

```

C300: 19      INCLUDE C3SPACE
C300: 1 *****
C300: 2 *
C300: 3 * THIS IS THE $3XX ROM SPACE:
C300: 4 * Note: This page must not be used by any routines
C300: 5 * called by the F8 ROM. When it is referenced, it claims
C300: 6 * the C800 space (kicking out anyone who was using it).
C300: 7 * This also means that peripheral cards cannot use the AUXMOVE
C300: 8 * and XFER routines from their C800 space.
C300: 9 *
C300: 10 *****
C300: 11 CNO0 EQU *
C300: 12 BASICINT EQU *
C300: 13      BIT SEV      ;set vflag (init)
C300: 14      BVS BASICINT ;(ALWAYS TAKEN)
C300: 15 *
C300: 16 * BASIC input entry point. After a PR#3, this is the
C300: 17 * address that is called to input each character.
C300: 18 *
C300: 19 BASICIN EQU *
C300: 20      SEC
C300: 21      DFB $90      ;BCC OPCODE (NEVER TAKEN)
C300: 22 *
C300: 23 * BASIC output entry point: After a PR#3, this is the
C300: 24 * address that is called to output each character.
C300: 25 *
C300: 26 BASICOUT EQU *
C300: 27      CLC
C300: 28      CLV      ;CLEAR VFLAG (NOT INIT)
C300: 29      BVC BASICINT ;(ALWAYS TAKEN)
C300: 30 *
C300: 31 * Pascal 1.1 Firmware Protocol table:
C300: 32 *
C300: 33 * This tables identifies this as an Apple //e 80 column
C300: 34 * card. It points to the four routines available to
C300: 35 * programs doing I/O using the Pascal 1.1 Firmware
C300: 36 * Protocol.
C300: 37 *
C300: 38      DFB $01      ;GENERIC SIGNATURE BYTE
C300: 39      DFB $88      ;DEVICE SIGNATURE BYTE
C300: 40 *
C300: 41      DFB #>JPINIT ;PASCAL INIT
C300: 42      DFB #>JPREAD ;PASCAL READ
C300: 43      DFB #>JPWRITE ;PASCAL WRITE
C300: 44      DFB #>JPSTAT ;PASCAL STATUS
C300: 45 *****
C300: 46 *
C300: 47 * 128K SUPPORT ROUTINE ENTRIES:
C300: 48 *
C300: 49      JMP MOVE      ;MEMORY MOVE ACROSS BANKS
C300: 50      JMP XFER      ;TRANSFER ACROSS BANKS
C300: 51 *****
C300: 52 *
C300: 53 BASICENT STA CHAR

```

```

C31A:98 54      TYA      ; AND Y
C31B:48 55      PHA
C31C:8A 56      TXA      ; AND X
C31D:48 57      PHA
C31E:08 58      PHP      ;SAVE CARRY & VFLAG
C31F: 59 *
C31F: 60 * If escape mode is allowed, the high bit of MSLOT is
C31F: 61 * clear. Set M.CTL to flag that 1) escapes are allowed, and
C31F: 62 * 2) that control characters should not be echoed.
C31F: 63 * M.CTL is cleared by BPRINT.
C31F: 64 *
C31F:AD FB 04 65      LDA MODE      ;else esc enable, ctl disable
C322:2C F8 07 66      BIT MSLOT      ;get MSLOT
C325:30 05 C32C 67      BMI NOGETLN      ;=>Esc disable, ctl char enable
C327:09 08 68      ORA #M.CTL
C329:8D FB 04 69      STA MODE
C32C: 70 *
C32C: 71 NOGETLN EQU *
C32C:20 6D C3 72      JSR SETC8      ;SETUP C8 INDICATOR
C32F:28 73      PLP      ;GET VFLAG (INIT)
C330:70 15 C347 74      BVS JBASINIT ;=>DO THE INIT
C332: 75 *
C332: 76 * If a PR#0 has been done, input should be transferred
C332: 77 * from the video firmware to KEYIN. This is detected
C332: 78 * if the high bit of the mode byte is set.
C332: 79 *
C332:90 10 C344 80      BCC JC8      ;=>output, no problem
C334:AA 81      TAX      ;test mode
C335:10 0D C344 82      BPL JC8      ;video firmware is on
C337:20 5B CD 83      JSR SETKEYIN ;else set FDI8 as input
C33A:68 84      PLA      ;restore registers
C33B:AA 85      TAX
C33C:68 86      PLA
C33D:A8 87      TAY
C33E:AD 7B 06 88      LDA CHAR
C341:6C 38 00 89      JMP (KSWL) ;go input the character
C344: 90 *
C344:4C 7C C8 91 JC8      JMP C8BASIC ;GET OUT OF CN SPACE
C347:4C 03 C8 92 JBASINIT JMP BASICINIT ;=>GOTO C8 SPACE
C34A: 93 *
C34A: 94 JPINIT EQU *
C34A:20 6D C3 95      JSR SETC8      ;SETUP C8 INDICATOR
C34D:4C B4 C9 96      JMP PINIT ;XFER TO PASCAL INIT
C350: 97 JPREAD EQU *
C350:20 6D C3 98      JSR SETC8      ;SETUP C8 INDICATOR
C353:4C D6 C9 99      JMP PREAD ;XFER TO PASCAL READ
C356: 100 JPWRITE EQU *
C356:20 6D C3 101      JSR SETC8      ;SETUP C8 INDICATOR
C359:4C F0 C9 102      JMP PWRITE ;XFER TO PASCAL WRITE
C35C: 103 *
C35C:AA 104 JPSTAT TAX      ;is request code = 0?
C35D:F0 08 C367 105      BEQ PIORDY      ;=>yes, ready for output
C35F:CA 106      DEX      ;check for any input
C360:D0 07 C369 107      BNE PSTERR ;=>bad request, return error

```