```
; File SYSEQU.TEXT  - Macintosh  system  equates  file.
; MACWORKS  copy
;_____________________________________________________________________
;
;
; WRITTEN  BY:  Bud Tribble          6-May-81
;
; All  system  routines  include  this equate  file.
;
;   MODIFICATION  HISTORY:
;
;       Integrate  Lisagraf                                    --  alo 23-Dec-81
;       Configured  for Alpha release  (one heap)              --  alo  5-Feb-82
;       Moved  Lisagraf  variables  up to allow  more  room    --  alo 10-Feb-82
;       Added  system  base date & time variables              --  alo 13-Feb-82
;       Allowed  20 bytes for system parameter  ram area       --  alo  1-Mar-82
;       Added  ptr for SYSOUTFIB,SYSINFIB,SYSREFNUM(macpaslib)
;                                                              --  alo 1-Apr-82
;       Added  SCC hardware  addresses;  removed  6551  stuff;
;       pulled  MBSTATE  and MBTICKS  from kbd driver  vars    --  LAK 10-May-82
;       added  SCC write reg 5 globals;  clock-keyboard
;       synchronization   flags  . . .                         --  LAK 27-May-82
;       got  rid  of VBL queue  element  "mode"  word          --  LAK 04-Jun-82
;       changed  KEYREPCOUNT  to KEYREPTIME;  added
;       procedural  interface  variables  for key mapping      --  LAK 07-Jun-82
;       updated  for 512-dots                                  --  LAK 26-Aug-82
;       changed  VIA addresses  for timing  problem            --  LAK 17-Sep-82
;       changed  IWM addresses  for better  margin             --  LAK 23-Sep-82
;       updated  to 384 current  version  (0.5D)               --  LAK 10-Oct-82
;       added  FSInitDbg  for Filler1  (sysinit,  debug)       --  LAK 19-Oct-82
;       added  TagData  field  for twiggy  driver              --  LAK 01-Nov-82
;       added  DrvInstalled  field  for twiggy  driver;
;        changed  WarmCold  to TwiggyVars                      --  LAK 02-Nov-82
;       removed  DrvInstalled  field;  removed  FIB pointers,
;        and changed  start  of system  heap;  changed  loader
;        variable  area                                        --  LAK 18 Nov 82
;       added  a compare  string  jump  vector                 --  LAK 10 Dec 82
;       changed  driver  definitons,  etc.                     --  LAK 17 Dec 82
;       added  VIA equates                                     --  LAK 03 Jan 83
;       added  new i/o param  blk equates  for new fs          --  LAK 17 Jan 83
;       added  storage  manager  error  codes                  --  LAK 04 Feb 83
;       add disk cal switch  error  code                       --  LAK 05 Feb 83
;       added  keyboard  task  vector  in $124  (KybdTask);
;        broke  error  codes  out into SysErr.Text
;        added  LoadTrap  lomem  var                           --  LAK 16 Feb 83
;       put dispatch  table  where  macsbug  globals  used
;        to be and moved  down toolbox  vars.                  --  LAK 18 Feb 83
;       Added  DSAlertTab  pointer                             --  AJH 29 Mar 83
;       Added  BootDrive  field                                --  AJH 04 Apr 83
;       Added  PollStack,  PollProc,  DskErr,  DskRtnAdr       --  LAK 11 Apr 83
;       Added  SonyVars,PWMBuf2                                --  LAK 17 Apr 83
;       Changed  screen,  PWM address  for 512K proto          --  AJH 01 May 83
;       Added  Resource  def                                   --  BJH 09 May 83
;       Added  PWMValue                                        --  LAK 09 May 83
;       Added  Heap check  hooks                               --  MPH 10 May 83
;       Changed  SCC addresses  (high bits on)                 --  LAK 25 May 83
;       Added  DQFSID,
;        Stretched  KeyPadMap  to 8 bytes                      --  LAK 01 Jun 83
;       Added  Mask constant,  changed  Heap check  hooks      --  MPH 02 Jun 83
;       Added  equates  for MinStack,  DefltStack  for memory
;        manager;  moved  UnitEntries  equate to StartInit;
;        added  UnitNtryCnt  lomem  var instead.               --  LAK 16 Jun 83
```

```
;       Added "NoQueue" bit definition                          --  AJH 23 Jun 83
;       Added "CurIOTrap" (replacing "Filler3" )                --  AJH 26 Jun 83
;       Added "forTwiggy" conditional assembly switch           --  AJH 27 Jun 83
;       Cleaned up: changed unused vars to Fillerx equates;
;        changed ARdCmd,AWrCmd,ACtlCmd,AStsCmd    to match
;        trap numbers . . .                                     --  LAK 15 Jul 83
;       Added mmInOK for memory manager checking.               --  LAK 24 Jul 83
;       Added mmDefFlags for memory manager zone init.          --  LAK 31 Jul 83
;       Added DskVerify for disk driver verify/read flag
;         in place of DskDump                                   --  LAK 06 Aug 83
;       Removed DNeedsStorage equate (no longer used)           --  LAK 08 Aug 83
;       Added Loader global definitions, no more SPBot          --  AJH 11 Aug 83
;       Added IWM lomem variable . . .                          --  LAK 15 Aug 83
;       Added EvtBufCnt, JCrsrTask, moved FinderName            --  LAK 17 Aug 83
;       Moved JCrsrTask to GrafEqu.                             --  LAK 18 Aug 83
;       Added scrap vars to $960 .                              --  SC  18 Aug 83
;       Added StkLowPt var for useful measurements.             --  LAK 19 Aug 83
;       Added DoubleTime and CaretTime                          --  SC  19 Aug 83
;       Added SPClikCaret to parameter ram                      --  SC  19 Aug 83
;       Folded SPKbd stuff in parameter ram                     --  SC  19 Aug 83
;       Corrected ROM size counts (RomBSize, RomWSize)          --  AJH 22 Aug 83
;       Added keyClick bit in parameter ram                     --  AJH 11 Sep 83
;       Added GrowZone handle/pointer warnings                  --  SC  12 Sep 83
;       Added Resource Manager equates                          --  BLH 22 Sep 83
;       Added 16 bytes of printing globals                      --  AJH 30 Sep 83
;       Added desk ornament message equates,changed events      AJH 08 Oct 83
;       Added DQDrvSize                                         --  LAK 02 Nov 83
;       Added equates for JGNEFilter, ScrDmpEnb, ScrDmpType     LAK 09 Nov 83
;       Added ScreenRow                                     -- AJH/WDA  10 Nov 83
;       Added equates for DSAlertRect, DSDrawProc,
;         EjectNotify, CoreEditVars, QDExist,
;         WWExist and JShell                                    --  LAK 16 Nov 83
;       Added equate for AlarmState                             --  LAK 22 Nov 83
;       Added equate for InitApplZone notify proc               --  LAK 06 Dec 83
;       Added equates for screen vertical, horiz res            --  LAK 19 Dec 83
;       Added NeedsGoodBye                                      --  AJH 03 Jan 84
;       Added SysVersion                                        --  AJH 06 Jan 84
;       Added DSWndUpdate, SEvtEnb flags                        --  LAK 11 Jan 84
;       Added FontFlag for font manager bug fix                 --  AJH 12 Jan 84

;       Modified for MacWorks                                   --  JWP 1983/1984
;       Added TermProc routine pointer                          --  KWK 12 Jun 84
;       Added equates for hard disk driver                      --  RDC 26 Jun 84
;       Added HDiskStart routine pointer                        --  RDC 28 Jun 84
;       Added notes about low-mem MacWorks globals              --  KWK 12 Aug 84
;       RomStart = $80000, DispatchTab = RomStart+$11000--      twm 13 Sep 84
;_____________________________________________________________________________

;_____________________________________________________________________________
;
;                       System Low Memory Layout
;
;          0000-00FF        Exception Vectors defined by the hardware
;          0100-0340        System Communications Area
;          0340-03FF        File System globals (172 bytes used)
;          0400-07FF        system support globals for Lisa 2
;          0800-08FF        Mouse/Cursor low-memory globals
;          0900-097F        more os stuff
;          0980-0AFF        toolbox variables
;          0B00-XXXX        Start of system heap
;
;     Some other important system data structures are allocated on the heap
```

```
;
;_____________________________________________________________

; conditional  assembly  switch  to select  Twiggy  or Sony

forTwig     .EQU     0

; conditional  assembly  switch  to select  ROM code for Macintosh  or Lisa or Yacc

onMac     .EQU     0
onLisa    .EQU     0
onYacc    .EQU     1

;---------------------------------------------------------------
; System  Constants
;---------------------------------------------------------------
;
; Hardware  trap  routine  numbers
;

.IF      onLisa
_SetNMIKey          .EQU     116
_SetToggleKey       .EQU     162
_KeyRoutine         .EQU     118
_SetKeyRoutine      .EQU     120
_VBLRoutine         .EQU     172
_SetVBLRoutine      .EQU     174
_KeyIsDown          .EQU     52
_Keyboard           .EQU     56
_Beep               .EQU     50
_SetVolume          .EQU     44
_Noise              .EQU     46
_Silence            .EQU     48
_TimeStamp          .EQU     88
_SetTimeStamp       .EQU     90
_PowerDown          .EQU     38
_RampContrast       .EQU     32
.ENDC

; Interrupt  disable  masks
.IF      onMac

VIAIntMask          .EQU     $0100                ; no interrupts  from VIA
SCCIntMask          .EQU     $0300                ; no interrupts  from SCC

.ENDC
.IF      onYacc

VIAIntMask          .EQU     $0300                ; no interrupts  from VIA
SCCIntMask          .EQU     $0500                ; no interrupts  from SCC

.ENDC

; Physical  Device  Equates

.IF      onMac
;   note -- all  screen  and sound  addresses  are for the 512K Mac,
;   which  will  also  work  for the 128K machine  since the address
;   space  wraps

ScreenLow           .EQU     $0007A700            ; top of screen  screen  address
SoundLow            .EQU     $0007FD00            ; low sound  buffer  address
```

```
PWMBuffer       .EQU    $0007FD01           ; PWM bytes are low bytes
SndBufWLen      .EQU    $0172               ; sound/disk buffer word length
OvlyRAM         .EQU    $00600000           ; low RAM address when overlay is on
OvlyScreen      .EQU    $0067A700           ; top of screen with overlay

ROMStart        .EQU    $00400000           ; starting address of ROM code
ROMWSize        .EQU    $00008000           ; 32768 words in present ROM
ROMBSize        .EQU    $00010000           ; 65536 bytes in present ROM
MemLSize        .EQU    $00008000           ; memory contains 32K long words
MemWSize        .EQU    $00010000           ; 64K words
MemBSize        .EQU    $00020000           ; and 128K bytes
LineLen         .EQU    $40                 ; horizontal screen line has 64 bytes
.ENDC                                       ; of .if onMac

.IF     onYacc
RomHihWord      .equ    $0009               ; current high word address of rom image
ScreenLow       .EQU    $000B4000           ; top of screen screen address
ScrnLSize       .EQU    $000025D0           ; there are 9680 longs for 1 plane
ScrnLong        .EQU    $000025D0           ; number of longs for entire screen
ScreenOff       .EQU    $9800               ; offset from one frame buffer to next
LineLen         .EQU    $50                 ; horizontal screen line has 80 bytes
MacIconLoc      .EQU    ScreenLow+<LineLen*141>+38      ; Mac Icon Location
FaceLoc         .EQU    MacIconLoc+<LineLen*7>+1        ; Face Overlay Location
IconLoc         .EQU    MacIconLoc                      ; Disk Icon Location
OverLoc         .EQU    IconLoc+<LineLen*15>+1          ; '?' & 'X' Overlay Location
YaccMidDiff     .EQU    $00470040           ; delta point between lisa/mac screen
SoundLow        .EQU    $000BD600           ; low sound buffer address
PWMBuffer       .EQU    $00DCFFFE           ; PWM values are low 10 bits
SndBufWLen      .EQU    $0200               ; sound buffer word length
OvlyRAM         .EQU    $00700000           ; low RAM address when overlay is on
OvlyScreen      .EQU    $007B4000           ; top of screen with overlay with MMU
LoadBase        .EQU    $1000               ; where ROM image is loaded

STKbaseL        .EQU    $3FFE               ; Low word of Stack -- no overlay
STKbaseH        .EQU    $000B               ; High word of Stack -- no overlay
StackBase       .EQU    $000B3FFE           ; Start Stack at base of scrren
ROMslct         .EQU    $008C               ; High word of bank 0 ROM address
ROMStart        .EQU    $00090000           ; starting address of ROM code
ROMWSize        .EQU    $00008000           ; 32768 words in present ROM
ROMBSize        .EQU    $00010000           ; 65536 bytes in present ROM
MemLSize        .EQU    $00020000           ; memory contains 128K long words
MemWSize        .EQU    $00040000           ; 256K words
MemBSize        .EQU    $00080000           ; and 512K bytes

VidDmaReg       .EQU    $00EFFFFE           ; bits 16-1 for start of video buffer
SndDmaReg       .EQU    $00EEFFFE           ; bits 16-1 for start of sound buffer
MemMngBase      .EQU    $009FE800           ; base of 2K x 16 MMU registers
VidMapBase      .EQU    $00ADFE00           ; base of 256 x 16 video map
UserOffset      .EQU    $0                  ; offset to user mode registers
SuperOffset     .EQU    $1000               ; offset to supervisor mode registers
VectorSz        .EQU    $40                 ; $100 / 4 bytes per vector

.ENDC                                       ; of .if onYacc

; Device Equates for Lisa

.IF     onLisa
ScreenLow       .EQU    $660                ; ptr to address of top of Lisa screen
MidOffset       .EQU    $3FCA               ; offset to middle of Lisa screen
LisaMidDiff     .EQU    $00210068           ; delta point between lisa/mac screen
SoundLow        .EQU    $0007FD00           ; low sound buffer address
PWMBuffer       .EQU    $0007FD01           ; PWM bytes are low bytes
```

```
SndBufWLen      .EQU    $0172           ; sound/disk  buffer  word  length
OvlyRAM         .EQU    $00600000       ; low RAM address  when  overlay  is  on
OvlyScreen      .EQU    $0067A700       ; top  of  screen  with  overlay

ROMStart        .EQU    $00400000       ; starting  address  of  ROM  code
ROMWSize        .EQU    $00008000       ; 32768  words  in  present  ROM
ROMBSize        .EQU    $00010000       ; 65536  bytes  in  present  ROM
MemLSize        .EQU    $00020000       ; memory  contains  128K  long  words
MemWSize        .EQU    $00040000       ; 256K  words
MemBSize        .EQU    $00080000       ; and  512K  bytes
LineLen         .EQU    $5A             ; horizontal  screen  line  has  90  bytes
ScrnLSize       .EQU    $1FFE           ; there  are  8K-2  longs  in  the  screen
.ENDC                                   ; of  .if  onLisa

.IF     onMac
; VIA  (6522)
; Absolute  Addresses

VBase           .EQU    $EFE1FE         ; base  address
AVBufB          .EQU    VBase           ; buffer  B
AVBufA          .EQU    $EFFFFE         ; buffer  A
AVBufM          .EQU    AVBufB          ; buffer  with  mouse  button  bit
AVIFR           .EQU    $EFFBFE         ; interrupt  flag  register
AVIER           .EQU    $EFFDFE         ; interrupt  enable  register

; Offsets

VBufB           .EQU    512*0           ; BUFFER  B
VBufAH          .EQU    512*1           ; buffer  a  (with  handshake)  [ Dont  use!  ]
VDIRB           .EQU    512*2           ; DIRECTION  B
VDIRA           .EQU    512*3           ; DIRECTION  A
VT1C            .EQU    512*4           ; TIMER  1  COUNTER  (L.O.)
VT1CH           .EQU    512*5           ; timer  1  counter  (high  order)
VT1L            .EQU    512*6           ; TIMER  1  LATCH  (L.O.)
VT1LH           .EQU    512*7           ; timer  1  latch  (high  order)
VT2C            .EQU    512*8           ; TIMER  2  LATCH  (L.O.)
VT2CH           .EQU    512*9           ; timer  2  counter  (high  order)
VSR             .EQU    512*10          ; SHIFT  REGISTER
VACR            .EQU    512*11          ; AUX.  CONTROL  REG.
VPCR            .EQU    512*12          ; PERIPH.  CONTROL  REG.
VIFR            .EQU    512*13          ; INT.  FLAG  REG.
VIER            .EQU    512*14          ; INT.  ENABLE  REG.
VBufA           .EQU    512*15          ; BUFFER  A

VBufD           .EQU    VBufA           ; disk  head  select  buffer

; Buffer  A:

VAOut           .EQU    $7F             ; VBufA  output  bits
VAInit          .EQU    $7B             ; VBufA  initial  values  med.  volume
VSound          .EQU    $07             ; sound  volume  bits
VSndPg2         .EQU    3               ; select  sound  page  2  if  0
VOverlay        .EQU    4               ; overlay  bit  (overlay  when  1)
VHeadSel        .EQU    5
VPage2          .EQU    6               ; select  video  page  2  if  0
VSCCWrReq       .EQU    7               ; SCC  write/request  line

; Buffer  B:

VBOut           .EQU    $87             ; VBufB  output  bits
VBInit          .EQU    $07             ; VBufB  initial  values
RTCData         .EQU    0
```

```
RTCClk          .EQU    1
RTCEnb          .EQU    2                       ; enabled when 0
VSW             .EQU    3                       ; mouse switch (0 when down)
VX2             .EQU    4                       ; mouse X level
VY2             .EQU    5                       ; mouse Y level
VH4             .EQU    6                       ; horizontal sync
VSndEnb         .EQU    7                       ; /sound enable (reset when 1)
.ENDC

.IF     onYacc
; VIA (6522)
; Absolute Addresses

VBase           .EQU    $EDFF40                 ; base address of MAC compatible VIA
MiscViaBase     .EQU    $EDFF20                 ; base address of miscellaneuos VIA

; Offsets

VBufB           .EQU      2*0                   ; BUFFER B
VBufAH          .EQU      2*1                   ; buffer a (with handshake) [ Dont use! ]
VDIRB           .EQU      2*2                   ; DIRECTION B
VDIRA           .EQU      2*3                   ; DIRECTION A
VT1C            .EQU      2*4                   ; TIMER 1 COUNTER (L.O.)
VT1CH           .EQU      2*5                   ; timer 1 counter (high order)
VT1L            .EQU      2*6                   ; TIMER 1 LATCH (L.O.)
VT1LH           .EQU      2*7                   ; timer 1 latch (high order)
VT2C            .EQU      2*8                   ; TIMER 2 LATCH (L.O.)
VT2CH           .EQU      2*9                   ; timer 2 counter (high order)
VSR             .EQU      2*10                  ; SHIFT REGISTER
VACR            .EQU      2*11                  ; AUX. CONTROL REG.
VPCR            .EQU      2*12                  ; PERIPH. CONTROL REG.
VIFR            .EQU      2*13                  ; INT. FLAG REG.
VIER            .EQU      2*14                  ; INT. ENABLE REG.
VBufA           .EQU      2*15                  ; BUFFER A

AVBufB          .EQU    VBase                   ; buffer B
AVBufA          .EQU    VBase+VBufA             ; buffer A
AVBufM          .EQU    VBase                   ; buffer with mouse button bit
AVIFR           .EQU    VBase+VIFR              ; interrupt flag register
AVIER           .EQU    VBase+VIER              ; interrupt enable register
VBufD           .EQU    VBufA                   ; disk head select buffer

MVBufB          .EQU    MiscViaBase             ; buffer B
MVBufA          .EQU    MiscViaBase+VBufA       ; buffer A
MVIFR           .EQU    MiscViaBase+VIFR        ; interrupt flag register
MVIER           .EQU    MiscViaBase+VIER        ; interrupt enable register

; Buffer A:
;   MAC compatible VIA Port A

VAOut           .EQU    00110111B               ; $37-VBufA output bits
VAInit          .EQU    00100011B               ; $23-VBufA initial values med. volume
VSound          .EQU    00000111B               ; $07-sound volume bits         (out)
VVBlank         .EQU    3                       ; Vertical blank true when = 0  (in)
VOverlay        .EQU    4                       ; overlay bit (overlay when 1)  (out)
VHeadSel        .EQU    5                       ; Sony head select (side0 = 1)  (out)
VEvenScan       .EQU    6                       ; Even scan line true when = 0  (in)
VSCCWrReq       .EQU    7                       ; SCC write/request line        (in)

; Buffer B:
;   MAC compatible VIA Port B
```

```
VBOut           .EQU    10000111B           ; $87-VBufB  output bits
VBInit          .EQU    00000111B           ; $07-VBufB  initial values
RTCData         .EQU    0                   ; r/w parameter data                (bi)
RTCClk          .EQU    1                   ; used to latch data into chip      (out)
RTCEnb          .EQU    2                   ; enabled when 0                    (out)
VSW             .EQU    3                   ; mouse switch (0 when down)        (in)
VX2             .EQU    4                   ; mouse X level                     (in)
VY2             .EQU    5                   ; mouse Y level                     (in)
VH4             .EQU    6                   ; horizontal sync                   (in)
VSndEnb         .EQU    7                   ; /sound enable (reset when 1)      (out)

;   Micsellaneous  VIA Port A

MVAOut          .EQU    00000100B           ; $04-MVBufA  output bits
MVAInit         .EQU    00000100B           ; $04-Disable  TV genlock
MVPrIntr        .EQU    0                   ; Priam Interrupt true when = 1  (in)
MVPrParErr      .EQU    1                   ; Priam Parity Error true when = 1  (in)
MVTVEna         .EQU    2                   ; Ext TV Enable true when = 0    (out)
MVTVConn        .EQU    3                   ; Ext TV connected when = 0      (in)
MVIMId          .EQU    4                   ; IM Bus                         (bi)
MVIMClk         .EQU    5                   ; IM Bus                         (bi)
MVIMRst         .EQU    6                   ; IM Bus                         (bi)
MVIMD           .EQU    7                   ; IM Bus                         (bi)

;   Micsellaneous  VIA Port B

MVBOut          .EQU    10111100B           ; $BC-MVBufB  output bits
MVBInit         .EQU    00010000B           ; $10- VMap lookup & output enable
MVHSPPBusy      .EQU    0                   ; HighSpeed Parallel Port Busy   (in)
MVHSPPParErr    .EQU    1                   ; HSPP Parity Err                (in)
MVHSPPReset     .EQU    2                   ; HSPP Reset                     (out)
MVHSPPCmd       .EQU    3                   ; HSPP Command                   (out)
MVVMapAC        .EQU    4                   ; VideoMap  0 = R/W, 1= Lookup   (out)
MVVMapOE        .EQU    5                   ; VidMap Output Ena, 0 = enable  (out)
MVMTmrIn        .EQU    6                   ; timer in
MVMTmrOut       .EQU    7                   ; timer out

VMapOE_Hih      .EQU    $20                 ; byte mask for ORI & ANDI of output enable
VMapAC_Hih      .EQU    $10                 ; same for access control/lookup

.ENDC

; note:  CA1 = VSync true if 0    (in)
;        CA2 = 1 sec clock        (in)
;        CB1 = keyboard clock     (in)
;        CB1 = keyboard data      (bi)

;_____________________________________________________________________
;
; SCC SERIAL CHIP ADDRESSES

.IF     onYacc
SCCRBase        .EQU    $DEFFF8             ; SCC base read address
SCCWBase        .EQU    $DEFFF9             ; SCC base write address
SCCWrite        .EQU    $1                  ; general offset for write from read
.ENDC
.IF     onMac
SCCRBase        .EQU    $9FFFF8             ; SCC base read address
SCCWBase        .EQU    $BFFFF9             ; SCC base write address
SCCWrite        .EQU    $200001             ; general offset for write from read
.ENDC
.IF     onLisa
```

```
SCCRBase        .EQU    $FCD201         ; SCC base read address
SCCWBase        .EQU    $FCD201         ; SCC base write address (same)
SCCWrite        .EQU    $0              ; general offset for write from read
.ENDC

SCCData         .EQU    4               ; general offset for data from control

AData           .EQU    6               ; offset for A channel data
ACtl            .EQU    2               ; offset for A channel control
BData           .EQU    4               ; offset for B channel data
BCtl            .EQU    0               ; offset for B channel control

RxBF            .EQU    0               ; SCC receive buffer full
TxBE            .EQU    2               ; SCC transmit buffer empty

;_____________________________________________________________________
;
; DISK ADDRESS

.IF     onMac
DBase           .EQU    $DFE1FF         ; disk address base
DPh0L           .EQU    DBase           ; phase 0 low
DPh0H           .EQU    $DFE3FF         ; phase 0 high
DMtrOff         .EQU    $DFF1FF         ; IWM Motor off
DMtrOn          .EQU    $DFF3FF         ; IWM Motor on
DiskQ6L         .EQU    $DFF9FF         ; shift register
DiskQ6H         .EQU    $DFFBFF
DiskQ7L         .EQU    $DFFDFF
DiskQ7H         .EQU    $DFFFFF

Ph0L            .EQU    512*0           ; disk address offsets from base
Ph0H            .EQU    512*1
Ph1L            .EQU    512*2
Ph1H            .EQU    512*3
Ph2L            .EQU    512*4
Ph2H            .EQU    512*5
Ph3L            .EQU    512*6
Ph3H            .EQU    512*7

MtrOff          .EQU    512*8
MtrOn           .EQU    512*9
IntDrive        .EQU    512*10          ; enable internal drive address
ExtDrive        .EQU    512*11          ; enable external drive address
Q6L             .EQU    512*12
Q6H             .EQU    512*13
Q7L             .EQU    512*14
Q7H             .EQU    512*15
.ENDC                                   ; of .IF onMac

.IF     onYacc
DBase           .EQU    $DDFFE1         ; disk address base

Ph0L            .EQU    2*0             ; disk address offsets from base
Ph0H            .EQU    2*1
Ph1L            .EQU    2*2
Ph1H            .EQU    2*3
Ph2L            .EQU    2*4
Ph2H            .EQU    2*5
Ph3L            .EQU    2*6
Ph3H            .EQU    2*7

MtrOff          .EQU    2*8
```

```
MtrOn           .EQU    2*9
IntDrive        .EQU    2*10                ; enable internal drive address
ExtDrive        .EQU    2*11                ; enable external drive address
Q6L             .EQU    2*12
Q6H             .EQU    2*13
Q7L             .EQU    2*14
Q7H             .EQU    2*15

DPh0L           .EQU    DBase               ; phase 0 low
DPh0H           .EQU    DBase+Ph0H          ; phase 0 high
DMtrOff         .EQU    DBase+MtrOff        ; IWM Motor off
DMtrOn          .EQU    DBase+MtrOn         ; IWM Motor on
DiskQ6L         .EQU    DBase+Q6L           ; shift register
DiskQ6H         .EQU    DBase+Q6H
DiskQ7L         .EQU    DBase+Q7L
DiskQ7H         .EQU    DBase+Q7H

PWMDataReg      .EQU    $DCFFFE             ; PWM value (0..7 = low, 8..15 = high)
.ENDC                                       ; of .IF onYacc

.IF     onLisa
SonyStart       .EQU    $648                ; Sony driver entry point
IntDrive        .EQU    $FCC000             ; base address of internal drive
ExtDrive        .EQU    $FC0000             ; base address of external drive
SonyDone        .EQU    $4AE                ; Byte, Sony comand complete
.ENDC                                       ; of .IF onLisa

;_____________________________________________________________________
;
; PRIAM PORT COMMAND AND STATUS REGISTERS
; 2/22/85 fel

.IF     onYacc
PriamBase       .EQU    $B00001             ; Base addr for priam regs

PriamIO         .EQU    PriamBase+$50       ; Byte used for data I/O
PriamCmd        .EQU    PriamBase+$60       ; Set R2-R7, then write cmd
PriamSts        .EQU    PriamBase+$20       ; Status byte, read R2-R7

PriamP0         .EQU    PriamBase+$64       ; Regs: cmd params, results
PriamP1         .EQU    PriamBase+$66       ; Low order byte
PriamP2         .EQU    PriamBase+$68
PriamP3         .EQU    PriamBase+$6A
PriamP4         .EQU    PriamBase+$6C
PriamP5         .EQU    PriamBase+$6E
.ENDC

;_____________________________________________________________________
;
;  Interrupt Auto-vector and Exception Definitions
;_____________________________________________________________________

BusErrVct       .EQU    $08                 ; Bus Error Handler
LINE1010        .EQU    $28                 ; 1010 emulator trap (system routines)
LINE1111        .EQU    $2C                 ; 1111 emulator trap (reserved)
AUTOINT1        .EQU    $64                 ; level 1 auto-vector
AUTOINT2        .EQU    $68                 ; level 2 auto-vector
AUTOINT3        .EQU    $6C                 ; level 3 auto-vector
AUTOINT4        .EQU    $70                 ; level 4 auto-vector
AUTOINT5        .EQU    $74                 ; level 5 auto-vector
AUTOINT6        .EQU    $78                 ; level 6 auto-vector
AUTOINT7        .EQU    $7C                 ; level 7 auto-vector
```

```
;_____________________________________________________________
;
; Mouse Equates (From Rick's drivers)
;_____________________________________________________________
.IF onLisa
MouseX          .EQU    RomStart+ROMBSize+$00EC     ; location of mouse x coord <9/14
twm>
MouseY          .EQU    RomStart+ROMBSize+$00EE     ; location of mouse y coord <9/14
twm>
;_____________________________________________________________
;
; Clock/Calender Equates (conversion from Lisa <-> Mac Time stamp format)
;_____________________________________________________________

LisaOffset      .EQU    $5A39A80        ; F sec between 1/1/01 and 1/1/04
.ENDC
;_____________________________________________________________
;
; SYSCOM Equates (System Communication Area)
;_____________________________________________________________

SysCom          .EQU    $100            ; start of system communication area

; SYSTEM LOCATIONS

MonkeyLives     .EQU    $100            ; monkey lives if >= 0
ScrVRes         .EQU    $102            ; screen vertical resolution (dots/inch)
ScrHRes         .EQU    $104            ; screen horizontal resolution (dots/inch)
ScreenRow       .EQU    $106            ; rowBytes of Mac screen
MemTop          .EQU    $108            ; Ptr to top of memory
BufPtr          .EQU    $10C            ; Ptr to bottom of code buffer
StkLowPt        .EQU    $110            ; Lowest stack as measured in VBL task
HeapEnd         .EQU    $114            ; Ptr to end of heap
TheZone         .EQU    $118            ; Ptr to current heap zone
UTableBase      .EQU    $11C            ; Ptr to unit I/O table
MacJmp          .EQU    $120            ; Ptr to MACSBUG jumptable

DskRtnAdr       .EQU    $124            ; temp for disk driver
TwiggyVars      .EQU    $128            ; Ptr to twiggy driver locals
DskVerify       .EQU    $12C            ; used by sony driver for read/verify
LoadTrap        .EQU    $12D            ; set to non-zero to trap before pgm strt
mmInOK          .EQU    $12E            ; non-zero when initial mem mngr cks ok
DskWr11         .EQU    $12F            ; try 1-1 disk writes when non-zero
ApplLimit       .EQU    $130            ;

SonyVars        .EQU    ApplLimit+4     ; pointer to 3-1/2 disk driver vars
PWMValue        .EQU    SonyVars+4      ; current PWM value
PollStack       .EQU    PWMValue+2      ; SCC poll data start stack location
PollProc        .EQU    PollStack+4     ; proc which handles SCC poll data
DskErr          .EQU    PollProc+4      ; disk routine result code

SysEvtMask      .EQU    DskErr+2        ; System event mask
SysEvtBuf       .EQU    SysEvtMask+2    ; Ptr to system event queue element buffer
EventQueue      .EQU    SysEvtBuf+4     ; 5 words, event queue header
EvtBufCnt       .EQU    EventQueue+10   ; max number of events in SysEvtBuf - 1

RndSeed         .EQU    EvtBufCnt+2
SysVersion      .EQU    RndSeed+4       ; version # of RAM-based system
SEvtEnb         .EQU    SysVersion+2    ; 1 byte: 0 disables SysEvent calls from GNE
DSWndUpdate     .EQU    SEvtEnb+1       ; 1 byte: 0 flags GNE to paintBehind DS
AlertRect
```

```
FontFlag        .EQU    DSWndUpdate+1   ; boolean for font manager bug
Filler3         .EQU    FontFlag+1      ; 1 byte of filler

VBLQueue        .EQU    Filler3+1       ; 5 words, VBL queue header
Ticks           .EQU    VBLQueue+10     ; Tick count, time since boot in 1/60 increments
MBTicks         .EQU    Ticks+4         ; tick count when mouse button last changed
MBState         .EQU    MBTicks+4       ; current mouse button state
Tocks           .EQU    MBState+1       ; Lisa sub-tick count

KeyMap          .EQU    MBState+2       ; 2 longs, a bitmap of the keyboard
KeypadMap       .EQU    KeyMap+8        ; 1 long, a bitmap for numeric pad-18bits
KeyLast         .EQU    KeypadMap+8     ; lo byte: ASCII for last valid keycode
                                        ; hi byte: last valid keycode
                                        ; these are 0 when no key is down
KeyTime         .EQU    KeyLast+2       ; long tickcount when KEYLAST was rec'd
KeyRepTime      .EQU    KeyTime+4       ; long tickcount when key was last repeated
KeyThresh       .EQU    KeyRepTime+4    ; word containing threshold for repeat
KeyRepThresh    .EQU    KeyThresh+2     ; word containing repeat speed

Lvl1DT          .EQU    KeyRepThresh+2  ; Interrupt level 1 dispatch table
Lvl2DT          .EQU    Lvl1DT+32       ; Interrupt level 2 dispatch table

UnitNtryCnt     .EQU    Lvl2DT+32       ; count of entries in unit table
VIA             .EQU    UnitNtryCnt+2   ; VIA base address (use low memory addresses)
SCCRd           .EQU    VIA+4           ; SCC base read address (to save code space)
SCCWr           .EQU    SCCRd+4         ; SCC base write address
IWM             .EQU    SCCWr+4         ; IWM base address

GetParam        .EQU    IWM+4           ; Parameter blk for reading sys parameter area

SysParam        .EQU    GetParam+20     ; 20 bytes of system parameter area
SPValid         .EQU    SysParam        ; byte 1  = validation field ($A7)
SPOdometer      .EQU    SPValid+1       ;    2-4  = odometer
SPPortA         .EQU    SPOdometer+3    ;    5-6  = SCC port A configuration
SPPortB         .EQU    SPPortA+2       ;    7-8  = SCC port B configuration
SPAlarm         .EQU    SPPortB+2       ;    9-12 = alarm time
SPFont          .EQU    SPAlarm+4       ;   13-14 = default font id
SPKbd           .EQU    SPFont+2        ;      15 = kbd repeat thresh in 4/60ths(4)
                                        ;           kbd repeat rates in 2/60ths(4)
SPPrint         .EQU    SPKbd+1         ;      16 = print stuff
SPVolCtl        .EQU    SPPrint+1       ;      17 = volume control (low 3 bits)
SPClikCaret     .EQU    SPVolCtl+1      ;      18 = double time in 4/60ths(4)
                                        ;           caret blink time in 4/60ths(4)
SPMisc1         .EQU    SPClikCaret+1   ;      19 = english/metric (1),
                                        ;           extra (2),
                                        ;           country code (5)
SPMisc2         .EQU    SPMisc1+1       ;      20 = paranoia level (1), mouse
                                        ;           scaling(1), keyClick (1), boot
                                        ;           disk (1), menu flash (2), help
                                        ;           level (2)
Time            .EQU    SysParam+20     ; clock time when last read (extrapolated)
BootDrive       .EQU    Time+4          ; drive number of boot drive

JShell          .EQU    Time+6          ; used by journaling shell . . .
Filler3A        .EQU    JShell+2        ; 2 bytes

KbdVars         .EQU    Filler3A+2      ; Keyboard manager variables (4 bytes)
JKybdTask       .EQU    KbdVars+4       ; keyboard VBL task hook
KbdType         .EQU    JKybdTask+4     ; high byte holds keyboard model number
AlarmState      .EQU    KbdType+1       ; Bit7=parity, Bit6=beeped, Bit0=1=enabled

CurIOTrap       .EQU    KbdType+2       ; trap the caused current IO transaction
```

```
DiskVars        .EQU    CurIOTrap+2     ; Disk driver variables (62 bytes)

SdVolume        .EQU    DiskVars+62     ; Global volume control (1 byte)
SdEnable        .EQU    SdVolume+1      ; byte to enable 4 voice engine
SoundVars       .EQU    SdEnable+1      ; Sound driver variables (really only 20 bytes)

SoundPtr        .EQU    SoundVars       ;pointer to 4VE sound definition table
SoundBase       .EQU    SoundPtr+4      ;base address of sound bitMap
SoundVBL        .EQU    SoundBase+4     ;vertical retrace control element
SoundDCE        .EQU    SoundVBL+16     ;pointer to sound driver DCE
SoundActive     .EQU    SoundDCE+4      ;boolean specifying if sound is enabled
SoundLevel      .EQU    SoundActive+1   ;byte specifying current level in buffer
CurPitch        .EQU    SoundLevel+1    ;word holding current pitch value

SoundLast       .EQU    CurPitch+2      ;address past last sound variable

ScreenVars      .EQU    SoundVars+48    ; Screen driver variables

JGNEFilter      .EQU    ScreenVars+8    ; GetNextEvent filter proc

Key1Trans       .EQU    JGNEFilter+4    ; pointer to procedure handling keyboard
                                        ;  key translation
Key2Trans       .EQU    Key1Trans+4     ; pointer to procedure handling numeric
                                        ;  keypad key translation
SysZone         .EQU    Key2Trans+4     ; pointer to system heap zone
ApplZone        .EQU    SysZone+4       ; pointer to application heap zone
ROMBase         .EQU    ApplZone+4      ; start of ROM addresses (jump table?)
RAMBase         .EQU    ROMBase+4       ; start of RAM addresses
BasicGlob       .EQU    RAMBase+4       ; pointer to Basic globals
DSAlertTab      .EQU    BasicGlob+4     ; pointer to deep shit alerts
ExtStsDT        .EQU    DSAlertTab+4    ; SCC ext/sts secondary dispatch table
SCCASts         .EQU    ExtStsDT+16     ; SCC read reg 0 last ext/sts rupt - A
SCCBSts         .EQU    SCCASts+1       ; SCC read reg 0 last ext/sts rupt - B
SerialVars      .EQU    SCCBSts+1       ; async driver variables
FinderName      .EQU    SerialVars+16   ; 16 byte name of Finder
doubleTime      .EQU    FinderName+16   ; double click time(Set up by boot blocks)
caretTime       .EQU    doubleTime+4    ; caret blink time
ScrDmpEnb       .EQU    caretTime+4     ; screen dump enabled when non-zero
ScrDmpType      .EQU    ScrDmpEnb+1     ; FF dumps screen, FE dumps front window
TagData         .EQU    ScrDmpType+1    ; sector tag info for twiggy drivers
DrvQHdr         .EQU    TagData+14      ; queue header of drive numbers in system
PWMBuf2         .EQU    DrvQHdr+10      ; pointer to PWM buffer 1 (or 2 if sound
                                        ;  buffer 2 is being used)
HpChk           .EQU    PWMBuf2+4       ; Points to heap check RAM code
MaskBC          .EQU    HpChk+4         ; Memory Manager Byte Count Mask
MaskHandle      .EQU    MaskBC          ; Memory Manager Handle Mask
MaskPtr         .EQU    MaskBC          ; Memory Manager Pointer Mask
MinStack        .EQU    MaskBC+4        ; min stack size (1024) used in InitApplZone
DefltStack      .EQU    MinStack+4      ; default size of stack (for growing AZone)
mmDefFlags      .EQU    DefltStack+4    ; default zone flags
GZRootHnd       .EQU    mmDefFlags+2    ; root handle for GrowZone
GZRootPtr       .EQU    GZRootHnd+4     ; root pointer for GrowZone
GZMoveHnd       .EQU    GZRootPtr+4     ; moving handle for GrowZone
DSDrawProc      .EQU    GZMoveHnd+4     ; alternate deepshit box draw proc
EjectNotify     .EQU    DSDrawProc+4    ; proc to call and notify upon ejects
IAZNotify       .EQU    EjectNotify+4   ; proc to call and notify upon world swaps
endofvars       .EQU    IAZNotify+4     ; end of final defined vars

;_____________________________________________________________________
;
; The File System local variables are kept from $340 to $3FF
```

```
;_____________________________________________________________________

FileVars      .EQU     $340                ;
DSAlertRect   .EQU     $3F8                ; rectangle  for disk-switch  alert

;_____________________________________________________________________
;
;   MacWorks*/LisaBug   globals   $400-$7FF
;_____________________________________________________________________

;The following  globals  are declared  elsewhere  in sysequ/sonyutil/drivers
;
; SYSUTIL  name                    SONYUTIL  name    DRIVERS  name
;
; SonyDone        .EQU     $4AE     TWGDONE                          (byte)
;                 .EQU     $4AF     TWGSTAT                          (byte)
;                 .EQU     $610                     AltScrnAddr      (ptr)
; DskBase         .EQU     $620                                      (ptr)
;                 .EQU     $624     XDRIVE                           (ptr)
; SonyStart       .EQU     $648     DRVRENTRY
; ScreenLow       .EQU     $660                     ScrnAddr         (ptr)
;                 .EQU     $670                     AltScrnPhys      (ptr)
;                 .EQU     $674                     ScrnPhys         (ptr)
; DiskExt         .EQU     $744     TWGDON2                          (byte)
; DiskRslt        .EQU     $745     TWGSTA2                          (byte)
; WheresSony      .EQU     $746     EXTBASE
;                 .EQU     $798     CARDSID                          (3 longs?)
;                 .EQU     $7A4                     MemoryBase       (ptr)
;                 .EQU     $7B0                     KeybdQueue       (32 bytes)

.IF      onLisa
MacWorks          .EQU     $700
RegSave           .EQU     MacWorks          ;end of MacWorks  globals  E $7FF
SavePC            .EQU     RegSave+64        ;place deepshit  will stuff  registers
DiskExt           .EQU     SavePC+4          ;semaphore  for external  disk drive
DiskRslt          .EQU     DiskExt+1         ;error  code for external  disk drive
WheresSony        .EQU     DiskRslt+1        ;address  of external  Sony card
TopOfMem          .EQU     WheresSony+4      ;top of memory  for Mac environment
TermProc          .EQU     TopOfMem+4        ;pointer  to termination  routine
HDiskStart        .EQU     TermProc+4        ;pointer  to start of hard disk  driver
EndMW             .EQU     HDiskStart+4      ;last  MW var
.ENDC
;_____________________________________________________________________
;
;   For Macsbug  running  on the Lisa, the high memory  globals
;   from $410900  to approx.  $410B00  are used for the debugger.   The
;   debug  stack  grows  down  from $411000.(?  twm 9/13)
;_____________________________________________________________________

;_____________________________________________________________________
;
;   System  Core Routine  Dispatch  Table  resides  from $xxxx-$yyyy.   The user
;   can intercept  any core routine  by changing  these  pointers.
;_____________________________________________________________________
.IF      onLisa
DispatchTab       .EQU     RomStart+ROMBSize+$1000
.ENDC
.IF      onMac
DispatchTab       .EQU     $400
.ENDC
.IF      onYacc
DispatchTab       .EQU     $B3000       ; below  the start  of the screen
```

```
RegSave         .EQU    $B2FB0      ; high mem place to save registers
SavePC          .EQU    RegSave+64      ;place deepshit will stuff registers
.ENDC

;_____________________________________________________________
;
;   The cursor jump table interface and cursor state variables start at $800
;   For their definitions, see the file "GRAFEQU.TEXT". Starting at $8FC and
;   working down we have addresses of OS routines which are not accessed
;   through traps.
;_____________________________________________________________

WWExist         .EQU    $8F2            ; zero when window manager is initialized
QDExist         .EQU    $8F3            ; zero when quickdraw is initialized
JFetch          .EQU    $8F4            ; fetch a byte routine for I/O drivers
JStash          .EQU    $8F8            ; stash a byte routine for I/O drivers
JIODone         .EQU    $8FC            ; IODone entry location


LoadVars        .EQU    $900            ; loader variables

; low-memory Loader globals

CurApRefNum     .EQU    LoadVars            ; refNum of current resFile
LaunchFlag      .EQU    CurApRefNum+2       ; boolean to distinguish launch/chain
CurrentA5       .EQU    LaunchFlag+2        ; current value of A5
CurStackBase    .EQU    CurrentA5+4         ; current stack base
LoadFiller      .EQU    CurStackBase+4      ; empty for now...
CurApName       .EQU    LoadFiller+4        ; place to save name of application
SaveSegHandle   .EQU    CurApName+32        ; place to save seg 0 handle
CurJTOffset     .EQU    SaveSegHandle+4     ; word for current jump table offset
CurPageOption   .EQU    CurJTOffset+2       ; current page 2 configuration word
LoaderPBlock    .EQU    CurPageOption+4     ; param block for internal ExitToShell
LastLGlobal     .EQU    LoaderPBlock+10     ; address past last loader global

PrintVars       .EQU    LastLGlobal         ; 16 bytes for Owen
LastPGlobal     .EQU    PrintVars+16        ; address of last global
CoreEditVars    .EQU    LastPGlobal         ; 12 bytes for core edit

; Low memory vars for the scrap manager

scrapVars   .EQU        $960

scrapInfo   .EQU        scrapVars               ; scrap info
scrapHandle .EQU        scrapInfo+4             ; handle to memory scrap
scrapCount  .EQU        scrapHandle+4           ; validation byte
scrapState  .EQU        scrapCount+2            ; state of scrap defined below
scrapName   .EQU        scrapState+2            ; pointer to scrap name
scrapTag    .EQU        scrapName+4             ; actual name goes here
scrapEnd    .EQU        scrapTag+16             ; end of scrap vars


ToolVars        .EQU    $980            ; toolbox variables

;_____________________________________________________________
;
; Finally, we have the start of the heap.  Note that some important system
; tables are kept on the heap.
;
;_____________________________________________________________

HeapStart       .EQU    $0B00           ; start of the heap
```

```
;_____________________________________________________________________
;
; System Data Structures
;
;_____________________________________________________________________

Lock            .EQU    7           ; lock bit in a master pointer
Purge           .EQU    6           ; bit for purgeable/unpurgeable
Resource        .EQU    5           ; bit to flag a resource handle

; Queue Header Definition

QHeadSize       .EQU    10          ; A queue header is 10 bytes long

QFlags          .EQU    0           ; misc. flags
QHead           .EQU    2           ; 32-bit ptr to first element
QTail           .EQU    6           ; 32-bit ptr to last element
QInUse          .EQU    7           ; queue-in-use flag bit

; Queue Element Type Definitions

VType           .EQU    1           ; VBL queue element is type 1
IOQType         .EQU    2           ; I/O queue element is type 2
TimerType       .EQU    3           ; timer queue element is type 3
EvType          .EQU    4           ; event queue element is type 4
FSQType         .EQU    5           ; File System VCB element

; General Purpose Queue Element Definition

QLink           .EQU    0           ; link to next queue element
QType           .EQU    4           ; queue element type

; Drive queue element offsets

DQElLnth    .EQU    12              ; drive queue element length including hdr
;change to 14 except for sony driver!
DQDrive     .EQU    6               ; drive number
DQRefNum    .EQU    8               ; driver refnum
DQFSID      .EQU    10              ; file system handling this drive
DQDrvSize   .EQU    12              ; number of blocks this drive

; Unit Table Definition -- there is one entry in the unit table for each
; logical driver. The number of entries is hardwired to 16. Each entry consists
; of one 32-bit memory handle; the first entry corresponds to RefNum -1, the
; last to RefNum -32. The Unit Table is made during sysInit time and is
; allocated from permanent system memory; lo-memory location UTableBase points
; to the table. UnitNtryCnt is initialized to UnitEntries by StartInit: all
; other code should use this low mem variable.

; Device Control Entry Definition: for each Unit Table Entry, one driver may be
; installed. If the driver is installed, the corresponding memory handle in the
; unit table will be the handle of a Device Control Entry.

DCtlEntrySize   .EQU    40          ; each entry is 40 bytes long

DCtlDriver      .EQU    0           ; offset of pointer to driver
DCtlFlags       .EQU    4           ; offset of flags word
DCtlQueue       .EQU    6           ; offset of queue header
DCtlQHead       .EQU    8           ; offset of queue first-element pointer
DCtlQTail       .EQU    12          ; offset of queue last-element pointer
DCtlPosition    .EQU    16          ; offset of longword position pointer
```

```
DCtlStorage     .EQU    20          ; handle of driver's private storage
DCtlRefNum      .EQU    24          ; refNum of this driver
DCtlCurTicks    .EQU    26          ; long counter for timing systemTask calls
DCtlWindow      .EQU    30          ; pointer to driver's window (if any)
DCtlDelay       .EQU    34          ; word for number of ticks between sysTask calls
DCtlEMask       .EQU    36          ; word for desk ornament event mask
DCtlMenu        .EQU    38          ; word for menu associated with driver

; Vertical Blanking Control Block Queue Element

VBLINK          .EQU    0           ; LINK TO NEXT ELEMENT
VBLTYPE         .EQU    4           ; UNIQUE ID FOR VALIDITY CHECK
VBLADDR         .EQU    6           ; ADDRESS OF SERVICE ROUTINE
VBLCOUNT        .EQU    10          ; COUNT FIELD FOR TIMEOUT
VBLPHASE        .EQU    12          ; PHASE TO ALLOW SYNCHRONIZATION
;
INVBL           .EQU    6           ; bit index for "in VBL" flag


; Driver Format Definitions

DrvrFlags       .EQU     0          ; various flags and permissions
DrvrDelay       .EQU     2          ; word indicating # of ticks between systask calls
DrvrEMask       .EQU     4          ; word indicating what events to accept
DrvrMenu        .EQU     6          ; word indicating driver menu ID
DrvrOpen        .EQU     8          ; word offset to open routine
DrvrPrime       .EQU    10          ; word offset to prime routine
DrvrCtl         .EQU    12          ; word offset to control routine
DrvrStatus      .EQU    14          ; word offset to status routine
DrvrClose       .EQU    16          ; word offset to warmstart reset routine
DrvrName        .EQU    18          ; length byte and name of driver

; Driver Flags Word Bit Field Definitions

DReadEnable     .EQU    0           ; driver enabled for read operations
DWritEnable     .EQU    1           ; driver enabled for writing
DCtlEnable      .EQU    2           ; driver enabled for control operations
DStatEnable     .EQU    3           ; driver enabled for status operations
DNeedGoodBye    .EQU    4           ; driver needs a "goodbye kiss"
DNeedTime       .EQU    5           ; driver needs "main thread" time
DNeedLock       .EQU    6           ; driver needs to be accessed at interrupt level

; Run-Time Flags (in DCtlFlags+1)

DOpened         .EQU    5           ; flag to mark driver 'Open'
DRAMBased       .EQU    6           ; 1=RAM-based Driver, 0=ROM-based
DrvrActive      .EQU    7           ; flag to mark the driver active

; Desk Ornament Message Definitions

accEvent        .EQU    64          ; feedEvent message
accRun          .EQU    65          ; SystemTask "run" message
accCursor       .EQU    66          ; cursor message for topMost window
accMenu         .EQU    67          ; menu message
accCut          .EQU    68          ; cut message from system edit
accCopy         .EQU    69          ; copy message from system edit
accPaste        .EQU    70          ; paste message from system edit
accUndo         .EQU    71          ; undo message from system edit
accClear        .EQU    72          ; clear message from system edit


;  General system equates:
```

```
IOQElSize     .EQU     50          ; length of I/O parameter block

IOLink        .EQU     0           ; queue link in header
IOType        .EQU     4           ; type byte for safety check
IOTrap        .EQU     6           ; FS: the Trap
IOCmdAddr     .EQU     8           ; FS: address to dispatch to

IOCompletion  .EQU     12          ; long pointer to completion routine
IOResult      .EQU     16          ; IO result code (=D0 if sync call)

IOFileName    .EQU     18          ; file name pointer
IOVRefNum     .EQU     22          ; volume refnum
IODrvNum      .EQU     IOVRefNum   ; (used for Eject and MountVol)
IORefNum      .EQU     24          ; reference number for I/O operation

IOFileType    .EQU     26          ; specified along with FileName (byte)
IOPermssn     .EQU     27          ; Open: permissions (byte)

IONewName     .EQU     28          ; Rename: new name pointer (4)
IOLEOF        .EQU     28          ; GetEOF,SetEOF: logical end-of-file (4)
IOOwnBuf      .EQU     28          ; Open: optional ptr to locked 522-byte buffer
IONewType     .EQU     28          ; SetFileType: new type byte (1)

IOBuffer      .EQU     32          ; data buffer pointer
IOByteCount   .EQU     36          ; requested byte count
IOReqCount    .EQU     36          ; (equate used by file system Allocate)
IONumDone     .EQU     40          ; actual byte count completed
IOActCount    .EQU     40          ; (equate used by file system Allocate)

IOPosMode     .EQU     44          ; initial file positioning mode, eol char
IOPosOffset   .EQU     46          ; file position offset (long word)

NoQueueBit    .EQU     9           ; tells I/O system not to queue the request
AsynTrpBit    .EQU     10          ; bit in high byte of trap specifying async

; equates specific to GetFileInfo,SetFileInfo

IOFQElSize    .EQU     80          ; File command parameter block byte length
                                   ; 1st 28 bytes are the same as the general block
IOFDirIndex   .EQU     28          ; GetFileInfo - directory index
IOFlAttrib    .EQU     30          ; GetFileInfo - in-use bit=7, lock bit=0
IOFFlType     .EQU     31          ;  file type
IOFlUsrWds    .EQU     32          ; Get, SetFileInfo - user info (16 bytes)
IOFFlNum      .EQU     48          ; GetFileInfo - file number

IOFlStBlk     .EQU     52          ; GetFileInfo - Start file block (0000 if none)
IOFlLgLen     .EQU     54          ;  File logical length (EOF)
IOFlPyLen     .EQU     58          ;  File physical length in bytes.
IOFlRStBlk    .EQU     62          ;  Start file block, resource fork (0000 if none)
IOFlRLgLen    .EQU     64          ;  File logical length (EOF), resource fork
IOFlRPyLen    .EQU     68          ;  File physical length, resource fork

IOFlCrDat     .EQU     72          ;  File creation date & time (32 bits in seconds)
IOFlMdDat     .EQU     76          ;  last modification date & time (32 bits in seconds)

; equates specific to GetVolInfo,GetVolume,SetVolume,MountVol,UnmountVol,Eject

IOVQElSize    .EQU     64          ; Volume command parameter block byte length
IOVDrvNum     .EQU     IOVRefNum   ; 1st 28 bytes are the same as the general block
IOVNPtr       .EQU     IOFileName
IOVolIndex    .EQU     28          ; GetVol: volume index number
```

```
IOVCrDate     .EQU     30          ; GetVolInfo:  creation  date & time
IOVLsBkUp     .EQU     34          ; GetVolInfo:  last backup  date & time
IOVAtrb       .EQU     38          ; GetVolInfo:  Volume attributes
IOVNmFls      .EQU     40          ; GetVolInfo:  # files in directory
IOVDirSt      .EQU     42          ; GetVolInfo:  start block of file dir
IOVBlLn       .EQU     44          ; GetVolInfo:  length of dir in blocks
IOVNmAlBlks   .EQU     46          ; GetVolInfo:  num blks (of alloc size) this dev
IOVAlBlkSiz   .EQU     48          ; GetVolInfo:  alloc blk byte size
IOVClpSiz     .EQU     52          ; GetVolInfo:  bytes to try to allocate at a time
IOAlBlSt      .EQU     56          ; starting diskette (512-byte) block in block map
IOVNxtFNum    .EQU     58          ; GetVolInfo:  next free file number
IOVFrBlk      .EQU     62          ; GetVolInfo:  word of # free alloc blks for this vol

; I/O Command Equates for I/O Queue Elements (match trap numbers)

ARdCmd        .EQU     2           ; read command
AWrCmd        .EQU     3           ; write command
ACtlCmd       .EQU     4           ; control command
AStsCmd       .EQU     5           ; status command

; special offsets for the control and status core routines

CSCode        .EQU     26          ; a word for the control/status code
CSParam       .EQU     28          ; operation-defined parameters (20-bytes max)

; System Event Definitions

NullEvt           .EQU     0            ; event 0 is the null event
MButDwnEvt        .EQU     1            ; mouse button down is event 1
MButUpEvt         .EQU     2            ; mouse button up is event 2
KeyDwnEvt         .EQU     3            ; key down is event 3
KeyUpEvt          .EQU     4            ; key up is event 4
AutoKeyEvt        .EQU     5            ; auto-repeated key is event 5
UpdatEvt          .EQU     6            ; update event
DiskInsertEvt     .EQU     7            ; disk-inserted event
ActivateEvt       .EQU     8            ; activate/deactive event
AbortEvt          .EQU     9            ; abort?
NetWorkEvt        .EQU     10           ; network event
ReserveEvt        .EQU     10           ; reserved for system use
IODrvrEvt         .EQU     11           ; driver-defined event

App1Evt           .EQU     12           ; application defined events
App2Evt           .EQU     13
App3Evt           .EQU     14
App4Evt           .EQU     15

; Event Record Definition

EvtMax            .EQU     30           ; maximum number of events in buffer
EvtBlkSize        .EQU     16           ; size in bytes of the event record
EvtQBlkSize       .EQU     EvtBlkSize+6;  size of event record counting queue info

EvtNum            .EQU     0            ; word containing ID number of event
EvtMessage        .EQU     2            ; longword containing event-defined msg
EvtTicks          .EQU     6            ; longword holding TICKS when event occured
EvtMouse          .EQU     10           ; longword holding mouse pos when event occured
EvtMeta           .EQU     14           ; byte containing meta key flags
EvtMBut           .EQU     15           ; byte indicating state of mouse button

; system device constants
```

```
DskRfN          .EQU    $FFFB   ; disk's reference number
MouseCap        .EQU    6       ; Mouse button keycap number

; Control Call Codes

VerifyCode      .EQU    5       ; control call verify code
FormatCode      .EQU    6       ; control call format code
EjectCode       .EQU    7       ; control call eject code
KillCode        .EQU    1       ; KillIO code

; Status Call Codes

DrvStsCode      .EQU    8       ; status call code for drive status

; Resource Manager Equates

; High bits of the Locn long-word--the resource attributes:
; (Code optimization used frequently is a TST.B RAttr(A2) instead
; of BTST #ResSysRef, RAttr(A2). This saves a lot of bytes
; in RMgr)

ResSysRef       .EQU    7               ; reference to system/local reference
ResSysHeap      .EQU    6               ; In system/in application heap
ResPurgeable    .EQU    5               ; Purgeable/not purgeable
ResLocked       .EQU    4               ; Locked/not locked
ResProtected    .EQU    3               ; Protected/not protected
ResPreload      .EQU    2               ; Read in on openResource?
ResChanged      .EQU    1               ; Existing resource changed since last update
ResUser         .EQU    0               ; User defined resource attribute.

; When reading/writing, clear resChanged bit.

RCBMask         .EQU    $FD             ; ResChanged, byte mask AND.

; MAttr map attributes and masks

MapReadOnly     .EQU    7               ; is this file read-only?
MapCompact      .EQU    6               ; Is a compact necessary?
MapChanged      .EQU    5               ; Is it necessary to write map?

MCCMask         .EQU    $60             ; MapCompact + MapChanged
MChMask         .EQU    $20             ; MapChanged
MCoMask         .EQU    $40             ; MapCompact

; RDC
; Hard Disk driver equates

HDDrive     .EQU        4           ; default drive # for hard disk
HDRfNum     .EQU        $FFFE       ; refnum for hard disk driver
HDskID      .EQU        1           ; driver id
InitCode    .EQU        30          ; driver code for init controller
LisaDisk    .EQU        -99         ; error code for Lisa disk attached
; (RDC)
```