100   0F
     0101   02
     0102   FC
0234 0103   A2          MOVP    %ZERO,DRIVE             turn drive off
     0104   00
     0105   06
0235 0106   72          MOV     %OK,STATUS              set up ok status
     0107   00
     0108   20
0236 0109   88          MOVD    %0,DLEN                 return zero data le
     010A 0000
     010C   22
0237 010D   8C          BR      @RETDL
     010E 002C'
0238                    *---------------------------------------------------------
0239                    * interrupt 2 routine for wafer write
0240 0110   D2  WBITDT DEC     BITCON                  decrement bit count
     0111   02
0241 0112   77          BTJZ    %1,BITCON,WINVRS        test for 2nd bit-ha
     0113   01
     0114   02
     0115   04
0242 0116   92          MOVP    B,WAFER                 output true bit-hal
     0117   0A
0243 0118   CC          RR      B                       rotate data right
0244 0119   0B          RETI
0245 011A   A5  WINVRS XORP    %INVBIT,WAFER           inverted output
     011B   01
     011C   0A
0246 011D   0B          RETI
```

```
0248                *-------------------------------------------------------
0249 011E   8C   BUSERR BR      @HSKERR                    bus time-out error
     011F 0264'
```

```
0251                    *---------------------------------------------------------
0252                    * read wafer & transmit bus
0253 0121   88  XINPUT  MOVD    %WBITDT,INT2V                  set up int2 vector
     0122 0110'
     0124   1A
0254 0125   A2          MOVP    %>FF,TIME                      set up max time
     0126   FF
     0127   02
0255 0128   72          MOV     %6,BITCON                      set up macro loop
     0129   06
     012A   02
0256 012B   A2          MOVP    %I2CS,IOCNTL                   select & clear int2
     012C   8C
     012D   00
0257 012E   A2          MOVP    %MTSN,DRIVE                    turn on motor & sen
     012F   0C
     0130   06
0258 0131   A7  EOTR    BTJZP   %EOT,WAFER,EOTR                wait for EOT
     0132   02
     0133   0A
     0134   FC
0259 0135   A2          MOVP    %>9F,TIMER                     start timer
     0136   9F
     0137   03
0260 0138   88          MOVD    %ISYNC2,INT2V                  set up int2 vector
     0139 0249'
     013B   1A
0261 013C   88          MOVD    %ISYNC3,INT3V                  set up int3 vector
     013D 025A'
     013F   1C
0262 0140   76  BOTR    BTJO    %>FF,BITCON,BOTR               wait for safe area
     0141   FF
     0142   02
     0143   FC
0263 0144   A2          MOVP    %>82,PSCALE                    prescale of 3, star
     0145   82
     0146   03
0264 0147   A2          MOVP    %I23CS,IOCNTL                  clear & select int2
     0148   BC
     0149   00
0265 014A   01          IDLE                                   wait for tape trans
0266 014B   72          MOV     %>20,BITCON                    set up valid-bit co
     014C   20
     014D   02
0267 014E   D5          CLR     CHKSUM                         clear chksum
     014F   16
0268 0150   D5          CLR     CHKSUM-1
     0151   15
0269 0152   01  RSYNC   IDLE                                   wait for next trans
0270 0153   91          MOVP    CAPTUR,B                       read bit time
     0154   03
0271 0155   C4          INV     B                              get true count
0272 0156   48          ADD     B,CHKSUM                       add to cumulative
     0157   01
     0158   16
0273 0159   79          ADC     %0,CHKSUM-1
     015A   00
```

```
     015B   15
0274 015C   05          EINT                                after possible int2
0275 015D   DA          DJNZ    BITCON,RSYNC                test for end of syn
     015E   02
     015F   F2
0276                *
0277                ********** add check for bit time in above loop ***********
0278                *
0279 0160   88          MOVD    %ISTART,INT3V               set up int3 vector
     0161 0260'
     0163   1C
0280 0164   A2          MOVP    %I3CS,IOCNTL                select & clear int3
     0165   B0
     0166   00
0281 0167   72          MOV     %5,BITCON                   divide cumulative b
     0168   05
     0169   02
0282 016A   DD  SHFTIM  RRC     CHKSUM-1                    shift msb right
     016B   15
0283 016C   DD          RRC     CHKSUM
     016D   16
0284 016E   DA          DJNZ    BITCON,SHFTIM               test for end of div
     016F   02
     0170   F9
0285 0171   32          MOV     CHKSUM,B
     0172   16
0286 0173   92          MOVP    B,TIME                      set up 2/3 bit time
     0174   02
0287 0175   D5  PREPDL  CLR     NIBCON                      set nibble count to
     0176   14
0288 0177   D5          CLR     NIBCON-1
     0178   13
0289 0179   D5          CLR     CHKSUM                      set chksum to 0
     017A   16
0290 017B   D5          CLR     CHKSUM-1
     017C   15
0291 017D   01  MIDWT   IDLE                                wait for mid bit
0292 017E   A7          BTJZP   %INPUT,WAFER,MIDWT         test for bit-edge
     017F   08
     0180   0A
     0181   FB
0293 0182   88          MOVD    %RBITDT,INT2V               set up int2 vector
     0183 0235'
     0185   1A
0294 0186   A2          MOVP    %I2CS,IOCNTL                select & clear int2
     0187   8C
     0188   00
0295 0189   72          MOV     %8,DLPCNT                   set up dl pointer/c
     018A   08
     018B   1E
0296 018C   C5  STARTW  CLR     B                           wait for start nibb
0297 018D   72          MOV     %4,BITCON                   3 more zeros
     018E   04
     018F   02
0298 0190   56  START2  BTJO    %>F,B,STARTW
     0191   0F
     0192   F9
```

```
0299 0193  76          BTJO   %>F,BITCON,START2          wait for full start
     0194  0F
     0195  02
     0196  F9
0300 0197  72          MOV    %4,BITCON                  restart nibble bit-
     0198  04
     0199  02
0301 019A  76  RNIBDL  BTJO   %7,BITCON,RNIBDL           wait for end of nib
     019B  07
     019C  02
     019D  FC
0302 019E  72          MOV    %4,BITCON                  restart nibble bit-
     019F  04
     01A0  02
0303 01A1  62          MOV    B,A                        move data through q
0304 01A2  C5          CLR    B                          set all input bits
0305 01A3  A6          BTJOP  %HSK,BSTAT,RBSERR          test for inactive b
     01A4  01
     01A5  11
     01A6  16
0306 01A7  82          MOVP   A,BDATA                    send data over bus
     01A8  10
0307 01A9  A2          MOVP   %DROP,BCNTL                drop hsk
     01AA  01
     01AB  11
0308 01AC  A2          MOVP   %HSKSET,BCNTL              let hsk float
     01AD  00
     01AE  11
0309 01AF  77          BTJZ   %1,DLPCNT,RNIBL2           test for ms/ls nibb
     01B0  01
     01B1  1E
     01B2  01
0310 01B3  B7          SWAP   A                          ms nibble
0311 01B4  76  RNIBL2  BTJO   %>C,DLPCNT,RLSB            test for next byte
     01B5  0C
     01B6  1E
     01B7  08
0312 01B8  44          OR     A,NIBCON-1                 store ms byte
     01B9  00
     01BA  13
0313 01BB  E0          JMP    RNIBL3
     01BC  09
0314       *------     --------  ----------  ---------
0315 01BD  8C  RBSERR  BR     @HSKERR                    bus time out error
     01BE  0264'
0316       *------     --------  ----------  ---------
0317 01C0  44  RLSB    OR     A,NIBCON                   store ls byte
     01C1  00
     01C2  14
0318 01C3  7A          SUB    %2,DLPCNT                  non-linear adjust
     01C4  02
     01C5  1E
0319 01C6  DA  RNIBL3  DJNZ   DLPCNT,RNIBDL               test for end of dl
     01C7  1E
     01C8  D1
0320 01C9  76  RNIBL4  BTJO   %>F,BITCON,RNIBL4          test for end of nib
     01CA  0F
```

```
      01CB  02
      01CC  FC
0321  01CD  72          MOV     %>04,BITCON                 restart bit counter
      01CE  04
      01CF  02
0322  01D0  C5          CLR     B                           set all input bits
0323  01D1  48          ADD     NIBCON,CHKSUM               add dl to checksum
      01D2  14
      01D3  16
0324  01D4  49          ADC     NIBCON-1,CHKSUM-1
      01D5  13
      01D6  15
0325  01D7  DF          RLC     NIBCON                      multiply nibcon by
      01D8  14
0326  01D9  DF          RLC     NIBCON-1                    (byte count -> nibb
      01DA  13
0327              * read wafer & transmit bus
0328              * critical path for reading from microtape
0329  01DB  76  RNIBDT BTJO    %>07,BITCON,RNIBDT          wait for end of nib
      01DC  07
      01DD  02
      01DE  FC
0330  01DF  72          MOV     %>04,BITCON                 restart bit counter
      01E0  04
      01E1  02
0331  01E2  62          MOV     B,A                         move data through q
0332  01E3  C5          CLR     B                           set all input bits
0333  01E4  A6          BTJOP   %HSK,BSTAT,RBSERR           test for inactive b
      01E5  01
      01E6  11
      01E7  D5
0334  01E8  82          MOVP    A,BDATA                     send data over bus
      01E9  10
0335  01EA  A2          MOVP    %DROP,BCNTL                 drop hsk
      01EB  01
      01EC  11
0336  01ED  A2          MOVP    %HSKSET,BCNTL               let hsk float
      01EE  00
      01EF  11
0337  01F0  23          AND     %LSN,A                      clear msn for check
      01F1  0F
0338  01F2  48          ADD     A,CHKSUM                    add data to checksu
      01F3  00
      01F4  16
0339  01F5  79          ADC     %0,CHKSUM-1
      01F6  00
      01F7  15
0340  01F8  DB          DECD    NIBCON                      decrement nibble co
      01F9  14
0341  01FA  E3          JC      RNIBDT                      test for end of dat
      01FB  DF
0342              *
0343  01FC  72          MOV     %8,DLPCNT                   set up non-linear c
      01FD  08
      01FE  1E
0344  01FF  76  RCHKSM BTJO    %7,BITCON,RCHKSM            wait for end of nib
      0200  07
```

```
      0201   02
      0202   FC
0345  0203   72          MOV     %4,BITCON                   restart bit count
      0204   04
      0205   02
0346  0206   62          MOV     B,A                         move data through q
0347  0207   C5          CLR     B                           mask off msn
0348  0208   77          BTJZ    %1,DLPCNT,RCHKS2            test for ms/ls nibb
      0209   01
      020A   1E
      020B   01
0349  020C   B7          SWAP    A                           ms nibble
0350  020D   76  RCHKS2  BTJO    %>C,DLPCNT,RCHKL            test for ms/ls byte
      020E   0C
      020F   1E
      0210   05
0351  0211   44          OR      A,COUNT-1                   store ms byte
      0212   00
      0213   13
0352  0214   E0          JMP     RCHKS3
      0215   06
0353  0216   44  RCHKL   OR      A,COUNT                     store ls byte
      0217   00
      0218   14
0354  0219   7A          SUB     %2,DLPCNT                   non-linear adjust
      021A   02
      021B   1E
0355  021C   DA  RCHKS3  DJNZ    DLPCNT,RCHKSM               test for end of chk
      021D   1E
      021E   E0
0356  021F   A2          MOVP    %ZERO,DRIVE                 turn motor off
      0220   00
      0221   06
0357  0222   4A          SUB     COUNT,CHKSUM                compare chksum
      0223   14
      0224   16
0358  0225   4B          SBB     COUNT-1,CHKSUM-1
      0226   13
      0227   15
0359  0228   E3          JC      CHKSOK                      test chksum
      0229   05
0360  022A   72          MOV     %6,STATUS                   store device error
      022B   06
      022C   20
0361  022D   E0          JMP     RLAST
      022E   03
0362  022F   72  CHKSOK  MOV     %0,STATUS                   store ok status
      0230   00
      0231   20
0363  0232   8C  RLAST   BR      @RTSTAT
      0233 003A'
0364               *-------------------------------------------------------------
0365               * interrupt 2 routine for wafer read
0366  0235   A4  RBITDT ORP     %I3C,IOCNTL                 clear interrupt 3 f
      0236   A0
      0237   00
0367  0238   D2          DEC     BITCON                      decrement bit count
```

```
      0239   02
0368  023A   CC          RR      B                              rotate data right
0369  023B   A7  RBITD2  BTJZP   %I3FLAG,IOCNTL,RBITD2          wait for int3 flag
      023C   A0
      023D   00
      023E   FC
0370  023F   A2          MOVP    %START,TIMER                   restart timer
      0240   80
      0241   03
0371  0242   A7          BTJZP   %INPUT,WAFER,RBITD3            test for 0 input bi
      0243   0B
      0244   0A
      0245   02
0372  0246   54          OR      %SETBIT,B                      if not zero, set bi
      0247   01
0373  0248   0B  RBITD3 RETI
0374              *--------------------------------------------------------------
0375  0249   72  ISYNC2 MOV     %>21,BITCON                    restart sync bit-co
      024A   21
      024B   02
0376  024C   D5          CLR     CHKSUM                         clear chksum
      024D   16
0377  024E   D5          CLR     CHKSUM-1
      024F   15
0378  0250   91          MOVP    CAPTUR,B                       pre-fix count
      0251   03
0379  0252   48          ADD     B,CHKSUM
      0253   01
      0254   16
0380  0255   79          ADC     %0,CHKSUM-1
      0256   00
      0257   15
0381  0258   06          DINT                                   no ints until after
0382  0259   0B          RETI
0383              *--------------------------------------------------------------
0384  025A   A2  ISYNC3 MOVP    %>85,TIMER                     restart timer for s
      025B   85
      025C   03
0385  025D   D2          DEC     BITCON                         decrement bit count
      025E   02
0386  025F   0B          RETI
0387              *--------------------------------------------------------------
0388  0260   A2  ISTART MOVP    %>81,TIMER                     restart timer for s
      0261   81
      0262   03
0389  0263   0B          RETI
0390              *--------------------------------------------------------------
0391  0264   88  HSKERR MOVD    %0,DLEN                        return no data
      0265 0000
      0267   22
0392  0268   72          MOV     %TIMOUT,STATUS                 return time-out err
      0269   FF
      026A   20
0393  026B   8C          BR      @RETDL
      026C 002C'
```

```
0395                    *---------------------------------------------------------
0396                    * receive and store data from bus
0397 026E   DB  RCVPAB  DECD    COUNT                   decrement byte coun
     026F   14
0398 0270   E3          JC      RNXPAB                  test for non-zero d
     0271   04
0399 0272   0A          RETS
0400 0273   A2  RNXHSK  MOVP    %HSKSET,BCNTL           release hsk
     0274   00
     0275   11
0401 0276   8E  RNXPAB  CALL    @RCVNRM                 receive lsd, raise
     0277 02A2'
0402 0279   62          MOV     B,A                     save lsd
0403 027A   8E          CALL    @RCVABN                 receive msd, hold h
     027B 02B1'
0404 027D   C7          SWAP    B                       justify byte
0405 027E   64          OR      B,A                     combine nibbles
0406 027F   9B          STA     *DATAP                  store byte
     0280   1E
0407 0281   DB          DECD    DATAP                   decrement pointer
     0282   1E
0408 0283   DB          DECD    COUNT                   decrement counter
     0284   14
0409 0285   E3          JC      RNXHSK                  test for end of dat
     0286   EC
0410 0287   0A          RETS
0411                    *---------------------------------------------------------
0412                    * load and transmit data to bus
0413 0288   DB  XMTPAB  DECD    COUNT                   decrement counter
     0289   14
0414 028A   E3          JC      WNXPAB                  test for non-zero d
     028B   04
0415 028C   0A          RETS
0416 028D   A2  WNXHSK  MOVP    %HSKSET,BCNTL           release hsk
     028E   00
     028F   11
0417 0290   9A  WNXPAB  LDA     *DATAP                  load data into a
     0291   1E
0418 0292   C0          MOV     A,B                     move nibble into b
0419 0293   8E          CALL    @XMTNRM                 transmit lsd, raise
     0294 02C0'
0420 0296   B7          SWAP    A                       position data
0421 0297   C0          MOV     A,B                     move nibble into b
0422 0298   8E          CALL    @XMTABN                 transmit msd, hold
     0299 02CD'
0423 029B   DB          DECD    DATAP                   decrement pointer
     029C   1E
0424 029D   DB          DECD    COUNT                   decrement counter
     029E   14
0425 029F   E3          JC      WNXHSK                  test for end of dat
     02A0   EC
0426 02A1   0A          RETS
```

```
0428                    *-------------------------------------------------------------
0429                    * receive nibble from bus
0430 02A2   A7  RCVNRM BTJZP   %IRQ,BSTAT,RCVNRM          test for bus data r
     02A3   01
     02A4   11
     02A5   FC
0431 02A6   A2          MOVP    %RELEAS,BCNTL            reset irq
     02A7   01
     02A8   11
0432 02A9   91          MOVP    BDATA,B                  read data from bus
     02AA   10
0433 02AB   A2          MOVP    %HSKSET,BCNTL            let hsk float
     02AC   00
     02AD   11
0434 02AE   53          AND     %LSN,B                   clear msn
     02AF   0F
0435 02B0   0A          RETS
0436 02B1   A7  RCVABN BTJZP   %IRQ,BSTAT,RCVABN         test for bus data r
     02B2   01
     02B3   11
     02B4   FC
0437 02B5   A2          MOVP    %RELEAS,BCNTL            reset irq
     02B6   01
     02B7   11
0438 02B8   91          MOVP    BDATA,B                  read data from bus
     02B9   10
0439 02BA   53          AND     %LSN,B                   clear msn
     02BB   0F
0440 02BC   0A          RETS
```

```
0442                *-------------------------------------------------------------
0443 02BD   8C   RXERR   BR      @HSKERR                         bus timeout
     02BE 0264'
0444                *-------------------------------------------------------------
0445                * transmit nibble to bus
0446 02C0   A7   XMTNRM  BTJZP   %HSK,BSTAT,XMTNRM               test for bus ready
     02C1   01
     02C2   11
     02C3   FC
0447 02C4   92           MOVP    B,BDATA                         send data over bus
     02C5   10
0448 02C6   A2           MOVP    %DROP,BCNTL                     drop hsk
     02C7   01
     02C8   11
0449 02C9   A2           MOVP    %HSKSET,BCNTL                   let hsk float
     02CA   00
     02CB   11
0450 02CC   0A           RETS
0451 02CD   A7   XMTABN  BTJZP   %HSK,BSTAT,XMTABN               test for bus ready
     02CE   01
     02CF   11
     02D0   FC
0452 02D1   92           MOVP    B,BDATA                         send data over bus
     02D2   10
0453 02D3   A2           MOVP    %DROP,BCNTL                     drop hsk
     02D4   01
     02D5   11
0454 02D6   0A           RETS
```

```
 0456                *-------------------------------------------------------
 0457 02D7   9C  INT1    BR      *INT1V                          indirect branch
      02D8   18
 0458 02D9   9C  INT2    BR      *INT2V                          indirect branch
      02DA   1A
 0459 02DB   9C  INT3    BR      *INT3V                          indirect branch
      02DC   1C
 0460                *-------------------------------------------------------
 0461 FFF8               AORG    >FFF8
 0462 FFF8 02DB'         DATA    INT3                            interrupt 3
 0463 FFFA 02D9'         DATA    INT2                            interrupt 2
 0464 FFFC 02D7'         DATA    INT1                            interrupt 1
 0465 FFFE 0000'         DATA    INIT                            reset
NO ERRORS, NO WARNINGS
```

LABEL	VALUE	DEFN	REFERENCES								
BCNTL	0111	0088	0137	0153	0157	0180	0211	0213	0307	0308	0335
			0336	0400	0416	0431	0433	0437	0448	0449	0453
BDATA	0110	0087	0212	0306	0334	0432	0438	0447	0452		
BITCON	0002	0055	0164	0170	0173	0181	0182	0190	0191	0201	0202
			0207	0208	0222	0223	0233	0240	0241	0255	0262
			0266	0275	0281	0284	0297	0299	0300	0301	0302
			0320	0321	0329	0330	0344	0345	0367	0375	0385
BITIME	0000	0093	0171								
BLEN	0024	0070									
BOTR	0140'	0262	0262								
BOTW	0064'	0170	0170								
BSTAT	0111	0089	0210	0305	0333	0430	0436	0446	0451		
BUSERR	011E'	0249	0210								
CAPTUR	0003	0080	0270	0378							
CCODE	0028	0073	0143	0146							
CHKSOK	022F'	0362	0359								
CHKSUM	0016	0060	0175	0176	0177	0178	0215	0216	0226	0228	0267
			0268	0272	0273	0282	0283	0285	0289	0290	0323
			0324	0338	0339	0357	0358	0376	0377	0379	0380
CINPUT	0003	0094	0143								
COUNT	0014	0059	0140	0150	0154	0183	0351	0353	0357	0358	0397
			0408	0413	0424						
CPRINT	0004	0095	0146								
DATA	001F	0066	0067								
DATAP	001E	0064	0065	0141	0151	0155	0406	0407	0417	0423	
DCODE	0029	0074									
DDRD	000B	0086	0134								
DLEN	0022	0069	0151	0177	0178	0184	0194	0196	0236	0391	
DLPCNT	001E	0065	0189	0193	0197	0198	0200	0221	0225	0229	0230
			0232	0295	0309	0311	0318	0319	0343	0348	0350
			0354	0355							
DLTIME	0000	0096									
DQUAL	0027	0072									
DRIVE	0006	0084	0133	0166	0168	0234	0257	0356			
DROP	0001	0099	0307	0335	0448	0453					
ENABLE	0000	0097	0137								
EOT	0002	0098	0167	0258							
EOTR	0131'	0258	0258								
EOTW	005A'	0167	0167								
GTPAB2	0025'	0146	0144								
HSK	0001	0100	0305	0333	0446	0451					
HSKERR	0264'	0391	0249	0315	0443						
HSKSET	0000	0101	0153	0157	0180	0213	0308	0336	0400	0416	0433
			0449								
I123C	00AA	0109	0136								
I23C	00A8	0112									
I23CS	00BC	0113	0264								
I2C	0088	0110									
I2CS	008C	0111	0165	0256	0294						
I3C	00A0	0114	0366								
I3CS	00B0	0115	0280								
I3FLAG	00A0	0116	0369								
INHIB	0004	0102									
INIT	0000'	0132	0465								
INITDR	0000	0103	0133								
INITWF	0001	0104	0134								
INPUT	0008	0105	0292	0371							

INT1	02D7'	0457	0464								
INT1V	0018	0061	0457								
INT2	02D9'	0458	0463								
INT2V	001A	0062	0162	0253	0260	0293	0458				
INT3	02DB'	0459	0462								
INT3V	001C	0063	0261	0279	0459						
INVBIT	0001	0108	0245								
IOCNTL	0000	0078	0136	0165	0256	0264	0280	0294	0366	0369	
IRQ	0001	0117	0210	0430	0436						
ISTART	0260'	0388	0279								
ISYNC2	0249'	0375	0260								
ISYNC3	025A'	0384	0261								
LSN	000F	0118	0214	0337	0434	0439					
MIDWT	017D'	0291	0292								
MT	0008	0119									
MTSN	000C	0120	0166	0257							
MTWE	000A	0121	0168								
NIBCON	0014	0058	0059	0172	0184	0186	0187	0218	0287	0288	0312
			0317	0323	0324	0325	0326	0340			
OK	0000	0122	0235								
PREPDL	0175'	0287									
PSCALE	0003	0081	0179	0263							
RBITD2	023B'	0369	0369								
RBITD3	0248'	0373	0371								
RBITDT	0235'	0366	0293								
RBSERR	01BD'	0315	0305	0333							
RCHKL	0216'	0353	0350								
RCHKS2	020D'	0350	0348								
RCHKS3	021C'	0355	0352								
RCHKSM	01FF'	0344	0344	0355							
RCVABN	02B1'	0436	0403	0436							
RCVNRM	02A2'	0430	0401	0430							
RCVPAB	026E'	0397	0142								
REGS	0013	0057	0058	0060	0061	0062	0063	0064	0066		
RELEAS	0001	0123	0211	0431	0437						
RETDL	002C'	0150	0237	0393							
RLAST	0232'	0363	0361								
RLSB	01C0'	0317	0311								
RNIBDL	019A'	0301	0301	0319							
RNIBDT	01DB'	0329	0329	0341							
RNIBL2	01B4'	0311	0309								
RNIBL3	01C6'	0319	0313								
RNIBL4	01C9'	0320	0320								
RNUM	0026	0071									
RNXHSK	0273'	0400	0409								
RNXPAB	0276'	0401	0398								
RSYNC	0152'	0269	0275								
RTSTAT	003A'	0154	0363								
RXERR	02BD'	0443									
SAB	0029	0067	0068	0069	0070	0071	0072	0073	0074	0141	
SETBIT	0001	0124	0372								
SHFTIM	016A'	0282	0284								
SOM	0011'	0139	0148	0158							
STACK	0003	0056	0057								
STACKL	0010	0054	0057								
START	0080	0125	0370								
START2	0190'	0298	0299								

LABEL	VALUE	DEFN	REFERENCES					
STARTW	018C'	0296	0298					
STATUS	0020	0068	0155	0235	0360	0362	0392	
SYNCTM	0000	0126						
TEST	0004	0083						
TIME	0002	0079	0163	0171	0254	0286		
TIMER	0003	0082	0135	0169	0259	0370	0384	0388
TIMOUT	00FF	0127	0392					
WAFER	000A	0085	0167	0242	0245	0258	0292	0371
WBITDT	0110'	0240	0162	0253				
WCHKSM	00E2'	0222	0222					
WCLSB	00F2'	0228	0225					
WINVRS	011A'	0245	0241					
WLAST	00FF'	0233	0233					
WLSB	00AA'	0196	0193					
WNIBD2	00DB'	0218						
WNIBDT	00BF'	0207	0207	0219				
WNXHSK	028D'	0416	0425					
WNXPAB	0290'	0417	0414					
WRTCS2	00FC'	0232	0230					
WRTCSM	00F7'	0230	0227					
WRTDL	00AF'	0198	0195					
WRTDL2	00B4'	0200	0198					
WRTDL3	00B7'	0201	0201					
WSYNC	0084'	0181	0181	0183				
WTSNDL	009A'	0190	0190	0200	0232			
XINPUT	0121'	0253	0145					
XMTABN	02CD'	0451	0422	0451				
XMTNRM	02C0'	0446	0419	0446				
XMTPAB	0288'	0413	0152	0156				
XPRINT	004A'	0162	0147					
ZERO	0000	0128	0135	0234	0356			