

```

FE89:      515 *
FE89:A9 00 516 SETKBD LDA #S00      ;DO 'IN#0'
FE8B:85 3E 517 INPORT STA A2L      ;DO 'IN#AREG'
FE8D:A2 38 518 INPRT LDX #KSWL
FE8F:A0 18 519 LDY #KEYIN
FE91:DO 08 FE9B 520 BNE IOPRT
FE93:      521 *
FE93:A9 00 522 SETVID LDA #S00      ;DO 'PR#0'
FE95:85 3E 523 OUTPORT STA A2L      ;DO 'PR#AREG'
FE97:A2 36 524 OUTPRT LDX #CSWL
FE99:A0 F0 525 LDY #COUT1
FE9B:A5 3E 526 IOPRT LDA A2L      ;SET INPUT/OUTPUT VECTORS
FE9D:29 0F 527 AND #0F
FE9F:F0 04 FE95 528 BEQ IOPRT1
FEA1:09 C0 529 ORA #CIOADR
FEA3:A0 00 530 LDY #S00
FEA5:94 00 531 IOPRT1 STY LOCO,X      ;save low byte of hook
FEA7:95 01 532 STA LOCI,X      ;save acc
FEA9:A0 0E 533 LDY #SE      ;code=PR#/IN#
FEAB:4C B4 FB 534 GOTOXCX JMP GOTOXCX ;perform call
FEAE:      535 *
FEAE:EA 536 NOP
FEAF:00 537 CKSUMFIF DFB 0      ;/BRA0981
FEB0:      538 * ;-->CORRECT CKSUM AT CREATE TIME.
FEB0:4C 00 E0 539 XBASIC JMP BASIC      ;TO BASIC, COLD START
FEB3:4C 03 E0 540 BASCONT JMP BASIC2      ;TO BASIC, WARM START
FEB6:20 75 FE 541 GO JSR AIPC      ;ADDR TO PC IF SPECIFIED
FEB9:20 3F FF 542 JSR RESTORE      ;RESTORE FAKE REGISTERS
FEBB:6C 3A 00 543 JMP (PCL)      ;AND GO!
FEBF:4C D7 FA 544 REGZ JMP REGDSP      ;GO DISPLAY REGISTERS
FEC2:60 545 TRACE RTS      ;TRACE IS GONE
FEC3:EA 546 NOP
FEC4:60 547 STEP2 RTS      ;STEP IS GONE
FEC5:      548 *
FEC5:      549 * Return here from GOTOXCX
FEC5:      550 *
FEC5:      551 * NOTE: This address is hard-coded in BFUNC of the
FEC5:      552 * video firmware
FEC5:      553 *
FEC5:8D 06 C0 554 RETCX1 STA SFTSLOTXCXROM ;restore bank
FEC8:60 555 RETCX2 RTS      ;simply return
FEC9:EA 556 NOP
FECA:      557 *
FECA:4C FB 03 558 USR JMP USRADR      ;JUMP TO CONTROL-Y VECTOR IN RAM
FECD:      559 *
FECD:A9 40 560 WRITE LDA #$40
FECD:8D 07 C0 561 WRT2 STA SETINTCXROM ;set internal ROM
FED2:20 AA C5 562 JSR WRITE2      ;write to tape
FED5:F0 2C FF03 563 BEQ RD2      ;=>always set slots, beep
FED7:      564 *
FED7:      565 * SEARCH is called with a Monitor command of the form
FED7:      566 * HHL,LCADR1.ADR2 in which ADR1 < ADR2 and LL precedes HH
FED7:      567 * in memory. If HH is 0, or omitted (LL,LCADR1.ADR2), then
FED7:      568 * the single byte LL is searched for. You cannot search for

```

```

FED7:      569 * a two byte pair with a high byte of 0. A list of all
FED7:      570 * addresses containing the specified pattern is displayed.
FED7:      571 *
FED7:A0 01 572 SEARCH LDY #1      ;set Y to 1
FED9:A5 43 573 LDA A4H      ;is high byte 0?
FEDB:F0 04 FE01 574 BEQ SRCH1      ;=>yes, only look for low byte
FEDD:D1 3C 575 CMP (A1L),Y      ;check high byte first
FEDF:DO 0A FE0B 576 BNE SRCH2      ;=>no match, try next byte
FE01:88 577 SRCH1 DEY      ;match, now check low byte
FE02:A5 42 578 LDA A4L      ;get low byte
FE04:D1 3C 579 CMP (A1L),Y      ;does it match?
FE06:DO 03 FE0B 580 BNE SRCH2      ;=>no match, try next byte
FE08:20 92 FD 581 JSR PRAI      ;bytes match, print address
FE0B:20 BA FC 582 SRCH2 JSR NXTAL      ;increment address
FE0E:90 E7 FED7 583 BCC SEARCH      ;set Y back to 1
FE10:60 584 RTS
FE11:      585 *
FE11:A0 0D 586 MINI LDY #SD      ;dispatch mini-assembler call to
FE13:20 B4 FB 587 JSR GOTOXCX      ;get internal ROM switched in
FE16:      588 *
FE16:20 00 FE 589 CRMON JSR BLI      ;HANDLE CR AS BLANK
FE1F:68 590 PLA      ; THEN POP STACK
FE2A:68 591 PLA      ; AND RETURN TO MON
FE2B:DO 6C FF69 592 BNE MONZ      ;(ALWAYS)
FE2D:      593 *
FE2D:8D 07 C0 594 READ STA SETINTCXROM ;set internal ROM
FF00:20 D1 C5 595 JSR XREAD      ;do tape read
FF03:8D 06 C0 596 RD2 STA SETSLOTXCXROM ;restore slot CX
FF06:F0 32 FF3A 597 BEQ BELL      ;read (write) ok, beep
FF08:DO 23 FF2D 598 BNE PRERR      ;error, print message
FF0A:      599 *
FF0A:C1 F0 F0 EC 600 TITLE ASC "Apple" //e"
FF13:      601 *
FF13:      602 * NNBL gets the next non-blank for the mini-assembler
FF13:      603 *
FF13:20 FD FC 604 NNBL JSR UPMON      ;get char, upshift, INY
FF16:C9 A0 605 CMP #SA0      ;is it blank?
FF18:F0 F9 FF13 606 BEQ NNBL      ;yes, keep looking
FF1A:60 607 RTS
FF1B:      608 *
FF1B:80 6D FF8A 609 LOOKASC BCS DIG      ;it was a digit
FF1D:C9 A0 610 CMP #SA0      ;check for quote ('')
FF1F:DO 28 FF49 611 BNE RTS6      ;nope, return char
FF21:B9 00 02 612 LDA $200,Y      ;else get next char
FF24:A2 07 613 LDX #7      ;for shifting asc into A2L and A2H
FF26:C9 8D 614 CMP #S8D      ;was it CR?
FF28:F0 7D FFA7 615 BEQ GETNUM      ;yes, go handle CR
FF2A:C8 616 INY      ;advance index
FF2B:DO 63 FF90 617 BNE NXTBIT      ;=>(always) into A2L and A2H
FF2D:      618 *
FF2D:A9 C5 619 PRERR LDA #SC5      ;PRINT 'ERR', THEN FALL INTO
FF2F:20 ED FD 620 JSR COUT      ; FWEEPER.
FF32:A9 D2 621 LDA #SD2
FF34:20 ED FD 622 JSR COUT

```