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

;-----------------------------------------------------------------
; System Constants
;-----------------------------------------------------------------
;
; Physical Device Equates

;   The YACC is a 1 meg machine with memory management that allows mapping of
;   the memory into 1 kilobyte pages.  There are 8 video planes which are 640*480
;   pixels in size.  This is 37.5K of ram per plane; the 1K resolution of the
;   pages force an allocation of 38K for each plane.  The Video DMA system uses
;   physical addresses 19 to 17 for the selection of the 8 planes, thus forcing
;   the division of the physical 1 meg of memory into 8 128K partitions.  There is
;   a 16 bit register that is loaded with a physical address that points to the
;   start of the frame buffer in each of the 8 planes.
;
;   The Sound subsystem makes all accesses from partition 0, which has addresses
;   17, 18, and 19 = 0.  It also has a 16 bit register which points to the start
;   of the sound buffer.  The system will fetch a 16 bit value at the start of
;   video scan line (15.734 KHz horizontal scan rate), of which the low 10 bits
;   are used as input to a PWM circuit.  The PWM circuit divides the line into 585
;   cycles; a value of 585 or higher will turn on the sound for the entire line
;   while a value of 0 will turn off the sound for the line.  The Sound DMA system
;   will continue to fetch successive words of data until a word is read that has
;   bit 15 = 1, whereupon the circuit will reset the starting address to the intial
;   value in the 16 bit register.
;
;   The floppy disk PWM circuit for speed control is two 8 bit registers/counters
;   that are written in parallel.  When 16 bits are written, the low 8 bits are
;   loaded into a PWM_Low register and the high 8 bits are laoded into a PWM_High
;   register.  The circuit provides Pulse Width Modulation and Pulse Frequency
;   Modulation.  Each register is fed into a counter that increments at an 8MHz
;   rate until it reachs 255, whereupon it halt and reload itself.  The other
;   register/counter is enabled and it follows an identical process.  The number
;   of times that both counters increment is proportional to the square-wave
;   frequency that is produced.  With a sum of counter increments equal to 256,
;   a square wave of frequency 31.25Khz is produced (1000/(256*(1000000/8MHz))).
;   The value in the PWM_Low register/counter divided by 256 is the proportion
;   of the square wave that will be low and the value of PWM_High divided by 256
;   is the proportion of time that will square wave will be high.
;
;   The memory management hardware consists of 2K x 16 of read/write registers.
;   8 megabytes can be mapped but only 1 megabyte can be mapped at any one time.
;   Any 2 accesses that has the low 19 bits the same and the high 3 bits are
;   different will cause a memory translation error.  To address 1 meg, bits
;   19-10 are provided by the MMU and 9-1 come directly from the address.  The
;   high 3 bits (22-20) are used by the MMU.  To write the MMU, the 16 bits are
;   mapped as follows:
;
;   Bit 15       - Referenced bit, = 1 if page has been referenced.
;   Bit 14       - Modified bit, = 1 if page has been written to.
;   Bit 13       - Vaild bit, = 1 if page is mapped to physical address
;   Bits 12:03   - Physical address.  This is the actual physical address used.
;   Bits 02:00   - Tag field. These map the 8 logical meg into 1 physical meg.
;
;   There are two maps; 1 is for access in supervisor mode and the 2nd is for
;   mapping of user mode accesses.
;
;   Below is the actual Physical memory layout of the 1 megabyte of memory,
;   along with the chosen  mapping  of the 128K partitions  for the actual  locations
;   of the Frame  Buffer,  the Sound  Buffer,  and CPU ram.
;
;  Address
; 19-18-17-16       --       0
;          ---------------
;  0 0 0   |     128K      |
;          |               |
;          ---------------                        -------------
;  0 0 1   |     128K      |                     /     |       |
;          |               |                    /      | 37.5K | - Frame
;          ---------------       -------------/        |       |   Buffer
;  0 1 0   |     128K      |     /     |       |       |       |
;          |               |    /      |  38K  |       |       |
;          ---------------     /       |       |       |       |
;  0 1 1   |     128K      |  /        |       |      ---------
;          |               | /        ---------\       |  0.5K | - Unused  or
;          ---------------/            |       | \     |       |   Sound  Buffer
;  1 0 0   |     128K      |           |  90K  |  \    |       |
;          |               |           |       |   -------------
;          ---------------\            |       |       |  1.0K | - CPU  ram
;  1 0 1   |     128K      | \         |       |      ---------
;          |               |  \        |       |       |  1.0K | - CPU  ram
;          ---------------     \       |       |      ---------
;  1 1 0   |     128K      |    \      |       |           .
;          |               |     \     |       |           .
;          ---------------        ------------\            .
;  1 1 1   |     128K      |                     \    ---------
;          |               |                      \    |  1.0K | - CPU  ram
;          ---------------                         -----------
;
;
;
;    With memory  management,  program RAM is mapped from 0 to $B3FFF  (8*90*1024).
;    The Video planes are mapped from $B4000  to $FFFFF  (8*38*1024).
;    The Sound buffer is mapped from $BD600  to $BD7FF  (512), into the unused
;    portion  of the first plane of the frame buffer.
;
;          ---------------
;          |      720K     |    -   $00000
;          |               |
;          |    CPU  Ram   |
;          |               |
;          |               |
;          |               |    -   $B3FFF
;          ---------------
;          |      304K     |    -   $B4000
;          |               |
;          | Frame  Buffer |    -   $FFFFF
;          ---------------
;          |      0.5K     |    -   $BD600
;          |               |
;          | Sound  Buffer |    -   $BD7FF
;          ---------------

ScreenLow       .EQU    $000B4000           ; top of screen  screen  address
ScreenOff       .EQU    $9800               ; offset  from one frame buffer  to next
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
```

```
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    $008C0000           ; starting address of ROM code
ROMWSize        .EQU    $00008000           ; 32768 words in present ROM
ROMBSize        .EQU    $00010000           ; 65536 bytes in present ROM
MemLSize        .EQU    $00040000           ; memory contains 256K long words
MemWSize        .EQU    $00080000           ; 512K words
MemBSize        .EQU    $00100000           ; and 1024K bytes
LineLen         .EQU    $50                 ; horizontal screen line has 80 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

; VIA (6522)
; Absolute Addresses

VBase           .EQU    $EDFF40             ; base address of MAC compatible VIA
MiscViaBase     .EQU    $EDFF20             ; miscellaneuos VIA
SlowViaBase     .EQU    $EDFF80             ; slow speed multipurpose 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

; Buffer A:
;   MAC compatible VIA Port A

VAOut           .EQU    00110111B           ; $37-VBufA output bits
VAInit          .EQU    00100011B           ; $33-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)

;   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)

;   Slow Speed Parallel  Port A & Port B

SSVAOut         .EQU    00000000B               ; $00 - All inputs
SSVAInit        .EQU    00000000B               ; $00 - data will be ignored
SSVBOut         .EQU    11111111B               ; $FF - All outputs
SSVBInit        .EQU    00000000B               ; $00 - null data
SSVia0          .EQU    0                       ; Bit 0 of port
SSVia1          .EQU    1
SSVia2          .EQU    2
SSVia3          .EQU    3
SSVia4          .EQU    4
SSVia5          .EQU    5
SSVia6          .EQU    6
SSVia7          .EQU    7

; Buffer B:
;   MAC compatible  VIA Port B

VBOut           .EQU    10000111B               ; $87-VBufB  output bits
VBInit          .EQU    10000111B               ; $87-VBufB  initial  values
RTCData         .EQU    0                       ; ??                             (bi)
RTCClk          .EQU    1                       ; ??                             (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 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

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

;______________________________________________________________________________
;
; SCC SERIAL  CHIP  ADDRESSES
```

```
SCCRBase          .EQU     $DEFFF8               ; SCC base read address
SCCWBase          .EQU     $DEFFF9               ; SCC base write address

SccChanA          .equ     $02
SccChanB          .equ     $00
SccData           .equ     $04
SccCntrl          .equ     $00

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

Scc00             .EQU     0                     ; equates for 16 registers
Scc01             .EQU     1
Scc02             .EQU     2
Scc03             .EQU     3
Scc04             .EQU     4
Scc05             .EQU     5
Scc06             .EQU     6
Scc07             .EQU     7
Scc08             .EQU     8
Scc09             .EQU     9
Scc10             .EQU     10
Scc11             .EQU     11
Scc12             .EQU     12
Scc13             .EQU     13
Scc14             .EQU     14
Scc15             .EQU     15

SccReset          .EQU     $C0                   ; WR9 <== 11000000B

;______________________________________________________________________
;
; DISK ADDRESS

.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

PWMData           .EQU     $DCFFFE               ; PWM value (0..7 = low, 8..15 = high)
```