```
  SSSSSSSS    UU      UU    PPPPPPPP      EEEEEEEEEE    RRRRRRRR
  SSSSSSSS    UU      UU    PPPPPPPP      EEEEEEEEEE    RRRRRRRR
SS            UU      UU    PP      PP    EE            RR      RR
SS            UU      UU    PP      PP    EE            RR      RR
SS            UU      UU    PP      PP    EE            RR      RR
SS            UU      UU    PP      PP    EE            RR      RR
  SSSSSS      UU      UU    PPPPPPPP      EEEEEEEE      RRRRRRRR
  SSSSSS      UU      UU    PPPPPPPP      EEEEEEEE      RRRRRRRR
        SS    UU      UU    PP            EE            RR  RR
        SS    UU      UU    PP            EE            RR  RR
        SS    UU      UU    PP            EE            RR    RR          ....
        SS    UU      UU    PP            EE            RR    RR          ....
SSSSSSSS      UUUUUUUUUU    PP            EEEEEEEEEE    RR      RR        ....
SSSSSSSS      UUUUUUUUUU    PP            EEEEEEEEEE    RR      RR        ....
```

```
LL              SSSSSSSS    TTTTTTTTTT        ;;;;          11
LL              SSSSSSSS    TTTTTTTTTT        ;;;;          11
LL            SS                TT            ;;;;        1111
LL            SS                TT            ;;;;        1111
LL            SS                TT                          11
LL            SS                TT                          11
LL              SSSSSS          TT            ;;;;          11
LL              SSSSSS          TT            ;;;;          11
LL                    SS        TT            ;;;;          11
LL                    SS        TT            ;;;;          11
LL                    SS        TT              ;;          11
LL                    SS        TT              ;;          11
LLLLLLLLLL    SSSSSSSS          TT            ;;          111111
LLLLLLLLLL    SSSSSSSS          TT            ;;          111111
```

```
1   ********************************************************************************
2   ********************************************************************************
3   ********************************************************************************
4   ***                                                                          ***
5   ***                                                                          ***
6   ***                                                                          ***
7   ***            SUPERVISOR MODULE FOR  ISI  PRINTERS COMPATIBLE               ***
8   ***                    with  IBM  3274/6 PROTOCOL                            ***
9   ***                                                                          ***
10  ***                                                                          ***
11  ***                    INTERFACE SYSTEMS, INC.                               ***
12  ***                    ANN ARBOR, MICHIGAN  48103                            ***
13  ***                                                                          ***
14  ***                                                                          ***
15  ***                                                                          ***
16  ***            FOR MOTOROLA  6809  OR COMPATIBLE MICROPROCESSOR              ***
17  ***                    USING A GENERATION III INTERFACE                      ***
18  ***                                                                          ***
19  ***                                                                          ***
20  ***                                                                          ***
21  ********************************************************************************
22  ********************************************************************************
23  ********************************************************************************
24  *
25  *
26  *                BY:     INTERFACE SYSTEMS, INCORPORATED
27  *                        462 JACKSON PLAZA
28  *                        ANN ARBOR, MICHIGAN  48103
29  *
30  *                        RICHARD L. COLE
31  *
32  *                        COPYRIGHT 1982, 1983
33  *                                BY
34  *                        INTERFACE SYSTEMS, INC.
35  *
36  *                        ALL RIGHTS RESERVED
37  *
38  *
39  ********************************************************************************
40  ********************************************************************************
41  ***                                                                          ***
42  ***                                                                          ***
43  *       This module contains or defines the following:
44  *
45  *       1.      Interrupt trap vector
46  *       2.      Printer Output Area
47  *       3.      Control Unit Output Area
48  *       4.      Interrupt handlers for FIRQ, IRQ, NMI, RESET, SWI
49  *       5.      Timer Routines
50  *       6.      Front Panel Interface
51  *       7.      Idle Program
52  *       8.      Order Request (SOP) Processor
```

```
                                53   *       9.      Initialization for Interface Elements
                                54   *       10.     Miscellaneous routines:  ERROR
                                55   *       11.
                                56   *       12.
                                57   ***                                                                                  ***
                                58   ***                                                                                  ***
                                59   ************************************************************************************
                                60   ************************************************************************************
                                61   ***                                                                                  ***
                                62   ***                                                                                  ***
                                63   ***                                                                                  ***
                                64           NAM     SUPERVISOR
                                65   *
                                66           INCLUD  DK3:[140,2]RANGE.ACT
                                67   *
          =2000                 68   .RAMB.  EQU     $2000           target/target variable RAM
          =27FF                 69   .RAME.  EQU     $27FF
          =C000                 70   .ROMB.  EQU     $C000           target program base address
          =FFFF                 71   .ROME.  EQU     $FFFF           target program limit
                                72   *
                                73   *
     =FFF0 =FFFF                74           SECT    .VEC.,REL,RANGE=.ROME.-$F:.ROME.        trap vector section
     =2000 =27FF                75           SECT    SUPRAM,REL,RANGE=.RAMB.:.RAME.  variable area
     =C000 =FFFF                76           SECT    SUPER,REL,RANGE=.ROMB.:.ROME.   program
     =0000 =2000                77           SECT    PRTBUF,REL,RANGE=0:.RAMB.       printer/control unit common
     =2700 =27FF                78           SECT    SYSTAK,REL,RANGE=.RAME.-$FF:.RAME.      primary stack area
                                79   ***                                                                                  ***
                                80   ***                                                                                  ***
                                82   ***                                                                                  ***
                                83   ************************************************************************************
                                84   ***                                                                                  ***
                                85   ***                                                                                  ***
```

```
                                87
          =0000                 88           RSECT   SYSTAK
          =0100                 89   STCKLN  EQU     $100            define stack length
0000'     =0100                 90   STACK   RMB     STCKLN          reserve space for stack
          =0100'                91   STKTOP  EQU     STACK+STCKLN    define initial SP value
```

```
                        93          SUBTTL  EQUATES
                        94  *
                        95  ********************************************************************************
                        96  ***                                                                          ***
                        97  ***                                                                          ***
                        98  *       HARDWARE DICTATED EQUATES
                        99  *
                       100  *               The following addresses, masks and constants are
                       101  *               related to the hardware on which this program is
                       102  *               to be executed.  That hardware is currently defined
                       103  *               to be an ISI 736 letter quality printer, using a
                       104  *               model 1193 interface.
                       105  *
=8000                  106  VIA     EQU     $8000           base address of  6522 "VIA"
=8000                  107  VIAORB  EQU     VIA+0               data output register "B"
=8001                  108  VIAORA  EQU     VIA+1               data output register "A"
=8000                  109  VIAIRB  EQU     VIA+0               data input register "B"
=8001                  110  VIAIRA  EQU     VIA+1               data input register "A"
=8002                  111  VIADDB  EQU     VIA+2               data direction control reg. "B"
=8003                  112  VIADDA  EQU     VIA+3               data direction control reg. "A"
=8004                  113  T1L.La  EQU     VIA+4               timer 1 data latch, lsb, write
=8004                  114  T1C.L   EQU     VIA+4               timer 1 counter. lsb, read, clears flag
=8005                  115  T1C.H   EQU     VIA+5               timer 1 counter, msb, w causes ll to lc xfer
=8006                  116  T1L.Lb  EQU     VIA+6               timer 1 data latch, lsb
=8007                  117  T1L.H   EQU     VIA+7               timer 1 data latch, msb, w clears flag
=8008                  118  T2L.L   EQU     VIA+8               timer 2 data latch, lsb, write
=8008                  119  T2C.L   EQU     VIA+8               timer 2 counter, lsb, read
=8009                  120  T2C.H   EQU     VIA+9               timer 2 counter, msb
=800B                  121  VIAACR  EQU     VIA+11              auxiliary control register
=800C                  122  VIAPCR  EQU     VIA+12              peripheral control register
=800D                  123  VIAIFR  EQU     VIA+13              interrupt flag register
=800E                  124  VIAIER  EQU     VIA+14              interrupt enable register
=800F                  125  VIAOAN  EQU     VIA+15              output data register "A", no handshake
=BFFF                  126  VIADMK  EQU     $BFFF               direction control const.  write to VIADDB
=7017                  127  VIATCN  EQU     (6000!M256)*256+6000/256        6ms timer const. (ref. TICK)
=0040                  128  VIATC1  EQU     $40                 constant selects t1, free-run mode
=00C0                  129  VIATC2  EQU     $C0                 write to IER to enable T1 interrupt
=0040                  130  VIATC3  EQU     $40                 write to IER to disable T1 interrupt
=0082                  131  VIAPC1  EQU     $82                 write to IER to enable printer interrupt
=0002                  132  VIAPC2  EQU     $2                  write to IER to disable printer interrupt
=00CA                  133  VIAPC3  EQU     $CA                 initial value for PCR
                       134  *
=5000                  135  SWSEL   EQU     $5000           switch group select register
=3000                  136  SWBUS   EQU     $3000           selected front panel switches
=0001                  137  SWSEL1  EQU     1               selects test selector and 4 control switches
=00F0                  138  SWMSK1  EQU     $F0             mask for group 1 rotary test switch
=0002                  139  SWSEL2  EQU     2               selects more control switches
=0000                  140  SWMSK2  EQU     0               mask for DIP switches 5, 6 & 7
=0004                  141  SWSEL3  EQU     4               selects rocker switches
=00FF                  142  SWMSK3  EQU     $FF             mask for rocker switches
=0008                  143  SWSEL4  EQU     8               selects form length thumbwheel switches
=00FF                  144  SWMSK4  EQU     $FF             mask for form length switches
```

```
                                145   *
=0005                           146   HOLDSV  EQU     5               HOLD/ENABLE switch code
                                147   *
=3800                           148   SWDIP   EQU     $3800           "E48"  on-card DIP switches
=0000                           149   SWMSK0  EQU     0               mask for E48 switches
=00F8                           150   LNGMSK  EQU     $F8             5 bits select a "language"
=0007                           151   MODLMK  EQU     $7              3 bits specify the "model"
                                152   *
=2800                           153   STATUS  EQU     $2800           printer and Top Board status
=00C0                           154   TBSMSK  EQU     $C0             Top Board "command" mask
                                155   *
=4000                           156   LEDBUS  EQU     $4000           front panel LEDs and beeper
=0080                           157   BEEPMK  EQU     $80             beeper bit
=0080                           158   ALRMSK  EQU     $80             alarm complementing mask
=007F                           159   LEDMSK  EQU     $7F             LED complementing mask
=0040                           160   LED3    EQU     $40             LED 3 bit
=0020                           161   LED2    EQU     $20             LED 2 bit
=0010                           162   LED1    EQU     $10             LED 1 bit
=0004                           163   LEDHLD  EQU     4               HOLD indicator LED
=0002                           164   LEDCHK  EQU     2               CHECK indicator LED
=0001                           165   LEDRDY  EQU     1               READY indicator LED
                                166   *
=0040                           167   FRQMSK  EQU     $40             CC FIRQ mask bit
=6000                           168   FIRQFL  EQU     $6000           FIRQ (top board) flag, clear on write
0                               169                                   also generates int ack to T.B.
                                170   *
=7000                           171   TBREG   EQU     $7000           Top Board communication register (write only)
=0001                           172   TBRSTM  EQU     1               Top board RESET bit mask
=0000                           173   TBRSTB  EQU     0               Top board RESET bit bit
=0002                           174   TBDISM  EQU     2               Top board DISABLE bit mask
=0002                           175   TBDISB  EQU     2               Top board DISABLE bit bit
=0008                           176   TBBBFM  EQU     8               Top board BUFFER SIZE indicator bit mask
=0008                           177   TBBBFB  EQU     8               Top board BUFFER SIZE indicator bit bit
                                178   *
                                179   ********************************************************************************
                                180   ********************************************************************************
                                181   ********************************************************************************
                                182   *
                                183   *       SOFTWARE DETERMINED EQUATES
                                184   *
=0008                           185   MXTIMR  EQU     8               up to 8 timers allowed
=FFFF                           186   TRUE.   EQU     -1              boolean "TRUE"
=0000                           187   FALSE.  EQU     0               boolean "FALSE"
=0010                           188   ECBIT   EQU     16              IBM printer status "equipment check" bit
=0008                           189   IRBIT   EQU     8                "     "      "    "intervention required" bit
=0020                           190   OCBIT   EQU     32               "     "      "    "order complete" bit
=0004                           191   SDABIT  EQU     4                "     "      "    "sense data available" bit
=0001                           192   ABOCOD  EQU     1               IBM cammand definition   --  abort
=0002                           193   SSACOD  EQU     2                "     "          "       --  Set status avail
=0003                           194   PRICOD  EQU     3                "     "          "       --  Print
=0004                           195   LPSCOD  EQU     4                "     "          "       --  Load Prog. Symbols
=0006                           196   TICK    EQU     6               clock interval in ms (see: VIATCN above)
```

```
=2710                197   TICKM   EQU     10000           number of intervals for 1 minute
=0004                198   RDYLED  EQU     4               select code for 'ready' LED
                     199   *
                     200   ********************************************************************************
                     201   ********************************************************************************
                     202   ********************************************************************************
                     203   *
                     204   FILE    MACRO   TITL,FILNAM
                     205           PAGE
                     206           SUBTTL  TITL    (FILNAM)
                     207           NLIST
                     208           INCLUD  DK3:[140,2]FILNAM
                     209           SPACE   5
                     210           ENDM



                     212+          FILE    MACROS,<SUPERMACS.SRC>
```

```
                        SUBTTL  MACROS    (SUPERMACS.SRC)
                *
                *       THE FOLOWING ARE MACRO DEFINITIONS FOR USE BY THE SUPERVISOR
                *
                *
                *               CLEAR CLOCK FLAG   (6522 VIA)
                *
                *               Use is:
                *                       CCF
                *
                *
CCF             MACRO
                TST     TIC.L           clears the clock flag
                ENDM
                *
                *
                *
                *               INTERRUPT OFF
                *               INTERRUPT ON
                *
                *               Use is:
                *                       IOF     <F,I,B> for FIRQ, IRQ, BOTH
                *                       ION     <F,I,B> for FIRQ, IRQ, BOTH
                *                               B is equivalent to F,I
                *
.IOXAR          MACRO   *ARGS
.MSK            SET     0               initial value of mask variable
                IFNB    <*ARGS>         do no more if no arguments
                IRP     *ARG,<*ARGS>
                IFIDN   <*ARG>,<B>      check for a "B"
.MSK            SET     .MSK!+$50       F,I set
                ENDIF
                IFIDN   <*ARG>,<I>      check for an "I"
.MSK            SET     .MSK!+$10       I set
                ENDIF
                IFIDN   <*ARG>,<F>      check for an "F"
.MSK            SET     .MSK!+$40       F set
                ENDIF
                ENDR
                ENDIF
                ENDM
                *
                *
                *
IOF             MACRO   *ARGS
                NLIST
                .IOXAR  <*ARGS>         form required constant
                LIST
                ORCC    #.MSK           turn *ARGS interrupts off
                ENDM
                *
                *
```

```
                    275A  ION     MACRO   *ARGS
                    276A          NLIST
                    277A          .IOXAR  <*ARGS>         form required constant
                    278A          LIST
                    279A          ANDCC   #!N.MSK         turn *ARGS interupts on
                    280A          ENDM
                    281A  *
                    282A          INCLUD  DK3:[140,2]CALL.MAC
                    283A  *       'CALL' MACRO DEFINED
                    342A  *



                    344+          FILE    BUFFER,<PRICRUBUF.SRC>
```

```
                        346A          SUBTTL  BUFFER    (PRICRUBUF.SRC)
                        350A  *
                        351A  ********************************************************************************
                        352A  ********************************************************************************
                        353A  *
                        354A  *
                        355A  *       the following defines the Printer / Control Unit communication region
                        356A  *               aka the print buffer
                        357A  *
                        358A  *       it is in 3 parts:  the printer output area (information from the
                        359A  *               printer to the control unit),  the control unit output area
                        360A  *               (information to the printer from the control unit), and
                        361A  *               the print buffer proper (which contains the actual data to
                        362A  *               be printed or operated on).
                        363A  *
                        364A  ********************************************************************************
                        365A  ********************************************************************************
                        366A  *
     =0000              367A          RSECT   PRTBUF
                        368A  *
                        369A          INTERN  PSTATS
                        370A          INTERN  PSWSTA
                        371A          INTERN  PKEYIN
                        372A          INTERN  PSENSE
                        373A          INTERN  PIDENT
                        374A          INTERN  PMODE
                        375A          INTERN  PMSA
                        376A          INTERN  PMSL
                        377A          INTERN  PORDER
                        378A          INTERN  PPARS
                        379A          INTERN  PMPP
                        380A          INTERN  PTSTM
                        381A          INTERN  PBUFR
                        382A          INTERN  PEACT
                        383A          INTERN  PEAB
                        384A  *
                        385A  ********************************************************************************
                        386A  *
0000'    =0001          387A  PSTATS  RMB     1               printer status byte (sec 2.2.1)
0001'    =0001          388A  PSWSTA  RMB     1               swwitch status, 6/8, s/d, etc. (sec 2.2.2)
0002'    =0001          389A  PKEYIN  RMB     1               key input code, PA, (sec 2.2.3)
0003'    =0001          390A  PSENSE  RMB     1               sense data code, Can, DR, etc.  (sec 2.2.4)
         =0007          391A          RMB     7               (reserved)
0008'    =0005          392A  PIDENT  RMB     5               printer config. code (sec 2.2.5)
                        393A  *
0010'    =0002          394A  PMODE   RMB     2               mode bytes  (sec 2.3.1)
0012'    =0002          395A  PMSA    RMB     2               message starting address (sec 2.3.2)
0014'    =0002          396A  PMSL    RMB     2               message length  (sec 2.3.2)
0016'    =0001          397A  ORDER   RMB     1               order  (sec 2.3.3)
0017'    =0001          398A  PPARS   RMB     1               order parameters  (sec 2.3.3)
0018'    =0001          399A  PMPP    RMB     1               maximum presentation position  (sec 2.3.5)
         =0031          400A          RMB     49              (reserved)
```

```
     004A'  =0004                401A  PTSTM   RMB     4               test message  (sec 1.4.4, RESET)
                                 402A  *
            =0002                403A          RMB     2               (reserved)
     0050'  =0FB0                404A  PBUFR   RMB     4016            message buffer
                                 405A  *
            =0010                406A          RMB     16              (reserved-EAB printer output counterpart)
     1010'  =0008                407A  PEACT   RMB     8               extended attribute correlation table (sec 2.3.4)
                                 408A  *
            =0038                409A          RMB     56              (reserved)
     1050'  =0FB0                410A  PEAB    RMB     4016            extended attribute buffer
                                 411A  *
                                 412A  ********************************************************************************
                                 413A  ********************************************************************************



                                 419+          FILE    VECTOR,<SUPERVEC.SRC>
```

```
                                421A            SUBTTL  VECTOR    (SUPERVEC.SRC)
                                425A    *
                                426A    ********************************************************************************
                                427A    *
                                428A    *       DEFINE THE INTERRUPT VECTOR ADDRESSES AT FFF0
                                429A    *
                                430A    ********************************************************************************
                                431A    *
            =0000               432A            SECT    .VEC.
                                433A    *
      0000'   0000'             434A    VECTOR  FDB     CSTART          reserved vector (debugging coldstart pointer)
      0002'   05A3'             435A            FDB     ERROR           SWI3 . . . . .  (may be alternately defined)
      0004'   05A8'             436A            FDB     ERROR           SWI2 . . . . .  (may be alternately defined)
      0006'   0281'             437A            FDB     TBINT           FIRQ . . . . .
      0008'   023E'             438A            FDB     IRQINT          IRQ  . . . . .
      000A'   0321'             439A            FDB     SFTINT          SWI  . . . . .  (disable request)
      000C'   05A8'             440A            FDB     ERROR           NMI  . . . . .
      000E'   0000'             441A            FDB     CSTART          reset vector  (cold start)
                                442A    *



                                444+            FILE    COLDSTART,<SUPERINIT.736>
```

```
                        446A          SUBTTL  COLDSTART    (SUPERINIT.736)
                        450A  *
                        451A  ********************************************************************************
                        452A  ********************************************************************************
                        453A  *
                        454A  *       C O L D S T A R T       --       "G R E Y   B O X"   (ISI  1261)
                        455A  *
                        456A  *               Routine takes care of the required initial set-up
                        457A  *                  after Power On
                        458A  *
                        459A  *               Routine includes (or calls) code to do the following
                        460A  *               1.   Memory test
                        461A  *               2.   Initial read of switches
                        462A  *               3.   Set up of printer
                        463A  *               4.   Set up of clock
                        464A  *               5.   Init. of print module
                        465A  *               6.   Init. of "SLU"
                        466A  *               7.   Init. of Enable/Disable control
                        467A  *               8.   Init. of System Stack
                        468A  *               9.   ...
                        469A  *
                        470A  ********************************************************************************
                        471A  ********************************************************************************
                        472A  *
                        473A  *       Initialize Equates
                        474A  *
=2000                   475A  MEMBAS  EQU     .RAMB.          beginning of stack/variable RAM
=0800                   476A  MEMLEN  EQU     .RAME.-.RAMB.+1 length of RAM
                        477A  *
                        478A  ********************************************************************************
                        479A  ********************************************************************************
                        480A  *
                        481A          INTERN  CSTART
                        482A  *
=0000                   483A          RSECT   SUPER           this is part of supervisor
                        484A  *
                        485A  *
0000'  12               486A  CSTART  NOP
0001'  10CE 0100'       487A          LDS     #STKTOP         set up the stack
                        488A  *
                        489A  *       Memory tests for:
                        490A  *
                        491A  *       2k buffer       0-7FF, Scratch RAM
                        492A  *       4k buffer       0-FFF, Scratch RAM
                        493A  *       2k w/EAE        0-7FF, 1000-17FF, Scratch RAM
                        494A  *       4k w/EAE        0-7FF, 800-FFF, 1000-17FF, 1800-1FFFF, Scratch RAM
                        495A  *
                        496A  *       front panel LEDs initialize to indicators off and display showing 000
                        497A  *
                        498A  *       begin RAM memory check
                        499A  *
0005'  8E 2000          500A          LDX     #MEMBAS         set base address of stack/variable RAM
```

```
0008' CC 0000          501A          LDD   #0                to clear and test memory
000B' A7 84            502A  1$      STA   ,X                clear
000D' 6D 80            503A          TST   ,X+               test and advance pointer
000F' 1026 0087        504A          LBNE  MEMERR            branch if it did not read back zero
0013' 8C 2800          505A          CMPX  #MEMBAS+MEMLEN    test for done with clear
0016' 25 F3            506A          BLO   1$
                       507A  *
                       508A  *       wiped:  now try other data
                       509A  *
0018' 8E 2000          510A          LDX   #MEMBAS
001B' CC 00FF          511A  2$      LDD   #$FF
001E' 6D 84            512A          TST   ,X                is cell still zero?
0020' 26 78            513A          BNE   MEMERR
0022' E7 84            514A          STB   ,X                store all ones
0024' E1 84            515A          CMPB  ,X                check it
0026' 26 72            516A          BNE   MEMERR
0028' C6 55            517A          LDB   #$55              try half ones
002A' E7 84            518A          STB   ,X
002C' E1 84            519A          CMPB  ,X                check it
002E' 26 6A            520A          BNE   MEMERR
0030' C6 AA            521A          LDB   #$AA              try other half ones
0032' E7 84            522A          STB   ,X
0034' E1 84            523A          CMPB  ,X                check it
0036' 26 62            524A          BNE   MEMERR
0038' A7 84            525A          STA   ,X                re-zero cell
003A' A1 80            526A          CMPA  ,X+               check it and point to next cell
003C' 26 5C            527A          BNE   MEMERR
003E' 8C 2800          528A          CMPX  #MEMBAS+MEMLEN    done?
0041' 25 D8            529A          BLO   2$
                       530A  *
                       531A  *       stack and variable RAM is checked and zeroed
                       532A  *               stack pointer can now be loaded
                       533A  *
0043' 8E 0000'         534A  STKSET  LDX   #PSTATS            test base print buffer
0046' BD 0058'         535A          JSR   MEMCHK
0049' B6 3800          536A          LDA   SWDIP             check buffer size
004C' 84 07            537A          ANDA  #MODLMK
004E' 81 03            538A          CMPA  #3                models 1&2 have 2k buffers
0050' 25 06            539A          BLO   1$
                       540A  *
0052' 8E 0800          541A          LDX   #$800             big buffer, check it
0055' BD 0058'         542A          JSR   MEMCHK
                       543A  *
                       544A  *       if there is EAB (or a way to determine that there might be one)
                       545A  *               insert MEMCHKs for EAB here
                       546A  *
0058' 12               547A  1$      NOP
0059' 20 44            548A          BRA   SYINIT            go initialize the program
                       549A  *
                       550A  *       routine MEMCHK.  enter with base address of 800 word test block in X
                       551A  *
005B' 1F 13            552A  MEMCHK  TFR   X,U               save base address
```

```
005D'  31 89 0800          553A          LEAY   $800,X        calculate end of block (+1) and save in stack
0061'  10AF E3             554A          STY    ,--S
0064'  4F                  555A          CLRA                 first zero memory bank
0065'  A7 84               556A  1$      STA    ,X
0067'  6D 80               557A          TST    ,X+
0069'  26 2F               558A          BNE    MEMERR
006B'  AC E4               559A          CMPX   ,S            done?
006D'  25 F6               560A          BLO    1$
                           561A  *
006F'  1F 31               562A          TFR    U,X           reload X
0071'  6D 84               563A  2$      TST    ,X            is cell still zero?
0073'  26 25               564A          BNE    MEMERR
0075'  D6 FF               565A          LDB    $FF           store all ones and check
0077'  E7 84               566A          STB    ,X
0079'  E1 84               567A          CMPB   ,X
007B'  26 1D               568A          BNE    MEMERR
007D'  C6 55               569A          LDB    #$55          try constant 55
007F'  E7 84               570A          STB    ,X
0081'  E1 84               571A          CMPB   ,X
0083'  26 15               572A          BNE    MEMERR
0085'  C6 AA               573A          LDB    #$AA          store & check w/AA
0087'  E7 84               574A          STB    ,X
0089'  E1 84               575A          CMPB   ,X
008B'  26 0D               576A          BNE    MEMERR
008D'  A7 84               577A          STA    ,X            re-zero
008F'  A1 80               578A          CMPA   ,X+           check it & point to next
0091'  26 07               579A          BNE    MEMERR
0093'  AC E4               580A          CMPX   ,S            check for done
0095'  25 DA               581A          BLO    2$
0097'  32 62               582A          LEAS   2,S           give back stack space
0099'  39                  583A          RTS
                           584A  *
                           585A  *       memory error on power_up tests
                           586A  *
009A'  86 03               587A  MEMERR  LDA    #3            LED3 (in addition to check)
009C'  7E 05A8'            588A          JMP    ERROR
                           589A  *
                           590A  ********************************************************************************
                           591A  *
                           592A  *       initialize various sections
                           593A  *
009F'  BD 03AD'            594A  SYINIT  JSR    SWSET         switch setup
00A2'  BD 0582'            595A          JSR    LEDINI        initialize LED display
00A5'  BD 0543'            596A          JSR    CLKINI        set up clock and timers
                           597A+         CALL   SSAPOR        initialize "secondary logical unit"
                           602B          EXTERN SSAPOR
00A8'  BD 0001*            606B          JSR    SSAPOR        (GO TO SSAPOR)
                           626A+         CALL   PSPOR         initialize programmed symbols
                           631B          EXTERN PSPOR
00AB'  BD 0002*            635B          JSR    PSPOR         (GO TO PSPOR)
                           655A+         ION    <I>           device POR requires timers active
00AE'  1C EF               682B          ANDCC  #!N.MSK       turn I interupts on
```

```
                                683A+          CALL    PRIPOR          initialize print module
                                688B           EXTERN  PRIPOR
00B0'  BD 0003*                 692B           JSR     PRIPOR          (GO TO PRIPOR)
                                712A+          IOF     <I>
00B3'  1A 10                    739B           ORCC    #.MSK           turn I interrupts off
00B5'  7F 0023'                 740A           CLR     SWFLAG          flags may have been set in PRIPOR
00B8'  7F 000F'                 741A           CLR     HLDREQ
00BB'  BD 00CC'                 742A           JSR     PRGINI          set up "goodies" in the program
                                743A  *
00BE'  86 04                    744A           LDA     #4              set "READY" LED
00C0'  C6 FF                    745A           LDB     #TRUE.
00C2'  BD 055F'                 746A           JSR     LEDSET
                                747A  *
00C5'  8E 00ED'                 748A           LDX     #BEGIN          set up to go--address
00C8'  4F                       749A           CLRA                        --both interrupts will be enabled
00C9'  34 12                    750A           PSHS    A,X
00CB'  3B                       751A           RTI                     "return" to BEGIN
                                752A  *
                                753A  ************************************************************************************
                                754A  *
                                755A  *       superviser variable initialize
                                756A  *
00CC'  86 FF                    757A  PRGINI   LDA     #TRUE.          set system to "enabled"
00CE'  B7 0006'                 758A           STA     ENAFLG
00D1'  7F 0010'                 759A           CLR     DISINH          clear "disabling" semiphore
00D4'  7F 0003'                 760A           CLR     DEVTOF          clear device (fault) timeout flag
00D7'  C6 01                    761A           LDB     #(!N(TBBBFB)!.(TBBBFM))!+(!N(TBRSTB)!.(TBRSTM))         set bits
                                762A  *                                ... for no reset & no big buffer
00D9'  B6 001A'                 763A           LDA     SWBYTO          set up buffer size in T.B. control byte
00DC'  84 07                    764A           ANDA    #MODLMK         mask "A" to model bits
00DE'  81 03                    765A           CMPA    #3
00E0'  25 02                    766A           BLO     1$
00E2'  C8 08                    767A           EORB    #TBBBFB         big buffer, set bit to indicate
00E4'  1F 98                    768A  1$       TFR     B,A             copy image to A
00E6'  FD 000D'                 769A           STD     TBRIMG          store result value in TB Reg image
                                770A  *
00E9'  B7 7000                  771A           STA     TBREG           enable T.B.
                                772A  *
00EC'  39                       773A           RTS
                                774A  *
                                775A  ************************************************************************************
                                776A  ************************************************************************************


                                778+           FILE    .MAIN.,<SUPERMAIN.SRC>
```

```
                                780A            SUBTTL  .MAIN.  (SUPERMAIN.SRC)
                                784A    *
                                785A    ********************************************************************************
                                786A    ********************************************************************************
                                787A    *
                                788A    *
                                789A            EXTERN  HTOFLG          HOLD PRINT timeout flag
                                790A            EXTERN  FLTFLG          fault time out flag
                                791A            EXTERN  FLTREQ
                                792A            EXTERN  FLTTIM          fault timeout control
                                793A    *
                                794A    *       main program RAM
                                795A    *
          =0000                 796A            RSECT   SUPRAM
                                797A    *
                                798A            INTERN  ABOREQ
                                799A            INTERN  ABOFLG
                                800A            INTERN  BUSY
                                801A            INTERN  ENAFLG
                                802A            INTERN  EXSTAT
                                803A            INTERN  RSTFLG
                                804A    *
0000'     =0001                 805A    ABOREQ  RMB     1               abort request pending
0001'     =0001                 806A    ABOFLG  RMB     1               abort request active
0002'     =0001                 807A    BUSY    RMB     1               BUSY indicator
0003'     =0001                 808A    DEVTOF  RMB     1               device fault time/out state flag
0004'     =0002                 809A    DEVTR   RMB     2               fault timeout -- time remaining
0006'     =0001                 810A    ENAFLG  RMB     1               enabled (T.) / disabled (F.) flag
0007'     =0001                 811A    EXSTAT  RMB     1               special status byte
0008'     =0002                 812A    PORDER  RMB     2               current (last) order w/parameter byte
000A'     =0001                 813A    RSTREQ  RMB     1               RESET command received
000B'     =0001                 814A    RSTFLG  RMB     1               RESET flag recognozed
000C'     =0001                 815A    SOPREQ  RMB     1               start op received
000D'     =0002                 816A    TBRIMG  RMB     2               T. B. control register image
                                817A    *
                                818A    ********************************************************************************
                                819A    ********************************************************************************
                                820A    *
          =00ED                 821A            RSECT   SUPER
                                822A    *
                                823A    *       commence with MAIN program
                                824A    *
                                825A    *               begin by disabling  --  no status
                                826A    *
00ED'     86 01                 827A    BEGIN   LDA     #1              "disable" function code
00EF'     8E 00F8'              828A            LDX     #2$
00F2'     3F                    829A    1$      SWI                     execute
00F3'     5D                    830A            TSTB
00F4'     26 FC                 831A            BNE     1$              this one shouldn't fail, but code it anyway
00F6'     20 02                 832A            BRA     IDLE
                                833A    *
00F8'     0000                  834A    2$      FDB     0               status control  --  no bits set or cleared
```

```
                        835A  *
                        836A  ********************************************************************************
                        837A  *
                        838A  *       IDLE  --  look for job to do
                        839A  *
00FA' B6 000B'          840A  IDLE    LDA     RSTFLG          look for RESET command
00FD' 1026 001C         841A          LBNE    RSTCOM
                        842A
0101' B6 000C'          843A          LDA     SOPREQ          check for receipt of a SOP command
0104' 1026 0056         844A          LBNE    SOPCOM
                        845A
0108' B6 000F'          846A          LDA     HLDREQ          test for HOLD PRINT request
010B' 1026 0111         847A          LBNE    HOLDST
                        848A  *
                        849A  *       no work to do, check up on device
                        850A  *
                        851A+         CALL    DEVSTA          is device ready
                        856B          EXTERN  DEVSTA
010F' BD 0008*          860B          JSR     DEVSTA          (GO TO DEVSTA)
0112' 7D 0005*          880A          TST     FLTFLG          is fault timeout present?
0115' 27 03             881A          BEQ     DCXIT           if no fault timeout, return to idle check
                        882A  *
0117' BD 022E'          883A          JSR     HOLDTO          fault timeout, force IR condition
011A' 7E 00FA'          884A  DCXIT   JMP     IDLE
                        885A  *
                        886A  ********************************************************************************
                        887A  ********************************************************************************
                        888A  *
                        889A  *       process a RESET command while idle
                        890A  *
011D' 8E 0002'          891A  RSTCOM  LDX     #PSTATS+2       clear printer & control unit areas
0120' 4F                892A          CLRA
0121' A7 80             893A  1$      STA     ,X+
0123' 8C 0051'          894A          CMPX    #PBUFR+1
0126' 2F F9             895A          BLE     1$              loop if not clear yet
                        896A  *
                        897A  *
                        898A+         CALL    DEVRST
                        903B          EXTERN  DEVRST
0128' BD 0009*          907B          JSR     DEVRST          (GO TO DEVRST)
                        927A+         CALL    PRIRST          reset   print module & device
                        932B          EXTERN  PRIRST
0128' BD 000A*          936B          JSR     PRIRST          (GO TO PRIRST)
                        956A+         CALL    SSAPOR            "     SLU processor
012E' BD 0001*          965B          JSR     SSAPOR          (GO TO SSAPOR)
                        985A+         CALL    PSPOR             "     Programmed Symbol processor
0131' BD 0002*          994B          JSR     PSPOR           (GO TO PSPOR)
                       1014A  *
0134' 86 00            1015A          LDA     #FALSE.         set a bunch of things to false
0136' B7 000B'         1016A          STA     RSTFLG
0139' B7 000A'         1017A          STA     RSTREQ
013C' B7 0002'         1018A          STA     BUSY
```

```
013F'  B7 000C'          1019A          STA     SOPREQ
0142'  B7 0000'          1020A          STA     ABOREQ
0145'  B7 0001'          1021A          STA     ABOFLG
0148'  B6 0000'          1022A          LDA     PSTATS          clear status bits other than IR & ST
014B'  84 09             1023A          ANDA    #9
014D'  B7 0000'          1024A          STA     PSTATS
0150'  BD 00CC'          1025A          JSR     PRGINI
0153'  B6 0007'          1026A          LDA     EXSTAT          update "special" status
0156'  8A 10             1027A          ORA     #16             ...reset
0158'  B7 0007'          1028A          STA     EXSTAT
015B'  7E 00ED'          1029A          JMP     BEGIN           restart with things fixed up
                         1030A  *
                         1031A  *       start operation  --  decode and start it
                         1032A  *
015E'  B6 0000'          1033A  SOPCOM  LDA     ABOREQ          abort request in idle is a special case
0161'  26 5C             1034A          BNE     IDLABO
0163'  86 00             1035A          LDA     #FALSE.         zap SOPREQ
0165'  B7 000C'          1036A          STA     SOPREQ
0168'  B6 0008'          1037A          LDA     PORDER          validate and decode the order
016B'  1027 0096         1038A          LBEQ    INVAL
016F'  81 04             1039A          CMPA    #LPSCOD
0171'  102E 009C         1040A          LBGT    INVAL
0175'  27 14             1041A          BEQ     SOPLPS
0177'  81 02             1042A          CMPA    #SSACOD
0179'  26 25             1043A          BNE     SOPPRT
                         1044A  *
                         1045A  *       operation is SSA  --  do it
                         1046A  *
017B'  B6 0007'          1047A  SOPSSA  LDA     EXSTAT          update "special" status
017E'  8A 01             1048A          ORA     #1              ... SSA received
0180'  B7 0007'          1049A          STA     EXSTAT
                         1050A+         ION     <F>             re-enable interrupt
0183'  1C BF             1077B          ANDCC   #!N.MSK         turn F interupts on
                         1078A+         CALL    SSA             do SSA
                         1083B          EXTERN  SSA
0185'  BD 000B*          1087B          JSR     SSA             (GO TO SSA)
0188'  7E 00FA'          1107A          JMP     IDLE
                         1108A  *
                         1109A  *       operation is Load Programmed Symbols  --  do it
                         1110A  *
018B'  B6 0007'          1111A  SOPLPS  LDA     EXSTAT          update "special" status
018E'  8A 08             1112A          ORA     #8              ... LPS order received
0190'  B7 0007'          1113A          STA     EXSTAT
0193'  86 00             1114A          LDA     #FALSE.         LPS can be disabled
0195'  B7 0010'          1115A          STA     DISINH
                         1116A+         ION     <F>             possible PDI
0198'  1C BF             1143B          ANDCC   #!N.MSK         turn F interupts on
                         1144A+         CALL    LPS
                         1149B          EXTERN  LPS
019A'  BD 000C*          1153B          JSR     LPS             (GO TO LPS)
019D'  7E 01DD'          1173A          JMP     IDLE1
                         1174A  *
```

```
                        1175A  *       operation is Print  --  do it
                        1176A  *
01A0'  B6 0007'         1177A  SOPPRT  LDA     EXSTAT          update "special" status
01A3'  8A 04            1178A          ORA     #4              ... PRINT
01A5'  B7 0007'         1179A          STA     EXSTAT
01A8'  B6 0011'         1180A          LDA     PMODE+1         check for a MODE=0 command
01AB'  84 07            1181A          ANDA    #7              other bits may be defined
01AD'  27 CC            1182A          BEQ     SOPSSA
01AF'  86 00            1183A          LDA     #FALSE.         release disable inhibit
01B1'  B7 0010'         1184A          STA     DISINH
                        1185A+         ION     <F>             possible PDI
01B4'  1C BF            1212B          ANDCC   #!N.MSK         turn F interupts on
01B6'  7F 0003'         1213A          CLR     DEVTOF          device fault becomes print problem
                        1214A+         CALL    PRINTM
                        1219B          EXTERN  PRINTM
01B9'  BD 000D*         1223B          JSR     PRINTM          (GO TO PRINTM)
01BC'  7E 01DD'         1243A          JMP     IDLE1
                        1244A  *
                        1245A  *
                        1246A  *       ABORT received in idle, dismis it
                        1247A  *
                        1248A  *
01BF'  86 00            1249A  IDLABO  LDA     #FALSE.         clear some things
01C1'  B7 000C'         1250A          STA     SOPREQ
01C4'  B7 0002'         1251A          STA     BUSY
01C7'  B7 0000'         1252A          STA     ABOREQ
01CA'  B7 0001'         1253A          STA     ABOFLG
01CD'  B7 0010'         1254A          STA     DISINH
01D0'  B6 0007'         1255A          LDA     EXSTAT          update "special" status
01D3'  8A 02            1256A          ORA     #2              ... ABORT
01D5'  B7 0007'         1257A          STA     EXSTAT
                        1258A+         ION     <F>
01D8'  1C BF            1285B          ANDCC   #!N.MSK         turn F interupts on
01DA'  7E 00FA'         1286A          JMP     IDLE
                        1287A  *
                        1288A  *
                        1289A  *       check for ABORT HONORED by action routine
                        1290A  *
                        1291A  *
01DD'  34 01            1292A  IDLE1   PSHS    CC              save interrupt state
                        1293A+         IOF     <F>
01DF'  1A 40            1320B          ORCC    #.MSK           turn F interrupts off
01E1'  B6 0001'         1321A          LDA     ABOFLG          check for ABORTA flag up
01E4'  27 1A            1322A          BEQ     NOABRT          branch if no abort requested during action
01E6'  B6 0002'         1323A          LDA     BUSY            if abort was after action, BUSY will be set
01E9'  26 15            1324A          BNE     NOABRT          branch if abort received after action disable
                        1325A  *
                        1326A  *       handle ABORT received during action routine
                        1327A  *
01EB'  B6 0007'         1328A          LDA     EXSTAT          update "special" status
01EE'  8A 02            1329A          ORA     #2              ... ABORT
01F0'  B7 0007'         1330A          STA     EXSTAT
```

```
01F3'  86 00              1331A           LDA     #FALSE.         clear abort request
01F5'  B7 0001'           1332A           STA     ABOFLG
01F8'  35 01              1333A           PULS    CC              restore interrupt control
                          1334A+          CALL    SSAPOR          SLU needs to know about good aborts
01FA'  BD 0001*           1343B           JSR     SSAPOR          (GO TO SSAPOR)
01FD'  7E 00FA'           1363A           JMP     IDLE
                          1364A *
                          1365A *       no abort or received after action routine dismissed
                          1366A *
0200'  35 01              1367A NOABRT    PULS    CC              restore interrupt control
0202'  7E 00FA'           1368A           JMP     IDLE
                          1369A *
                          1370A *       specified "order" is invalid  --  done already
                          1371A *
0205'  86 01              1372A INVAL     LDA     #1              disable
0207'  8E 0211'           1373A           LDX     #INVALD
020A'  3F                 1374A           SWI
020B'  5D                 1375A           TSTB
020C'  26 F7              1376A           BNE     INVAL           retry if failed
020E'  7E 00FA'           1377A           JMP     IDLE
                          1378A *
                          1379A *       return status "order complete" & "order error"
                          1380A *
0211'  34                 1381A INVALD    FCB     OCBIT!+SDABIT!+ECBIT
0212'  00 00 00 00        1382A IDLETC    FCB     0,0,0,0
0216'  04                 1383A           FCB     4
                          1384A *
                          1385A *
                          1386A ********************************************************************************
                          1387A *
0217'  1388               1388A IDLELI    FDB     30000/TICK              30 sec in TICK ms intervals
                          1389A *
                          1390A *
                          1391A *       IDLE ENABLE TIMEOUT  --  disable, SA
                          1392A *
0219'  86 01              1393A IDLETO    LDA     #1              DISABLE function code
021B'  8E 0212'           1394A           LDX     #IDLETC
021E'  3F                 1395A           SWI
021F'  39                 1396A           RTS                     no retry required
                          1397A *
                          1398A *       NOTE: disable will fail if printer is disabled already, in which case
                          1399A *             this disable is superfluous
                          1400A *
                          1401A ********************************************************************************
                          1402A *
                          1403A *       HOLD PRINT request active  --  go to hold
                          1404A *
                          1405A+ HOLDST   CALL    HOLD            go do whatever required
                          1410B           EXTERN  HOLD
0220'  BD 000E*           1414B           JSR     HOLD            (GO TO HOLD)
0223'  B6 0004*           1434A           LDA     HTOFLG          check for HOLD PRINT timeout
0226'  27 03              1435A           BEQ     1$
```

```
0228'  BD 022E'          1436A          JSR     HOLDTO          HOLD time-out, set IR
022B'  7E 00FA'          1437A  1$      JMP     IDLE
                         1438A  *
                         1439A  *       HOLD PRINT timeout discovered in idle
                         1440A  *               also:  used by device fault timeout in idle
                         1441A  *
022E'  86 08             1442A  HOLDTO  LDA     #IRBIT          see if IR already set
0230'  B4 0000'          1443A          ANDA    PSTATS
0233'  26 06             1444A          BNE     1$
0235'  86 01             1445A          LDA     #1              DISABLE code
0237'  8E 023C'          1446A          LDX     #HTOCON
023A'  3F                1447A          SWI
023B'  39                1448A  1$      RTS
                         1449A  *
023C'  08 00             1450A  HTOCON  FCB     IRBIT,0         control to set IR
                         1451A  *
                         1452A  *
                         1453A  ********************************************************************************
                         1454A  ********************************************************************************
                         1455A


                         1457+          FILE    INTERRUPTS,<SUPERINTS.736>
```

```
                        1459A          SUBTTL  INTERRUPTS    (SUPERINTS.736)
                        1463A  *
                        1464A  ********************************************************************************
                        1465A  ********************************************************************************
                        1466A  ********************************************************************************
                        1467A  ***                                                                          ***
                        1468A  ***                                                                          ***
                        1469A  ***                                                                          ***
                        1470A  ***     I N T E R R U P T   P R O C E S S O R                                ***
                        1471A  ***                                                                          ***
                        1472A  ***              INTERRUPTS ARE:                                             ***
                        1473A  ***                      (1)     Top Board  --  FIRQ                         ***
                        1474A  ***                      (2)     Printer -- "DEVICE"  --  IRQ                ***
                        1475A  ***                      (3)     Timer  --  IRQ                              ***
                        1476A  ***                      (4)     Supervisor trap  --  SWI                     ***
                        1477A  ***                                                                          ***
                        1478A  ***                                                                          ***
                        1479A  ***                                                                          ***
                        1480A  ********************************************************************************
                        1481A  ********************************************************************************
                        1482A  *
                        1483A          EXTERN  HLDFLG
                        1484A          INTERN  HLDREQ
                        1485A          INTERN  LSTCOM
                        1486A          INTERN  LSTORD
                        1487A  *
                        1488A  ********************************************************************************
                        1489A  ********************************************************************************
                        1490A  ***                                                                          ***
                        1491A  ***                                                                          ***
                        1492A  ***     LOCAL INTERRUPTS  (IRQ)  --  TIMER(S), PRINTER INTERFACE, ETC.       ***
                        1493A  ***                                                                          ***
                        1494A  ***                                                                          ***
                        1495A  ********************************************************************************
                        1496A  *
        =000F           1497A          RSECT   SUPRAM
                        1498A  *
000F'   =0001           1499A  HLDREQ  RMB     1               HOLD request indicator
0010'   =0001           1500A  DISINH  RMB     1               disable (poll) inhibit
0011'   =0001           1501A  LSTATS  RMB     1               save of last disable status
0012'   =0002           1502A  LDADR   RMB     2               save of last disable SWI address
0014'   =0002           1503A  LDPTR   RMB     2               save of last disable vector address
0016'   =0001           1504A  LSTCOM  RMB     1               save of last coax command received
0017'   =0001           1505A  LSTORD  RMB     1               save of last order received
0018'   =0002           1506A  POINTR  RMB     2               pointer to last used interrupt routine
                        1507A  *


        =023E           1509A          RSECT   SUPER
```

```
                                1510A  *
023E'  B6 800D                  1511A  IRQINT  LDA     VIAIFR          identify interrupt source
0241'  B4 800E                  1512A          ANDA    VIAIER          isolate enabled interrupts
0244'  84 42                    1513A          ANDA    #(VIATC3)!+(VIAPC2)     from known flags
0246'  85 40                    1514A          BITA    #VIATC3         test for clock
0248'  26 09                    1515A          BNE     CLKTRP
024A'  85 02                    1516A          BITA    #VIAPC2         test for printer
024C'  26 01                    1517A          BNE     PTRTRP
024E'  3B                       1518A          RTI                     invalid interrupt
                                1519A  *
                                1520A  *       PRINTER TRAP
                                1521A  *
                                1522A+ PTRTRP  CALL    DEVIRQ
                                1527B          EXTERN  DEVIRQ
024F'  BD 0010*                 1531B          JSR     DEVIRQ          (GO TO DEVIRQ)
0252'  3B                       1551A          RTI
                                1552A  *
                                1553A  *       CLOCK TRAP
                                1554A  *
0253'  BD 04EC'                 1555A  CLKTRP  JSR     CLKII           go do clock related things
0256'  BD 0411'                 1556A          JSR     SWTTRP          update switch readings
                                1557A  *
                                1558A  *       now do clock initiated SUPERVISOR checks
                                1559A  *
                                1560A+         IOF     <F>
0259'  1A 40                    1587B          ORCC    #.MSK           turn F interrupts off
025B'  7D 0023'                 1588A          TST     SWFLAG          switch flag set?
025E'  27 14                    1589A          BEQ     200$
0260'  7D 000F*                 1590A          TST     HLDFLG          "HOLD" in effect?
0263'  26 0F                    1591A          BNE     200$            when HOLD is on, HOLD handles switches
0265'  7F 0023'                 1592A          CLR     SWFLAG          clear flag
0268'  B6 001D'                 1593A          LDA     SWBYT3          ONLY HOLD is legal
026B'  81 05                    1594A          CMPA    #HOLDSV
026D'  26 05                    1595A          BNE     200$
026F'  86 FF                    1596A          LDA     #TRUE.          set HOLD REQUEST
0271'  B7 000F'                 1597A          STA     HLDREQ
                                1598A  *
0274'  B6 000A'                 1599A  200$    LDA     RSTREQ          copy reset request to reset flag
0277'  B7 000B'                 1600A          STA     RSTFLG
027A'  BA 0000'                 1601A          ORA     ABOREQ          aboflg = rstflg .v. aboreq
027D'  B7 0001'                 1602A          STA     ABOFLG
                                1603A  *
028C'  3B                       1604A  255$    RTI
                                1605A  *
                                1606A  ********************************************************************************
                                1607A  ********************************************************************************
                                1608A  *
                                1609A  *       " T O P   B O A R D "  INTERRUPT
                                1610A  *
                                1611A  *               VIA     FIRQ
                                1612A  *
                                1613A  *       TOP BOARD causes FIRQ to be generated to signal a request for a
```

```
                        1614A  *       "bottom board" function
                        1615A  *
                        1616A  *       request is accompanied by a function code
                        1617A  *
                        1618A  *       function code is determined by interrogating "STATUS" masked with
                        1619A  *       "TBSMSK".  code is then rotated for use as an index.
                        1620A  *
                        1621A  *       codes are:
                        1622A  *
                        1623A  *               0  -  DISABLE
                        1624A  *               1  -  ENABLE
                        1625A  *               2  -  RESET
                        1626A  *               3  -  SOP  (Start OPeration)
                        1627A  *
                        1628A  *       for Start OP (code 3),
                        1629A  *               order to be performed is determined by reading controller
                        1630A  *               loaded byte at PORDER.  currently defined operations are:
                        1631A  *
                        1632A  *               1  -  ABORT
                        1633A  *               2  -  SSA
                        1634A  *               3  -  PRINT
                        1635A  *               4  -  Load Programmed Symbols
                        1636A  *               F0 -  Reset (may accompany RESET code)
                        1637A  *
                        1638A  ********************************************************************************
                        1639A  ********************************************************************************
                        1640A  *
0281'  34 16            1641A  TBINT   PSHS    A,B,X           these registers are used to service interrupt
                        1642A  *
0283'  F6 2800          1643A          LDB     STATUS          read command and compute command table offset
0286'  59               1644A          ROLB
0287'  59               1645A          ROLB
0288'  59               1646A          ROLB
0289'  59               1647A          ROLB
028A'  C4 06            1648A          ANDB    #TBSMSK!>5      mask to needed bits only
028C'  8E 029D'         1649A          LDX     #CMTBL
028F'  30 95            1650A          LEAX    [B,X]           compute address of required routine
0291'  BF 0018'         1651A          STX     POINTR          ... and save for diagnostics
0294'  54               1652A          LSRB                    save command for status print
0295'  F7 0016'         1653A          STB     LSTCOM
0298'  AD 84            1654A          JSR     ,X              go to required service subr
                        1655A  *
029A'  35 16            1656A          PULS    A,B,X           restore registers
029C'  3B               1657A          RTI
                        1658A  *
                        1659A  *       COMMAND TABLE
                        1660A  *
029D'  02A5'            1661A  CMTBL   FDB     CMDIS           to service DISABLE
029F'  02C7'            1662A          FDB     CMENB           "     "    ENABLE
02A1'  030B'            1663A          FDB     CMRES           "     "    RESET
02A3'  02E4'            1664A          FDB     CMSOP           "     "    SOP
                        1665A  *
```

```
                                1666A  ********************************************************************************
                                1667A  ********************************************************************************
                                1668A  *
                                1669A  *       DISABLE REQUEST RECEIVED  --  IF LEGAL, DISABLE
                                1670A  *
02A5'  7D 0010'                 1671A  CMDIS   TST     DISINH          backgrcund disable in progress?
02A8'  26 16                    1672A          BNE     1$
02AA'  86 00                    1673A          LDA     #FALSE.         set "disabled" internally
02AC'  B7 0006'                 1674A          STA     ENAFLG
02AF'  BD 04AA'                 1675A          JSR     CTIMER          clear IDLE ENABLE timer
02B2'  0219'                    1676A          FDB     IDLETO
02B4'  B7 6000                  1677A          STA     FIRQFL          clear FIRQ request  (A) not important
02B7'  FC 000D'                 1678A          LDD     TBRIMG          set up and
02BA'  8A 02                    1679A          ORA     #TBDISB
02BC'  FD 7000                  1680A          STD     TBREG           effect the disable
02BF'  39                       1681A          RTS
                                1682A  *
                                1683A  *       background disable or SOP in progress
                                1684A  *
02C0'  A6 66                    1685A  1$      LDA     6,S             indicate overlapped disable by disabling
02C2'  8A 40                    1686A          ORA     #FRQMSK         FIRQ in return CC
02C4'  A7 66                    1687A          STA     6,S
02C6'  39                       1688A          RTS
                                1639A  *
                                1690A  ********************************************************************************
                                1691A  ********************************************************************************
                                1692A  *
                                1693A  *       "SIMPLE" ENABLE
                                1694A  *
02C7'  7D 0002'                 1695A  CMENB   TST     BUSY            if not busy, start IDLE ENABLE timer
02CA'  26 07                    1696A          BNE     1$
                                1697A  *
02CC'  BD 0475'                 1698A          JSR     STIMER          not busy, set timer
02CF'  0219'                    1699A          FDB     IDLETO
02D1'  0217'                    1700A          FDB     IDLELI
                                1701A  *
02D3'                           1702A  1$
02D3'  C6 FF                    1703A  CMENB1  LDB     #TRUE.          set "enabled"
02D5'  F7 0006'                 1704A          STB     ENAFLG
02D8'  B7 6000                  1705A          STA     FIRQFL          clear FIRQ request
02DB'  B6 0000'                 1706A          LDA     PSTATS          clear status except IR, EC
02DE'  84 18                    1707A          ANDA    #(IRBIT)!+(ECBIT)
02E0'  B7 0000'                 1708A          STA     PSTATS
02E3'  39                       1709A          RTS
                                1710A  *
                                1711A  ********************************************************************************
                                1712A  ********************************************************************************
                                1713A  *
                                1714A  *       SOP  --  START OPERATION
                                1715A  *
02E4'  BD 02D3'                 1716A  CMSOP   JSR     CMENB1          do a "simple" enable [also sets B=TRUE.]
02E7'  B6 0016'                 1717A          LDA     ORDER
```

```
02EA'  B7 0017'          1718A          STA     LSTORD          save order for status print
02ED'  81 01             1719A          CMPA    #ABOCOD         check for "ABORT" order
02EF'  27 08             1720A          BEQ     3$
                         1721A  *
                         1722A  *       not an abort order  --  copy to background save cell
                         1723A  *
02F1'  BE 0016'          1724A          LDX     ORDER           transfer order and parameter bytes
02F4'  BF 0008'          1725A          STX     PORDER
02F7'  20 08             1726A          BRA     2$
                         1727A  *
                         1728A  *       start operation  --  ABORT
                         1729A  *
02F9'  F7 0000'          1730A  3$      STB     ABOREQ          set abort request
02FC'  B6 0002'          1731A          LDA     BUSY            check for prior busy
02FF'  26 09             1732A          BNE     1$
0301'  F7 0010'          1733A  2$      STB     DISINH          inhibit poll disable until order recognized
0304'  F7 000C'          1734A          STB     SOPREQ
0307'  F7 0002'          1735A          STB     BUSY
030A'  39                1736A  1$      RTS
                         1737A  *
                         1738A  ********************************************************************************
                         1739A  ********************************************************************************
                         1740A  *
                         1741A  *       command RESET
                         1742A  *
030B'  86 FF             1743A  CMRES   LDA     #TRUE.          set reset request, enabled state
030D'  B7 000A'          1744A          STA     RSTREQ
0310'  B7 0006'          1745A          STA     ENAFLG
0313'  B6 000D'          1746A          LDA     TBRIMG          lock out top board
0316'  84 FE             1747A          ANDA    #!N(TBRSTM)
0318'  8A 00             1748A          ORA     #TBRSTB
031A'  B7 7000           1749A          STA     TBREG
031D'  B7 6000           1750A          STA     FIRQFL          clear FIRQ request
0320'  39                1751A          RTS
                         1752A  *
                         1753A  ********************************************************************************
                         1754A  ********************************************************************************
                         1755A  *
                         1756A  *       "SOFTWARE" "INTERRUPT"  --  SUPERVISOR REQUEST  (DISABLE, STATUS AVAIL)
                         1757A  *
0321'  7D 0006'          1758A  SFTINT  TST     ENAFLG          it is not allowed to disable while disabled
0324'  1027 0080         1759A          LBEQ    5$
0328'  B6 000A'          1760A          LDA     RSTREQ          if RESET requested, DISABLE is irrelevant
032B'  26 78             1761A          BNE     255$
032D'  86 FF             1762A          LDA     #TRUE.          enabled, set poll disable inhibit
032F'  B7 0010'          1763A          STA     DISINH
                         1764A+         ION     <F>             ok to have TBINTs now
0332'  1C BF             1791B          ANDCC   #!N.MSK         turn F interupts on
                         1792A  *
                         1793A  *       clear idle time-out timer
                         1794A  *
0334'  BD 04AA'          1795A          JSR     CTIMER
```

```
0337' 0219'                1796A          FDB     IDLETO
                           1797A  *
                           1798A  *       SET STATUS ACCORDING TO CALLers SPEC
                           1799A  *
0339' A6 01                1800A          LDA     1,X             get bits to clear
033B' 43                   1801A          COMA
033C' B4 0000'             1802A          ANDA    PSTATS          clear them
033F' AA 84                1803A          ORA     ,X              set requested 'set' bits
0341' B7 0000'             1804A          STA     PSTATS          STATUS byte complete
                           1805A  *
0344' 86 01                1806A          LDA     #1              check for 'switch transition' bit
0346' A5 84                1807A          BITA    ,X
0348' 27 0B                1808A          BEQ     1$
                           1809A  *
                           1810A  *       switch transition set, copy transitions
                           1811A  *
034A' E6 03                1812A          LDB     3,X             get clears, if any
034C' 53                   1813A          COMB
034D' F4 0001'             1814A          ANDB    PSWSTA          clear them
0350' EA 02                1815A          ORB     2,X             OR in the 'sets'
0352' F7 0001'             1816A          STB     PSWSTA          switch byte altered as requested
                           1817A  *
0355' 48                   1818A  1$      LSLA                    check for 'input code avail' bit
0356' A5 84                1819A          BITA    ,X
0358' 27 05                1820A          BEQ     2$
                           1821A  *
                           1822A  *       input code available set, copy new input code
                           1823A  *
035A' E6 04                1824A          LDB     4,X             move key code to KEYIN byte
035C' F7 0002'             1825A          STB     PKEYIN
                           1826A  *
035F' 48                   1827A  2$      LSLA                    check for 'sense data available'
0360' A5 84                1828A          BITA    ,X
0362' 27 05                1829A          BEQ     3$
                           1830A  *
                           1831A  *       sense data available set, copy sense data code
                           1832A  *
0364' E6 05                1833A          LDB     5,X             move sense code to sense byte
0366' F7 0003'             1834A          STB     PSENSE
                           1835A  *
                           1836A  *       status transferred as reqquested
                           1837A  *
0369' 86 20                1838A  3$      LDA     #$20            check for "order complete"
036B' A5 84                1839A          BITA    ,X
036D' 27 08                1840A          BEQ     4$
036F' 86 00                1841A          LDA     #FALSE.         order complete, zap "busy"
0371' B7 0002'             1842A          STA     BUSY
0374' B7 0000'             1843A          STA     ABOREQ          and abort request
                           1844A  *
                           1845A  *       set up to strobe DIS, then test for concurrent disable poll
                           1846A  *
0377' B6 0000'             1847A  4$      LDA     PSTATS          save info for debugging
```

```
037A' B7 0011'          1848A          STA     LSTATS                  returned status
037D' EC 64             1849A          LDD     4,S
037F' FD 0014'          1850A          STD     LDPTR                   control vector pointer
0382' AE 6A             1851A          LDX     10,S
0384' 30 1F             1852A          LEAX    -1,X
0386' BF 0012'          1853A          STX     LDADR                   SWI address
0389' FC 000D'          1854A          LDD     TBRIMG          load X with DIS STROBE code
038C' 8A 02             1855A          ORA     #TBDISB
038E' 1F 01             1856A          TFR     D,X
0390' CC FF00           1857A          LDD     #(TRUE.!<8)!+(FALSE.)   constant to mask ints and clr DISINH
                        1858A  *               (last place to FIRQ before DIS strobe)
0393' 1E A8             1859A          EXG     CC,A            read and set interrupt masks
0395' F7 0006'          1860A          STB     ENAFLG          set state = disabled
0398' F7 0010'          1861A          STB     DISINH          disable will again be allowed
039B' BF 7000           1862A          STX     TBREG           strobe DIS to T. B.
039E' 85 40             1863A          BITA    #FRQMSK         if FIRQ was masked, a POLL-DIS occurred
03A0' 27 03             1864A          BEQ     255$            if FIRQ was not masked, disable is complete
03A2' B7 6000           1865A          STA     FIRQFL          POLL-DIS occurred, clear FIRQreq to set SA
                        1866A  *
                        1867A  *       set return code and exit
                        1868A  *
03A5' 6F 62             1869A  255$    CLR     2,S             clear saved B reg.  success return code
03A7' 3B                1870A          RTI                     return to CALLing program
                        1871A  *
                        1872A  *       disabled when SWI occurred --- return failure code = 1
                        1873A  *
03A8' C6 01             1874A  5$      LDB     #1              set saved B reg. to 1.  failure return
03AA' E7 62             1875A          STB     2,S
03AC' 3B                1876A          RTI                     return to CALLer


                        1878+          FILE    SWITCHES,<SWITCHES.736>
```

```
                        1880A          SUBTTL  SWITCHES    (SWITCHES.736)
                        1884A  *
                        1885A  ********************************************************************************
                        1886A  ********************************************************************************
                        1887A  *
                        1888A  *       front panel switch routine for "old style" front panel  --  736Q
                        1889A  *
                        1890A  ********************************************************************************
                        1891A  ********************************************************************************


                        1893A          INTERN  SWBYT3
                        1894A          INTERN  SWFLAG
                        1895A          INTERN  READFL
                        1896A          INTERN  SWVEC


                        1898A  ********************************************************************************
                        1899A  ********************************************************************************
                        1900A  *
          =001A         1901A          RSECT   SUPRAM
                        1902A  *
          =001A'        1903A  SWVEC   EQU     *               define base of this vector for other modules
                        1904A  *
001A'     =0001         1905A  SWBYT0  RMB     1               language and model switches
001B'     =0001         1906A  SWBYT1  RMB     1               test, FFlc, case, spacing, LPI switches
001C'     =0001         1907A  SWBYT2  RMB     1               feeder, pitch select switches
001D'     =0001         1908A  SWBYT3  RMB     1               current (rocker) switch, if any, debounced
001E'     =0001         1909A  SWBYT4  RMB     1               encoded form length (0-99)
001F'     =0001         1910A  SWBYT5  RMB     1               (reserved)
0020'     =0001         1911A  SWBYT6  RMB     1                   "
0021'     =0001         1912A  SWBYT7  RMB     1                   "
0022'     =0001         1913A  SWBYT8  RMB     1                   "
0023'     =0001         1914A  SWFLAG  RMB     1               rocker switch change flag
0024'     =0001         1915A  SWPTRN  RMB     1               rocker switch pattern byte
0025'     =0001         1916A  SWREAD  RMB     1               rocker switch reading (bus)
0026'     =0001         1917A  SWCNT   RMB     1               rocker switch filter counter


          =03AD         1919A          RSECT   SUPER
                        1920A  *
                        1921A  *       take initial reading of switches.  certain switches are read only
                        1922A  *               at power_up
                        1923A  *
03AD'     B6 3800       1924A  SWSET   LDA     SWDIP           language and model select switches
03B0'     88 00         1925A          EORA    #SWMSK0         put into "true" form
03B2'     B7 001A'      1926A          STA     SWBYT0
```

```
                        1927A  *
03B5'  C6 FE            1928A          LDB     #!N(SWSEL1)     select group 1 switches
03B7'  F7 5000          1929A          STB     SWSEL
03BA'  B6 3000          1930A          LDA     SWBUS           read them
03BD'  88 F0            1931A          EORA    #SWMSK1         mask to "true"
03BF'  1F 89            1932A          TFR     A,B             accept only readings 0-9 for h-o digit
03C1'  54               1933A          LSRB
03C2'  C1 4F            1934A          CMPB    #$4F            40x is equivalent to 8 or 9
03C4'  2F 02            1935A          BLE     1$
03C6'  84 0F            1936A          ANDA    #$F             invalid, force zero
03C8'  B7 001B'         1937A  1$      STA     SWBYT1
                        1938A  *
03CB'  C6 FD            1939A          LDB     #!N(SWSEL2)     select group 2 switches
03CD'  F7 5000          1940A          STB     SWSEL
03D0'  B6 3000          1941A          LDA     SWBUS
03D3'  88 00            1942A          EORA    #SWMSK2         mask to "true"
03D5'  B7 001C'         1943A          STA     SWBYT2
03D8'  BD 03E6'         1944A          JSR     READFL          read form length switches
03DB'  4F               1945A          CLRA
03DC'  B7 0024'         1946A          STA     SWPTRN          clear "pattern" byte
03DF'  B7 0025'         1947A          STA     SWREAD          clear "read" byte
03E2'  B7 0026'         1948A          STA     SWCNT           clear filter count byte
03E5'  39               1949A          RTS
                        1950A  *
                        1951A  *       read form length switches.  convert decimal to binary.
                        1952A  *
03E6'  34 01            1953A  READFL  PSHS    CC              save caller's interrupt mode
                        1954A+         IDF     <I>
03E8'  1A 10            1981B          ORCC    #.MSK           turn I interrupts off
03EA'  C6 F7            1982A          LDB     #!N(SWSEL4)     select form length switches
03EC'  F7 5000          1983A          STB     SWSEL
03EF'  B6 3000          1984A          LDA     SWBUS           data from switches
03F2'  35 01            1985A          PULS    CC              restore calling interrupt status
03F4'  88 FF            1986A          EORA    #SWMSK4         convert to "true"
03F6'  34 02            1987A          PSHS    A               save reading
                        1988A  *
03F8'  1F 89            1989A          TFR     A,B             test switches for BCD value
03FA'  54               1990A          LSRB                    divide by two
03FB'  C1 4C            1991A          CMPB    #$4C            4C will occur w/98 or 99
03FD'  2E 0B            1992A          BGT     1$              branch on invalid
03FF'  C4 07            1993A          ANDB    #7              test lo digit
0401'  C1 04            1994A          CMPB    #4              4 will occur w/x8 or x9
0403'  2E 05            1995A          BGT     1$              branch on invalid
                        1996A  *
0405'  84 F0            1997A          ANDA    #$F0            mask to high order digit only
0407'  C6 60            1998A          LDB     #96             a carefully chosen constant
0409'  3D               1999A          MUL                     w/ nega, puts -6*(high order digit) in "A"
040A'  40               2000A  1$      NEGA                    invalid, return zero  (-X+X=0)
040B'  AB E0            2001A          ADDA    ,S+             add in the reading again
040D'  B7 001E'         2002A          STA     SWBYT4          put result in switch vector
0410'  39               2003A          RTS
```

```
                        2005A  *
                        2006A  *       interrupt!  read legal switches
                        2007A  *
                        2008A  *       The switches are read every 6ms and saved in READ.
                        2009A  *
                        2010A  *              READ is compared to LAST; if they are the same, COUNT is set to
                        2011A  *       zero and PATTERN is set to READ.  If they are different, READ is compared
                        2012A  *       to PATTERN; if they are the same, COUNT is reduced by one and checked for
                        2013A  *       zero.  If COUNT became zero, the switch flag is set, the switch transitions
                        2014A  *       are loaded into SWTRAN, and LAST is set to READ.  If READ and PATTERN are
                        2015A  *       different PATTERN is set to READ and COUNT is set to 8.
                        2016A  *
0411'  86 FE            2017A  SWTTRP  LDA     #!N(SWSEL1)      update test selection value
0413'  B7 5000          2018A          STA     SWSEL
0416'  B6 3000          2019A          LDA     SWBUS
0419'  88 F0            2020A          EORA    #SWMSK1          put in true form
041B'  B8 001B'         2021A          EORA    SWBYT1           detect changes
041E'  84 F0            2022A          ANDA    #$F0             mask to test selector only
0420'  B8 001B'         2023A          EORA    SWBYT1           effect changes in SWBYT1
0423'  1F 89            2024A          TFR     A,B              filter invalid readings
0425'  54               2025A          LSRB
0426'  C1 4F            2026A          CMPB    #$4F             40x represents test sw=8 or 9
0428'  2E 03            2027A          BGT     10$              do not save if invalid (Ax through Fx)
042A'  B7 001B'         2028A          STA     SWBYT1
                        2029A  *
                        2030A  *       now proceed to momentary switches
                        2031A  *
042D'  B6 0023'         2032A  10$     LDA     SWFLAG           new switches are not legal if last not processed
0430'  26 42            2033A          BNE     255$
0432'  86 FB            2034A          LDA     #!N(SWSEL3)      select rocker switches
0434'  B7 5000          2035A          STA     SWSEL
0437'  B6 3000          2036A          LDA     SWBUS            read switches
043A'  88 FF            2037A          EORA    #SWMSK3          set bits to 1 means "pushed"
043C'  27 0C            2038A          BEQ     3$               branch if no switch pressed
                        2039A  *
                        2040A  *       check for only one switch pressed
                        2041A  *
043E'  C6 08            2042A          LDB     #8               eight switches
0440'  48               2043A  1$      LSLA                     shift switch point to carry
0441'  25 03            2044A          BCS     2$               if carry = 1, activated switch is found
0443'  5A               2045A          DECB                     count switches in byte
0444'  20 FA            2046A          BRA     1$               loop, there MUST be a switch
                        2047A  *
0446'  26 2C            2048A  2$      BNE     255$             is there more than 1 switch pressed?
0448'  1F 98            2049A          TFR     B,A              no.  transfer switch number to A
                        2050A  *
                        2051A  *       switch code (0 to 8) in A, do filter
                        2052A  *
```

```
044A'  B7 0025'          2053A  3$     STA     SWREAD          save as read
044D'  B1 001D'          2054A         CMPA    SWBYT3          check against LAST value
0450'  26 08             2055A         BNE     4$              branch if changed
0452'  7F 0026'          2056A         CLR     SWCNT           same as LAST, reset count and PATTERN
0455'  B7 0024'          2057A         STA     SWPTRN
0458'  20 1A             2058A         BRA     255$            exit
                         2059A  *
                         2060A  *      read `= last, check for read = pattern
                         2061A  *
045A'  B1 0024'          2062A  4$     CMPA    SWPTRN          is it a new pattern
045D'  26 0D             2063A         BNE     5$
045F'  7A 0026'          2064A         DEC     SWCNT           same pattern, reduce debounce count
0462'  26 10             2065A         BNE     255$
                         2066A  *
                         2067A  *      debounce complete, set new value and flag
                         2068A  *
0464'  B7 001D'          2069A         STA     SWBYT3          set rocker switch byte to new value
0467'  73 0023'          2070A         COM     SWFLAG          set SWITCH FLAG to .TRUE
046A'  20 08             2071A         BRA     255$
                         2072A  *
                         2073A  *      read `= pattern, switch is still bouncing
                         2074A  *
046C'  B7 0024'          2075A  5$     STA     SWPTRN          set current pattern
046F'  86 08             2076A         LDA     #8              initialize debounce count
0471'  B7 0026'          2077A         STA     SWCNT
                         2078A  *
                         2079A  *      switches checked and appropriate action taken, exit
                         2080A  *
0474'  39                2081A  255$   RTS

                         2083+         FILE    TIMERS,<TIMERS.SRC>
```

```
                                2085A          SUBTTL  TIMERS    (TIMERS.SRC)
                                2089A  *
                                2090A  ********************************************************************************
                                2091A  *
                                2092A  *       TIMERS  --  SET, CLEAR, INTERRUPT
                                2093A  *
                =0027           2094A          RSECT   SUPRAM
0027'           =0010           2095A  TIMVEC  RMB     MXTIMR*2          timer, remaining durations here
0037'           =0010           2096A  TIMTRP  RMB     MXTIMR*2          timer, trap addresses here
0047'           =0001           2097A  NTIMR   RMB     1                 number of active timers
0048'           =0001           2098A  ZTIMRS  RMB     1                 number of expired timers found (this clock)



                =0475           2100A          RSECT   SUPER
                                2101A          INTERN  STIMER            set timer entry
                                2102A          INTERN  CTIMER            clear a timer entry
                                2103A          INTERN  CLKII             process a clock interrupt
                                2104A          INTERN  CLKINI            initialize clock/timer module (coldstart)

                                2106A  *
                                2107A  ********************************************************************************
                                2108A  *
                                2109A  *       STIMER  --  SET A TIMER TRAP TO OCCUR
                                2110A  *
                                2111A  *       calling sequence  --
                                2112A  *
                                2113A  *       CALL    STIMER
                                2114A  *       TRAP ADDRESS              address to trap to after
                                2115A  *       ADDRESS OF COUNT**        number of ticks to wait  (literal)
                                2116A  *       (return here)
                                2117A  *
                                2118A  *       **(NOTE:  count is NOT stored in this word.  this is a pointer.)
                                2119A  *
                                2120A  *
0475'  34 71                    2121A  STIMER  PSHS    CC,X,Y,U           save CALLer's interrupt mode and registers
                                2122A+ 3$      IOF     <B>                can't have interupts here
0477'  1A 50                    2149B          ORCC    #.MSK              turn B interrupts off
0479'  B6 0047'                 2150A          LDA     NTIMR              check count of timers now active
047C'  81 08                    2151A          CMPA    #MXTIMR            maximum already?
047E'  25 04                    2152A          BLO     1$                 branch if ok
                                2153A+         ION     <B>                wait for a timer available
0480'  1C AF                    2180B          ANDCC   #!N.MSK            turn B interupts on
0482'  20 F3                    2181A          BRA     3$
                                2182A  *
0484'  48                       2183A  1$      ASLA                       double the count
```

```
0485'  8E 0037'          2184A          LDX     #TIMTRP
0488'  EE 67             2185A          LDU     7,S             trap address pointer to U
048A'  10AE C4           2186A          LDY     ,U              actual trap adress to Y
048D'  10AF 86           2187A          STY     A,X             trap address to TIMTRP
0490'  8E 0027'          2188A          LDX     #TIMVEC
0493'  10AE D8 02        2189A          LDY     [2,U]           get specified count
0497'  26 04             2190A          BNE     2$              must be non-zero
                         2191A  *
0499'  108E 0001         2192A          LDY     #1              was zero, make it 1
049D'  10AF 86           2193A  2$      STY     A,X             store count
                         2194A  *
04A0'  33 44             2195A          LEAU    4,U             advance return pointer
04A2'  EF 67             2196A          STU     7,S
04A4'  7C 0047'          2197A          INC     NTIMR           increment timer count
04A7'  35 71             2198A          PULS    CC,X,Y,U        restore calling interrupt mode and registers
04A9'  39                2199A          RTS                     effect return
                         2200A  *
                         2201A  ********************************************************************************
                         2202A  *
                         2203A  *       CTIMER  --  CLEAR TIMER IF SET
                         2204A  *
                         2205A  *       calling sequence  --
                         2206A  *
                         2207A  *       CALL    CTIMER
                         2208A  *       TRAP ADDRESS            trap address of timer to clear
                         2209A  *       (return)
                         2210A  *
                         2211A  *
04AA'  34 71             2212A  CTIMER  PSHS    CC,X,Y,U        save CALLer's interrupt mode and registers
                         2213A+         IOF     <B>             can't have interrupts here
04AC'  1A 50             2240B          ORCC    #.MSK           turn B interrupts off
04AE'  B6 0047'          2241A          LDA     NTIMR           if no timers are set, routine is done
04B1'  27 1D             2242A          BEQ     255$
                         2243A  *
                         2244A  *       LOOK FOR THE SPECIFIED TRAP ADDRESS
                         2245A  *
04B3'  8E 0037'          2246A          LDX     #TIMTRP         base of vector
04B6'  10AE F8 07        2247A          LDY     [7,S]           specified trap address
04BA'  10AC 81           2248A  1$      CMPY    ,X++            is this the one?
04BD'  27 05             2249A          BEQ     100$            branch if yes
04BF'  4A                2250A          DECA                    reduce and check ccunt of unchecked traps
04C0'  26 F8             2251A          BNE     1$              loop if not all active traps tested
04C2'  20 0C             2252A          BRA     255$            if all have been, trap was not here, exit
                         2253A  *
                         2254A  *       SPECIFIED TRAP WAS FOUND  --  REMOVE IT
                         2255A  *
04C4'  10AE 88 EE        2256A  100$    LDY     TIMVEC-TIMTRP-2,X       check for timer expired*
04C8'  26 03             2257A          BNE     99$
04CA'  7A 0048'          2258A          DEC     ZTIMRS          reduce count for interrupt processer
04CD'  BD 04D9'          2259A  99$     JSR     TSQUEZ          squeeze out the timer
                         2260A  * *(timer may be expired if CTIMER is called by a trap processor)
                         2261A  *
```

```
                        2262A  *       TIMER IS REMOVED (OR WAS NOT THERE)  --  RETURN
                        2263A  *
04D0'  EE 67            2264A  255$    LDU     7,S             update return pointer
04D2'  33 42            2265A          LEAU    2,U
04D4'  EF 67            2266A          STU     7,S
04D6'  35 71            2267A          PULS    CC,X,Y,U        restore calling program's interrupt mode & regs
04D8'  39               2268A          RTS
                        2269A  *
                        2270A  *
                        2271A  *       "PHYSICALLY" REMOVE TIMER FROM VECTORS
                        2272A  *
                        2273A  *               ('X' points to TRAP following the one to remove)
                        2274A  *               ('A' contains count of timers at/after the one to be deleted)
                        2275A  *
04D9'  31 10            2276A  TSQUEZ  LEAY    TIMVEC-TIMTRP,X         point "Y" to time corresponding to "X"
04DB'  20 08            2277A          BRA     98$
04DD'  EE 81            2278A  99$     LDU     ,X++            get next trap address
04DF'  EF 1C            2279A          STU     -4,X            move it down
04E1'  EE A1            2280A          LDU     ,Y++            move corresponding time down
04E3'  EF 3C            2281A          STU     -4,Y
04E5'  4A               2282A  98$     DECA                    reduce and check count of remaining traps
04E6'  26 F5            2283A          BNE     99$
04E8'  7A 0047'         2284A          DEC     NTIMR           reduce overall count of active traps
04EB'  39               2285A          RTS
                        2286A  *
                        2287A  *
                        2288A  ********************************************************************************
                        2289A  ********************************************************************************
                        2290A  ********************************************************************************
                        2291A  *
                        2292A  *
                        2293A  *       CLKII  --  PROCESS CLOCK INTERRUPTS HERE
                        2294A  *
                        2295A+ CLKII   CCF                     clear clock flag
04EC'  7D 8004          2296B          TST     T1C.L           clears the clock flag
                        2297A  *
                        2298A  *       process requested clock traps
                        2299A  *
                        2300A  *               (decrement time counts on unexpired timers)
                        2301A  *               (form count of expired but still active timers)
                        2302A  *               ((include timers which expire this time))
                        2303A  *
04EF'  34 01            2304A          PSHS    CC              save entry value of 'F'
04F1'  32 7E            2305A          LEAS    -2,S            reserve stack space
                        2306A+         IOF     <F>             'F' interrupts often set/clear traps
04F3'  1A 40            2333B          ORCC    #.MSK           turn F interrupts off
04F5'  B6 0047'         2334A          LDA     NTIMR           check count of active traps
04F8'  27 44            2335A          BEQ     255$            exit if there are no timers active
04FA'  48               2336A          ASLA                    double count to use as an index
04FB'  5F               2337A          CLRB                    zero count of 'just expired' timers
04FC'  108E 0025'       2338A          LDY     #TIMVEC-2       get base of 'time left' vector
0500'  AE A6            2339A  1$      LDX     A,Y             reduce count on timer pointed to
```

```
0502'  27 06          2340A          BEQ     15$             MAY already be zero (however unlikely)
0504'  30 1F          2341A          LEAX    -1,X
0506'  AF A6          2342A          STX     A,Y
0508'  26 01          2343A          BNE     2$              check for count reduced to zero
050A'  5C             2344A  15$     INCB                    count timer as expired
050B'  80 02          2345A  2$      SUBA    #2              reduce remaining count
050D'  26 F1          2346A          BNE     1$
                      2347A  *
050F'  F7 0048'       2348A          STB     ZTIMRS          store # of timers 'just expired'
                      2349A  *
0512'  7D 0048'       2350A  3$      TST     ZTIMRS          check for expired timers
0515'  2F 27          2351A          BLE     255$            branch if none remain
                      2352A  *
                      2353A  *       FIND AN 'EXPIRED' TIMER
                      2354A  *
0517'  108E 0027'     2355A          LDY     #TIMVEC
051B'  B6 0047'       2356A          LDA     NTIMR
051E'  AE A1          2357A  20$     LDX     ,Y++            check a 'time remaining' word
0520'  27 05          2358A          BEQ     40$             branch when expired timer found
0522'  4A             2359A          DECA                    have all been checked?
0523'  26 F9          2360A          BNE     20$
0525'  20 17          2361A          BRA     255$            yes, exit
                      2362A  *
                      2363A  *       GOT A LIVE ONE
                      2364A  *
0527'  7A 0048'       2365A  40$     DEC     ZTIMRS          reduce 'remaining' count of 'just expired'
052A'  30 A8 10       2366A          LEAX    TIMTRP-TIMVEC,Y         SQUEZ requires list address of trap + 2
052D'  EE 1E          2367A          LDU     -2,X            ...in X.  get actual trap address from list
052F'  EF E4          2368A          STU     ,S              save it in the stack
0531'  BD 04D9'       2369A          JSR     TSQUEZ          squeeze out expired timer
0534'  A6 62          2370A          LDA     2,S             allow F interrupts during trap routine
0536'  1F 8A          2371A          TFR     A,CC            ...if they were enabled when CLKII entered
0538'  AD F4          2372A          JSR     [,S]            execute the trap routine
                      2373A+         IOF     <F>             disallow F during further clock processing
053A'  1A 40          24003          ORCC    #.MSK           turn F interrupts off
053C'  20 D4          2401A          BRA     3$              loop to check for more
                      2402A  *
                      2403A  *       ALL TRAPS ARE PROCESSED  --  EXIT
                      2404A  *
053E'  32 62          2405A  255$    LEAS    2,S             release added stack space
0540'  35 01          2406A          PULS    CC              restore entry 'F'
0542'  39             2407A          RTS                     exit
                      2408A  *
                      2409A  ********************************************************************************
                      2410A  ********************************************************************************
                      2411A  ********************************************************************************
                      2412A  *
                      2413A  *       CLOCK/TIMER INITIALIZE          (coldstart)
                      2414A  *
                      2415A  *       called by supervisor to set up clock hardware and softwware
                      2416A  *
0543'  7F 0047'       2417A  CLKINI  CLR     NTIMR           clear count of active timers
```

```
0546'  B6 800B          2418A          LDA     VIAACR          set up timer mode
0549'  84 3F            2419A          ANDA    #$3F            clear any mode previously set
054B'  8A 40            2420A          ORA     #VIATC1         set for free-running mode
054D'  B7 800B          2421A          STA     VIAACR
0550'  CC 7017          2422A          LDD     #VIATCN         set up the base interval timer
0553'  FD 8004          2423A          STD     T1L.La          activates timer
                        2424A+         CCF                     clear clock flag
0556'  7D 8004          2425B          TST     T1C.L           clears the clock flag
0559'  86 C0            2426A          LDA     #VIATC2         set bit to enable timer 1 interrupt
055B'  B7 800E          2427A          STA     VIAIER
055E'  39               2428A          RTS                     return to supervisor
                        2429A  *
                        2430A  ********************************************************************************
                        2431A  ********************************************************************************
                        2432A  ********************************************************************************


                        2434+          FILE    MISCELLANEOUS,<SUPERMISC.736>
```

```
                                2436A          SUBTTL  MISCELLANEOUS    (SUPERMISC.736)
                                2440A  *
                                2441A  ********************************************************************************
                                2442A  *
                                2443A  *
                                2444A  *       miscellaneous routines for system (p/o SUPERVISOR assembly)
                                2445A  *
                                2446A  *
                                2447A  ********************************************************************************
                                2448A  *
          =0049                 2449A          RSECT   SUPRAM
                                2450A  *
0049'     =0001                 2451A  LEDIMG  RMB     1               image of current LED selects (1=on)
004A'     =0001                 2452A  ALRMG   RMB     1               image of B.B. alarm status
                                2453A  *




                                2455A  *
                                2456A          INTERN  ERROR           fatal error lockup routine
                                2457A          INTERN  ALARM           alarm on/off routine
                                2458A          INTERN  LEDSET          LED on/off routine
                                2459A  *

          =055F                 2461A          RSECT   SUPER
                                2462A  *
                                2463A  *       LEDSET  turn on/off/load indicated LED
                                2464A  *
                                2465A  *               (LD)A   LED number
                                2466A  *               (LD)B   on/off/value code
                                2467A  *               CALL    LEDSET          to set/clear/load specified LED
                                2468A  *
055F'     34 11                 2469A  LEDSET  PSHS    X,CC            routine uses X & disables interrupts
0561'     8E 058C'              2470A          LDX     #BITVEC         base of LED number:bit correlation table
0564'     A6 86                 2471A          LDA     A,X             get specified bit
                                2472A+         IOF     <I>             suppress interrupts
0566'     1A 10                 2499B          ORCC    #.MSK           turn I interrupts off
0568'     5D                    2500A          TSTB                    test on/off switch
0569'     27 05                 2501A          BEQ     1$
                                2502A  *
                                2503A  *       turn it on
                                2504A  *
056B'     BA 0049'              2505A          ORA     LEDIMG          or in existing bit(s), if any
056E'     20 04                 2506A          BRA     2$
                                2507A  *
                                2508A  *       turn it off
```

```
                                2509A  *
                                2510A  1$       COMA                        form mask
0570'  43                       2511A           ANDA     LEDIMG             clear the specified bit
                                2512A  *
0574'  B7 0049'                 2513A  2$       STA      LEDIMG             store new image
0577'  35 11                    2514A           PULS     X,CC               restore X and interrupts
0579'  BA 004A'                 2515A           ORA      ALRMG              merge with alarm code
057C'  88 FF                    2516A           EORA     #LEDMSK!+ALRMSK            put data in form to drive bus
057E'  B7 4000                  2517A           STA      LEDBUS             and set register
0581'  39                       2518A           RTS                         return to caller
                                2519A  *
                                2520A  *        initialize LED image
                                2521A  *
0582'  4F                       2522A  LEDINI   CLRA                        clear image (no LEDs on)
0583'  B7 0049'                 2523A           STA      LEDIMG
0586'  88 FF                    2524A           EORA     #LEDMSK!+ALRMSK
0588'  B7 4000                  2525A           STA      LEDBUS             LEDs and alarm off
058B'  39                       2526A           RTS
                                2527A  *
058C'  02                       2528A  BITVEC   FCB      LEDCHK             "CHECK" is LED 0
058D'  10                       2529A           FCB      LED1
058E'  20                       2530A           FCB      LED2
058F'  40                       2531A           FCB      LED3
0590'  01                       2532A           FCB      LEDRDY             "READY" is LED 4
0591'  04                       2533A           FCB      LEDHLD             "HOLD" is LED 5
0592'  00                       2534A           FCB      0                  (reserved)
0593'  00                       2535A           FCB      0                   "
                                2536A  *
                                2537A  *
                                2538A  *        ALARM    --       ON or OFF
                                2539A  *
                                2540A  *        (LD)A    TRUE.  or  FALSE.
                                2541A  *        CALL     ALARM
                                2542A  *
0594'  34 01                    2543A  ALARM    PSHS     CC                 save interrupt mode
                                2544A+          IOF      <I>
0596'  1A 10                    2571B           ORCC     #.MSK              turn I interrupts off
0598'  84 80                    2572A           ANDA     #BEEPMK            mask to beeper bit
059A'  B7 004A'                 2573A           STA      ALRMG              store result in image byte
059D'  BA 0049'                 2574A           ORA      LEDIMG             combine with LEDs for output
05A0'  88 FF                    2575A           EORA     #LEDMSK!+ALRMSK            put in bus data form
05A2'  B7 4000                  2576A           STA      LEDBUS
05A5'  35 01                    2577A           PULS     CC                 restore interrupt mode
05A7'  39                       2578A           RTS
                                2579A  *
                                2580A  *
                                2581A  *        FATAL ERROR  --  set "CHECK" light, turn on alarm and hang
                                2582A  *
                                2583A  *
                                2584A+ ERROR    IOF      <B>                interrupts OFF
05A8'  1A 50                    2611B           ORCC     #.MSK              turn B interrupts off
05AA'  C6 FF                    2612A           LDB      #TRUE.
```

```
05AC'  BD 055F'          2613A          JSR     LEDSET          turn on requested LED
05AF'  4F                2614A          CLRA
05B0'  BD 055F'          2615A          JSR     LEDSET          turn on "CHECK" light
05B3'  C6 FF             2616A          LDB     #TRUE.          turn on alarm
05B5'  BD 0594'          2617A          JSR     ALARM
05B8'  7E 05B8'          2618A          JMP     *               hang until power off
                         2619A  *

                         2621           END
```

```
ABOCOD   0001          ABOFLG   0001'IN      ABOREQ   0000'IN      ALARM    0594'IN      ALRMG    004A'
ALRMSK   0080          BEEPMK   0080         BEGIN    00ED'        BITVEC   058C'        BUSY     0002'IN
CALL     Macro         CCF      Macro        CLKII    04EC'IN      CLKINI   0543'IN      CLKTRP   0253'
CMDIS    02A5'         CMENB    02C7'        CMENB1   02D3'        CMRES    030B'        CMSOP    02E4'
CMTBL    029D'         CSTART   0000'IN      CTIMER   04AA'IN      DCXIT    011A'        DEVIRQ   0010 EX
DEVRST   0009 EX       DEVSTA   0008 EX      DEVTOF   0003'        DEVTR    0004'        DISINH   0010'
ECBIT    0010          ENAFLG   0006'IN      ERROR    05A8'IN      EXSTAT   0007'IN      FALSE.   0000
FILE     Macro         FIRQFL   6000         FLTFLG   0005 EX      FLTREQ   0006 EX      FLTTIM   0007 EX
FRQMSK   0040          HLDFLG   000F EX      HLDREQ   000F'IN      HOLD     000E EX      HOLDST   0220'
HOLDSV   0005          HOLDTO   022E'        HTOCON   023C'        HTOFLG   0004 EX      IDLABO   01BF'
IDLE     00FA'         IDLELI   0217'        IDLETC   0212'        IDLETO   0219'        IDLE1    01DD'
INVAL    0205'         INVALD   0211'        IOF      Macro        ION      Macro        IRBIT    0008
IRQINT   023E'         LDADR    0012'        LDPTR    0014'        LEDBUS   4000         LEDCHK   0002
LEDHLD   0004          LEDIMG   0049'        LEDINI   0582'        LEDMSK   007F         LEDRDY   0001
LEDSET   055F'IN       LED1     0010         LED2     0020         LED3     0040         LNGMSK   00F8
LPS      000C EX       LPSCOD   0004         LSTATS   0011'        LSTCOM   0016'IN      LSTORD   0017'IN
MEMBAS   2000          MEMCHK   005B'        MEMERR   009A'        MEMLEN   0800         MODLMK   0007
MXTIMR   0008          NOABRT   0200'        NTIMR    0047'        OCBIT    0020         ORDER    0016'
PBUFR    0050'IN       PEAB     1050'IN      PEACT    1010'IN      PIDENT   000B'IN      PKEYIN   0002'IN
PMODE    0010'IN       PMPP     0018'IN      PMSA     0012'IN      PMSL     0014'IN      POINTR   0018'
PORDER   0008'IN       PPARS    0017'IN      PRGINI   00CC'        PRICOD   0003         PRINTM   000D EX
PRIPOR   0003 EX       PRIRST   000A EX      PSENSE   0003'IN      PSPOR    0002 EX      PSTATS   0000'IN
PSWSTA   0001'IN       PTRTRP   024F'        PTSTM    004A'IN      RDYLED   0004         READFL   03E6'IN
RSTCOM   011D'         RSTFLG   000B'IN      RSTREQ   000A'        SDABIT   0004         SFTINT   0321'
SOPCOM   015E'         SOPLPS   018B'        SOPPRT   01A0'        SOPREQ   000C'        SOPSSA   017B'
SSA      000B EX       SSACOD   0002         SSAPOR   0001 EX      STACK    0000'        STATUS   2800
STCKLN   0100          STIMER   0475'IN      STKSET   0043'        STKTOP   0100'        SWBUS    3000
SWBYT0   001A'         SWBYT1   0018'        SWBYT2   001C'        SWBYT3   001D'IN      SWBYT4   001E'
SWBYT5   001F'         SWBYT6   0020'        SWBYT7   0021'        SWBYT8   0022'        SWCNT    0026'
SWDIP    3800          SWFLAG   0023'IN      SWMSK0   0000         SWMSK1   00F0         SWMSK2   0000
SWMSK3   00FF          SWMSK4   00FF         SWPTRN   0024'        SWREAD   0025'        SWSEL    5000
SWSEL1   0001          SWSEL2   0002         SWSEL3   0004         SWSEL4   0008         SWSET    03AD'
SWTTRP   0411'         SWVEC    001A'IN      SYINIT   009F'        TBBBFB   0008         TBBBFM   0008
TBDISB   0002          TBDISM   0002         TBINT    0281'        TBREG    7000         TBRIMG   000D'
TBRSTB   0000          TBRSTM   0001         TBSMSK   00C0         TICK     0006         TICKM    2710
TIMTRP   0037'         TIMVEC   0027'        TRUE.    FFFF         TSQUEZ   04D9'        T1C.H    8005
T1C.L    8004          T1L.H    8007         T1L.LA   8004         T1L.LB   8006         T2C.H    8009
T2C.L    8008          T2L.L    8008         VECTOR   0000'        VIA      8000         VIAACR   800B
VIADDA   8003          VIADDB   8002         VIADMK   BFFF         VIAIER   800E         VIAIFR   800D
VIAIRA   8001          VIAIRB   8000         VIAOAN   800F         VIAORA   8001         VIAORB   8000
VIAPCR   800C          VIAPC1   0082         VIAPC2   0002         VIAPC3   00CA         VIATCN   7017
VIATC1   0040          VIATC2   00C0         VIATC3   0040         ZTIMRS   0048'        .IOXAR   Macro
.MSK     =0050         .RAMB.   2000         .RAME.   27FF         .ROMB.   C000         .ROME.   FFFF
```

1 error detected