

```

F800: 0029 38 BASH EQU $29
F800: 002A 39 BAS2L EQU $2A ;temp base for scrolling
F800: 002B 40 BAS2H EQU $2B
F800: 002C 41 H2 EQU $2C ;temp for lo-res graphics
F800: 002C 42 LMNEM EQU $2C ;temp for mnemonic decoding
F800: 002D 43 V2 EQU $2D ;temp for lo-res graphics
F800: 002D 44 RMNEM EQU $2D ;temp for mnemonic decoding
F800: 002E 45 MASK EQU $2E ;color mask for lo-res gr.
F800: 002E 46 CHKSUM EQU $2E ;temp for opcode decode
F800: 002E 47 FORMAT EQU $2E ;temp for opcode decode
F800: 002F 48 LASTIN EQU $2F ;temp for tape read csun
F800: 002F 49 LENGTH EQU $2F ;temp for opcode decode
F800: 0030 50 COLOR EQU $30 ;color for lo-res graphics
F800: 0031 51 MODE EQU $31 ;Monitor mode
F800: 0032 52 INVFLG EQU $32 ;normal/inverse(/flash)
F800: 0033 53 PROMPT EQU $33 ;prompt character
F800: 0034 54 YSAV EQU $34 ;position in Monitor command
F800: 0035 55 YSAV1 EQU $35 ;temp for Y register
F800: 0036 56 CSWL EQU $36 ;character output hook
F800: 0037 57 CSWH EQU $37 ;character input hook
F800: 0038 58 KSWL EQU $38
F800: 0039 59 KSWH EQU $39
F800: 003A 60 PCL EQU $3A ;temp for program counter
F800: 003B 61 PCH EQU $3B
F800: 003C 62 A1L EQU $3C ;A1-A5 are Monitor temps
F800: 003D 63 A1H EQU $3D
F800: 003E 64 A2L EQU $3E
F800: 003F 65 A2H EQU $3F
F800: 0040 66 A3L EQU $40
F800: 0041 67 A3H EQU $41
F800: 0042 68 A4L EQU $42
F800: 0043 69 A4H EQU $43
F800: 0044 70 A5L EQU $44
F800: 0044 71 MACSTAT EQU $44 ;machine state for break
F800: 0045 72 A5H EQU $45
F800: 0045 73 ACC EQU $45 ;Acc after break (destroys A5H)
F800: 0046 74 XREG EQU $46 ;X reg after break
F800: 0047 75 YREG EQU $47 ;Y reg after break
F800: 0048 76 STATUS EQU $48 ;P reg after break
F800: 0049 77 SPNT EQU $49 ;SP after break
F800: 004E 78 RNDL EQU $4E ;random counter low
F800: 004F 79 RNDH EQU $4F ;random counter high
F800: 80 *
F800: 0095 81 PICK EQU $95 ;CONTROL-U character
F800: 82 *
F800: 0200 83 IN EQU $0200 ;input buffer for GETLN
F800: 84 *
F800: 85 * Page 3 vectors
F800: 86 *
F800: 03F0 87 BRKV EQU $03F0 ;vectors here after break
F800: 03F2 88 SOFTEV EQU $03F2 ;vector for warm start
F800: 03F4 89 PWREDUP EQU $03F4 ;THIS MUST = EOR #$A5 OF SOFTEV+1
F800: 03F5 90 AMPERV EQU $03F5 ;APPLESOFT & EXIT VECTOR
F800: 03F8 91 USRADR EQU $03F8 ;Applesoft USR function vector

```

```

F800: 03FB 92 NMI EQU $03FB ;NMI vector
F800: 03FE 93 IRQLOC EQU $03FE ;Maskable interrupt vector
F800: 94 *
F800: 0400 95 LINE1 EQU $0400 ;first line of text screen
F800: 07F8 96 MSL0T EQU $07F8 ;current user of $C8 space
F800: 97 *
F800: 0000 98 * DO TEST
F800: 99 * ELSE
F800: C000 100 IOADR EQU $C000
F800: 101 * FIN
F800: 102 *
F800: C000 103 KBD EQU $C000
F800: C006 104 SLOTCXROM EQU $C006 ;enable slots 1-7
F800: C007 105 INTCXROM EQU $C007 ;swap out slots for firmware
F800: C010 106 KBDSTRB EQU $C010
F800: C01F 107 RD80VID EQU $C01F
F800: C020 108 TAPEOUT EQU $C020
F800: C030 109 SPKR EQU $C030
F800: C050 110 TXTCLR EQU $C050
F800: C051 111 TXTSET EQU $C051
F800: C052 112 MIXCLR EQU $C052
F800: C053 113 MIXSET EQU $C053
F800: C054 114 LOWSCR EQU $C054
F800: C055 115 HISCR EQU $C055
F800: C056 116 LORES EQU $C056
F800: C057 117 HIRFS EQU $C057
F800: C058 118 SETANO EQU $C058
F800: C059 119 CLRANO EQU $C059
F800: C05A 120 SETANI EQU $C05A
F800: C05B 121 CLRANI EQU $C05B
F800: C05C 122 SETAN2 EQU $C05C
F800: C05D 123 CLRAN2 EQU $C05D
F800: C05E 124 SETAN3 EQU $C05E
F800: C05F 125 CLRAN3 EQU $C05F
F800: C060 126 TAPEIN EQU $C060
F800: C064 127 PADDLO EQU $C064
F800: C070 128 PTRIG EQU $C070
F800: 129 *
F800: C3FA 130 IRQ EQU $C3FA ;IRQ entry in $C3 page
F800: C47C 131 IRQFIX EQU $C47C ;Restore state at IRQ
F800: 132 *
F800: C567 133 XHEADER EQU $C567
F800: C5D1 134 XREAD EQU $C5D1
F800: C5AA 135 WRITE2 EQU $C5AA
F800: 136 *
F800: CFFF 137 CLRROM EQU $CFFF
F800: E000 138 BASIC EQU $E000
F800: E003 139 BASIC2 EQU $E003
F800: 140 *
F800: 4A 141 PLOT LSR A ;Y-COORD/2
F801:08 142 PHP ;SAVE LSB IN CARRY
F802:20 47 F8 143 JSR GBASCALC ;CALC BASE ADR IN GBASL,H
F805:28 144 PLP ;RESTORE LSB FROM CARRY
F806:A9 0F 145 LDA #SOF ;MASK $0F IF EVEN

```