

# Apple IIe Supplement to What's Where in the Apple

## A.1 Overview

The latest Apple II, called the *"/e/* for "enhanced", has several features added that make it more standard and versatile. The keyboard has been improved and will now generate all 128 ASCII key codes, including screen display of lower case. The RESET key now requires pressing the CONTROL key simultaneously and rebooting can be accomplished by pressing CTRL-OPEN APPLE-RESET, saving wear and tear on the on/off switch, always a weak point. A CTRL-CLOSED APPLE RESET initiates a built-in self-test. The screen display has been improved to allow either 40 or 80 column display under software control. There is also a full cursor control in all four directions. The 16K language card has been made a built-in feature and slot 0 has been eliminated. International versions are available for European and Asian buyers with switchable character sets.

Despite all these additional features, compatibility was kept with most of the previous software. All of the standard monitor entry points were preserved so that, unless software uses undocumented monitor entries, it should run on the *"/e/*. The only other problem that might arise is the utilization of one formerly unused page zero location. A program that used that location will probably not function properly on the new Apple.

Another new feature is the addition of a 64K expansion available as an enhanced 80 column card, which will make additional memory available to sophisticated programs such as Visicalc.

## A.2 A Third Apple Monitor

There is now a third major version of the Apple monitor to go along with the Auto-Start and (old) System monitors. While all of the documented entry points remain the same, most of the routines jump to the new ROM in the \$C100-\$CFFF range. These new routines check on the availability and status of 80 column and

extended 80 column cards, and use this additional hardware for enhanced displays and cursor control, when available.

The major differences between the II+ and the *"/e/* are as follows:

a) RESET, OPEN APPLE and CLOSED APPLE keys: The Control key must now be pressed to initiate the RESET cycle. This will eliminate accidental RESETs as the keys are on opposite sides of the keyboard. The APPLE keys are paddle button extensions to the keyboard and can be used in conjunction with the RESET cycle to initiate the self diagnostic tests [CLOSED] or power-on reboot [OPEN].

b) EDITING: In addition to the I, J, K, and L diamond cursor control pattern, there are four arrow keys that can also be used to move the cursor on the screen. Pressing ESC to enter the editing mode changes the cursor to an inverse "+" to indicate editing mode. Additional commands are also available. ESC-R enters upper-case restrict mode, which allows only upper-case letters during keyboard entry except after typing a "'", when both upper and lower case are allowed for PRINT statement. Typing another "'" returns to upper-case only. ESC-T exits this mode. ESC-4 displays a 40 column screen similar to the II+, while ESC-8 shifts to the new 80 column screen display. ESC CTRL-Q exits the new mode entirely, returning to the old 40 column display, and turning off the 80 column card.

## A.3 The New Display

In order to maintain compatibility with the old II and II+, it was necessary to design a screen display that utilized the old screen memory (\$400-\$7FF). This was insufficient for 80 column display, so Apple designed an 80 column card with its own memory mapped into the same addresses. The hardware alternates its scans from one set of memory to the other when in 80 column mode. Characters are stored alternating from one address to the next, with all the odd screen locations in main memory and all the even ones on the auxiliary card.

There are routines in the new monitor areas that can convert an 80 column screen to 40 by moving the alternate characters to the main board and throwing away the last 40 characters in each column. The opposite switch is accomplished by a similar move to the auxiliary card, using only the leftmost 40 columns for the characters previously on the screen.

#### A.4

##### *Hardware Locations*

On the older Apples, the addresses \$C000-\$C00F were equivalent addresses and were only partly decoded by the hardware. This meant that reading any of those would yield the same result [reading the keyboard], which was also true of \$C010-\$C01F [clearing the keyboard strobe]. These addresses are now fully decoded and provide a set of soft switches/status indicators for the new 80 column card and extended 80 column card [with 64K memory expansion].

The switches include options to read and/or write either the main board locations or the auxiliary card locations, to set the standard zero page and system stack [main board] or the alternate zero page and system stack [auxiliary card], to turn on or off the \$CX00 ROMs, to enable or disable the 80 column display, and to turn on the normal or alternate character sets [normal has upper case flash instead of lower case inverse].

Additionally, there are a group of locations that can be read to determine the current switch settings so that any program changing the switches can save the current settings and restore them at the end. States that can be determined include READ/WRITE status, language card bank status, 80 column status, page status, and text mode.

#### A.5

##### *Software Status*

Apple has always reserved some unused locations in the text page RAM as scratch memory for the 7 hardware slots [1-7]. Several of these locations are now permanently assigned to the new 80 column cards, when they are in use, and are used to store the current cursor location, I/O status, and BASL/BASH in Pascal.

One particular location [\$4FB] is the software MODE status. Each bit is indicative of the current state of operations: BASIC/Pascal, interrupts set/cleared, Pascal 1.0/1.1, normal/inverse video, GOTOXY in progress/not in progress, upper case restrict/literal mode, BASIC input/print, and ESC-R active/inactive.

These locations enable a program to determine the current state of the machine more easily than before, and make it simpler to utilize the new hardware configurations in programming.

#### A.6

##### *Programming Considerations*

The standard Applesoft GET and INPUT [and associated monitor routine KEYIN] were not designed to work with an 80 column display and using them while in 80 column mode can cause loss of data or erasure of program in memory, but this can be overcome by a routine explained in Appendix E of the new Applesoft Tutorial. Reading the keyboard directly [\$C000] functions the same as before.

Do not assume an Apple //e or 80 column card when writing programs; one of the first routines should check for the type of machine being used. Apple supplies a program that will do this on "The Applesoft Sampler"; and Call A.P.P.L.E. has also published a routine for this purpose. HTAB will not function beyond the 40th column. While POKE 36,POS works most of the time, Apple recommends POKE 1403,POS [0-79] for the //e. This routine will not work at all for an old Apple.

It is the programmer's responsibility to turn off the 80 column card at the end of a program. Do not quit the card with the cursor beyond the 39th column, as this can cause unpredictable results including program erasure. In case of accidently executing this command, pressing RETURN immediately will usually recover the cursor to the left margin. It is also necessary to turn the 80 column card off before sending output to printers, modems, etc.

VTAB no longer works when a window is set [by POKing 32,33 etc.]. The solution is to VTAB to the location -1, and then do a PRINT prior to PRINTing the actual data. This causes the firmware to recognise the new VTAB location.

These cautions are a small price to pay for the increased versatility and flexibility of the new Apple //e.

---

*Editor's Note: This material is intended to be used in conjunction with the original version of What's Where in the Apple which did not contain Apple IIe material.*

There is 1 page 0 location that was not formerly used which is now used.

\$1F (31) [YSAV1] \N1\ Temporary storage for the Y register

There are several locations in the text page that are storage for permanent data in these unused screen locations. Any routine which sets page 2 must restore page 1 so that these data may be accessed.

\$47B (1144) [TEMP1] \N1\ A temporary storage location  
 \$47B (1147) [OLDCH] \N1\ Old CH set for user  
 \$4FB (1275) [MODE] \N1\ Current operating mode according to the following bits:

Bit 0 Off Normal mode (Pascal)  
 Bit 0 On Transparent mode (Pascal)  
 Bit 0 Off Caller set interrupts (BASIC)  
 Bit 0 On Caller cleared interrupts (BASIC)  
 Bit 1 Off Pascal 1.1 F/W active  
 Bit 1 On Pascal 1.0 interface  
 Bit 2 Off Normal video (Pascal)  
 Bit 2 On Inverse video (Pascal)  
 Bit 3 Off GOTOXY not in progress  
 Bit 3 On GOTOXY in progress  
 Bit 4 Off Upper case restrict mode  
 Bit 4 On Literal upper/lower case mode  
 Bit 5 Off Current language is BASIC  
 Bit 5 On Current language is Pascal  
 Bit 6 Off BASIC PRINT  
 Bit 6 On BASIC INPUT  
 Bit 7 Off ESC-R inactive  
 Bit 7 On ESC-R active

\$57B (1403) [OURCH] \N1\ 80 column CH  
 \$5FB (1531) [OURCV] \N1\ Cursor vertical  
 \$67B (1659) [CHAR] \N1\ In/Out character  
 \$6FB (1787) [XCOORD] \N1\ X coordinate in GOTOXY routine  
 \$77B (1915) [OLDBASL] \N1\ Pascal saved BASL  
 \$7FB (2043) [OLDBASH] \N1\ Pascal saved BASH

\$C000-\$C01F (49152-49183) \N1\ Hardware locations/switches  
 \$C000 (49152) [CLR80COL] \N1\ Disable 80 column store  
 \$C001 (49153) [SET80COL] \N1\ Enable 80 column store  
 \$C002 (49154) [RDMAINRAM] \N1\ Read RAM on mainboard  
 \$C003 (49155) [RDCARDRAM] \N1\ Read RAM on card  
 \$C004 (49156) [WRMAINRAM] \N1\ Write RAM on mainboard  
 \$C005 (49157) [WRCARDRAM] \N1\ Write RAM on card  
 \$C007 (49159) [SETINTXCROM] \N1\ Set internal CX00 ROM  
 \$C008 (49160) [SETST0ZP] \N1\ Set standard zero page/stack  
 \$C009 (49161) [SETALT2ZP] \N1\ Set alternate zero page/stack  
 \$C00B (49163) [SETSL0TC3ROM] \N1\ Enable C300 slot ROM  
 \$C00C (49164) [CLR80VID] \N1\ Disable 80 column video  
 \$C00D (49165) [SET80VID] \N1\ Enable 80 column video  
 \$C00E (49166) [CLRALTCCHAR] \N1\ Normal lower case, flash upper case  
 \$C00F (49167) [SETALTCCHAR] \N1\ Normal/inverse lower case, no flash  
 \$C011 (49169) [RDLCBK2] \N1\ Reads language card bank 2  
 \$C012 (49170) [RDLCRAM] \N1\ Reads language card RAM enable  
 \$C013 (49171) [RDDRAMRD] \N1\ Reads RAMREAD state  
 \$C014 (49172) [RDDRAMWRT] \N1\ Reads BANKWRT state  
 \$C018 (49176) [RDBDCOL] \N1\ Reads SET80COL  
 \$C019 (49177) [RDVBLBAR] \N1\ Reads VBL signal  
 \$C01A (49178) [RDTEXT] \N1\ Reads Text mode  
 \$C01C (49180) [RDPAGE2] \N1\ Reads page 1/2 status  
 \$C01F (49183) [RD80VID] \N1\ Reads SET80VID

\$C100-\$CFFF (49408-53247) [CX00ROM] \N5\ A new set of subroutines to handle the 80 column card and auxiliary memory in slot 3. It is entered from the GOTOXC subroutine in the F800 ROM which sets interrupts, turns on the CX00 ROMs, and JMPs to C100. Function code is in Y reg. Note: "B." routines are the new way. "F." routines are the old way. Stack has status of bank and IRQ. Uses A,Y registers.  
 Function Codes:

0 CLR0OP  
 1 HOME  
 2 SCROLL  
 3 CLREOL  
 4 CLEOLZ  
 5 INIT & RESET  
 6 KEYIN  
 7 Fix ESCape Character  
 8 SETWMD

If there is a card in the slot then the new video routines are used, since the screen hole locations belong to the card. Otherwise the FB ROM routines are duplicated to avoid slot 3 interference with another type of interface. Entry point for all routines with code in Y. Check first for KEYIN Y=8 Check for ESCape-fix Y=7 Test for card. If present, use the new routines, if not, old routines

\$C100 (49408) [B.FUNC] \N5\  
 \$C107 (49415) [B.FUNCNK] \N5\  
 \$C10E (49422) [B.FUNCNE] \N5\

HEX LOCN (DEC LOCN) [NAME] USE-TYPEX - DESCRIPTION

\$C11F (49439) (B.OLDFUNC) \SEN Pushes \$C1 on stack, and low byte address of the function -1 by looking up in F.TABLE indexed by Y. Then does fake RTS to routine.  
 \$C129 (49449) (F.CLRLOEP) \SEN Monitor S/R to clear from the cursor to the end of page.  
 \$C143 (49475) (F.HOME) \SEN Clear scroll window to blanks. Set cursor to top left corner.  
 \$C140 (49485) (F.SCROLL) \SEN Monitor S/R to scroll up one line.  
 \$C170 (49533) (F.CLRLOEL) \SEN Monitor S/R to clear to end of line.  
 \$C18A (49546) (F.SETWND) \SEN Monitor S/R to set normal low-resolution graphics window, cursor bottom left.  
 \$C19C (49564) (F.CLEOLZ) \SEN Monitor S/R to clear entire line.  
 \$C1A1 (49569) (F.GORET) \NL Exit routine to F.RETURN  
 \$C1A4 (49572) (B.FUNCO) \SEN Entry point to new routines. Sets the IRQ mode and screen holes, Y req.  
 \$C1CD (49613) (B.SCROLL) \SEN Entry point for monitor routine to scroll up one line  
 \$C1CE (49619) (B.CLRLOEL) \SEN Entry point for monitor routine to clear to end of line  
 \$C1D9 (49625) (B.CLEOLZ) \SEN Entry point for monitor routine to clear entire line  
 \$C1E1 (49633) (B.CLRLOEP) \SEN Entry point for monitor routine to clear to end of page  
 \$C1E7 (49639) (B.SETWND) \SEN Entry point for monitor routine to set text window  
 \$C1EA (49642) (B.RESET) \SEN Entry point for monitor routine to reset entire system  
 \$C1ED (49645) (B.HOME) \SEN Monitor S/R to clear the text page and put cursor in upper left corner  
 \$C1FF (49683) (B.VECTOR) \SEN Monitor S/R to check on 80 col use and get current Cursor Horizontal position (CH) Save CH in screenhole  
 \$C20E (49678) (B.GETCH) \SEN  
 \$C211 (49681) (B.FUNC1) \SEN Pushes \$C1 on stack, and low byte address of the function -1 by looking up in B.TABLE indexed by Y. Then does fake RTS to routine.  
 \$C219 (49689) (B.SETWNDX) \SEN Monitor S/R to set normal text window 40/80 columns  
 \$C234 (49716) (B.RESETX) \SEN Monitor routine to reset system, checks for "Apple" keys for cold start, else does warm restart without diagnostics, blasts memory from BPXX down to stack, checks 80 col board to see if CX ROM needs resetting, and returns  
 \$C261 (49761) (DIAGS) \SEN Entry point for monitor S/R diagnostics  
 \$C26E (49774) (B.ESCFIX) \SEN Monitor S/R to map i,j,k,m and (-,\*,-) and V into I,J,K,M for cursor movement  
 Returns with old form of character in A.  
 \$C280 (49792) (ESCIN) \PA Table of arrow keys  
 \$C284 (49796) (ESCOUT) \PA "I,J,K,M" translations for arrows  
 \$C28A (49800) (B.KEYIN) \SEN Monitor routine to read a key with new additions to save CX bank status, check interrupt status, put new cursor ASC"\$FF" on screen, JSR to KEYDLY (old RKEYE), restore the original screen character, put the new character in A reg., clear the keyboard strobe and return to caller.  
 \$C2C6 (49862) (KEYDLY) \SEN Monitor routine to get a key from KBD, also checking interrupts, and still incrementing RNOL and RBNH, the random locations  
 \$C2EB (49899) (F.RETURN) \SEN Monitor routine to exit from CX ROM routines either leaving I/O disabled or enabling it if it was on entry  
 \$C300 (49920) (BASICINT) \SEN Sets INIT Flag (V) and branches to BASIC I/O entry point  
 \$C307 (49927) (BASICOUT) \SEN Clears INIT Flag (V) and branches to BASIC I/O entry point  
 \$C308 (49931) (PASFP7) \PB Pascal 3.1 firmware protocol table  
 \$C311 (49937) (I28KJMP) \PB Jump table for I28K support routines  
 \$C317 (49943) (BASICINT) \SEN BASIC I/O entry point, saves CHAR, A, Y, X, and P, pulls P from stack, checks IRQ status, and sets appropriately.  
 \$C336 (49974) (BASICINT2) \SEN Turns off any slots using C8 area, sets CBSLOT to \$C3, checks INIT flag, and jumps to warm or cold BASIC in C8 ROM  
 \$C34B-\$C362 (49995-50018) (PJUMPS) Pascal jump table  
 \$C34B (49995) (JPINIT) \SEN Pascal INIT  
 \$C351 (50011) (JPREAD) \SEN Pascal READ  
 \$C357 (50017) (JPWRITE) \SEN Pascal WRITE  
 \$C350 (50013) (JPSTAT) \SEN Pascal STATUS  
 \$C363 (50019) (MOVE) \SEN Monitor S/R to move memory across memory banks. Call with A1 = Source start, A2 = Source end, A4 = Destination start. Carry set for Main to Card, Carry clear for Card to Main.  
 \$C3B0 (50096) (XFER) \SEN Transfer program control from main board to card or vice versa. \$3ED-\$3EE is address to be executed upon transfer, carry set means transfer to card, carry clear means transfer to main board, V flag clear means use standard zero page/stack, V flag set means use alternate zero page/stack.  
 Also uses \$3ED-\$3EE in destination bank.  
 Enter via JMP not JSR.  
 Setup IRQ C800 protocol. Stores \$C3 in CBSLOT.  
 \$C3EB (50155) (SETC8) \SEN  
 \$C800 (51200) (PINIT1) \SEN Pascal 1.0 init  
 \$C803 (51203) (BASICINIT) \SEN Checks the F8 ROM version, if not //w, copies ROM to RAM Card, and checks again, if still not good, hangs the system.  
 Set up BASIC I/O in CSW and KSW to point to BASICINT in the C3 ROM and set text or graphics windows  
 \$C816 (51222) (BINIT1) \SEN Pascal 1.0 input hook  
 \$C848 (51272) (PREAD1,0) Check for 80 column mode and enable, if so  
 \$C850 (51280) (BINIT2) \NL Monitor routine to set lower case mode, clear screen and clears carry  
 \$C85D (51293) (CLEARIT) \NL Monitor routine to check mode and set 80 column store in case Integer BASIC cleared also rounds WINDOW to next lower even, if odd in 80 column mode.  
 \$C866 (51302) (CBBASIC) \NL Monitor routine to check current CH and store it if different from OLOCH  
 \$C874 (51316) (CBB2) \NL Monitor routine to check RAM card for correct version and, if not, recopy the FBROM to RAM card, check again and hang if not correct.  
 \$C87E (51326) (CBB3) \NL Monitor routine to check carry, on clear-print a character, set-input a character  
 \$C890 (51344) (CBB4) \NL Monitor S/R to set MODE to BASIC input, falls through to BPRINT  
 \$C896 (51350) (BOUT) \NL Monitor S/R to check character in CHAR, checks for CTRL-S, clears high bit, checks for CTRL chars, if it is, process and return, if not, fall through to BPNCTL  
 \$C8A1 (51361) (BPRINT) \NL Monitor S/R to reload CHAR (to get 8th bit), and print the char on the screen, increments cursor horizontal and scrolls, if necessary  
 \$C8CC (51404) (BPNCTL) \SEN Monitor routine to store cursor position, restore X, Y, and A and return to BASIC  
 \$C8E2 (51426) (BIORET) \NL Monitor routine to set MODE to BASIC input, get the cursor position, and CHAR  
 \$C8F6 (51446) (BINPUT) \NL Monitor routine to get character at current position, get a char from the keyboard, remove cursor, and process char, including ESCAPES, if not ESC then JMP to NOESC.  
 \$C905 (51461) (B.INPUT) \SEN

\$C918 (51480) [ESCAPING] %SEN	Monitor routine to process ESCape command sequences. The commands are: # - Home and Clear screen E - Clear to end of line F - Clear to end of page A,K,→ - Cursor right B,J,← - Cursor left C,M,V - Cursor down D,I,^ - Cursor up R - Restrict to uppercase T - Turn off Esc-R 4 - Go to 40 column mode 8 - Go to 80 column mode CTRL-Q - Quit new routines. (PR#0/IN#0)
\$C972 (51570) [ESCTAB] %P17%	Places ESCape cursor on screen, GETs a command key, puts lower case into upper, checks the ESCTAB for a valid character. If the char is there, load A with the Y index into ESCCHAR, and "print" the control character, if its not, check for "M", "R" and "CTRL-Q" special functions and process, if its not, return to caller. If the ESCCHAR entry has the high bit set, return to ESCAPING, otherwise return to caller.
\$C983 (51587) [ESCHAR] %P17%	Table of corresponding control codes-high bit set for "remain in ESCape mode"
\$C994 (51604) [PSTAT] %SEN	pascal check if ready for input or output, return 3 in X if not ready (ILLEGAL OPERATION) Pascal 1.0 output hook
\$C9A6 (51622) [PHOOK] %SEN	Monitor routine to process normal characters. Checks for copy char (right arrow). literal input, double quotes to turn literal input off/on, and restricted case input before storing in CHAR and returning to caller
\$C9B7 (51639) [INDESC] %SEN	Monitor routine to check for cancelling literal mode
\$C9DF (51679) [B_CHKCAN] %LN	Monitor routine to switch the literal mode
\$C9F7 (51703) [B_FLIP] %LN	Monitor routine to cancel literal mode
\$CA02 (51714) [B_CANLIT] %LN	Monitor routine to up/shift the character in non-literal or restrict mode
\$CA0A (51722) [B_FIXCHAR] %LN	Monitor routine to return to caller from input
\$CA24 (51748) [B_INRET] %LN	Monitor S/R to check the character before the cursor. Uses OURCH, OURCV: destroys A, TEMP1; outputs BEG if character is double quote, BNE if not. Used for changing literal mode if backspacing over a double quote.
\$CA27 (51751) [GETPRIO] %SEN	Pascal initializing 1.0 Pascal initialization 1.1
\$CA4A (51786) [PINIT1.0] %SEN	Set up for running Pascal, set mode, set window, zero page, check for card, return X=9 if DEVICE; if missing, turn on character, set normal lower case mode, home and clear screen, put cursor on screen and return.
\$CA4F (51791) [PINIT] %SEN	Pascal input-Get a character, remove high bit, store in CHAR, if 1.1 return "C3" in X, 1.0 return CHAR in A
\$CA51 (51793) [PINIT2] %LN	Pascal output-Set zero page, turn cursor off, check GOTOXY Mode and process if necessary, check if GOTOXY and start if true, else store it on screen, increment cursor horizontal, check if transparent mode and do carriage return/line feed if necessary, replace the cursor and return.
\$CA74 (51828) [PREAD] %SEN	Monitor S/R to read the keyboard, incrementing the random locations while waiting, load the char into A, clear the keyboard strobe and return
\$CABE (51854) [PWRITE] %SEN	Monitor S/R to test for presence of 80 column card, destroys A,Y; returns BEG if card is there, BNE if not.
\$CB15 (51989) [GETKEY] %SEN	Monitor S/R to calculate base address for screen line using OURCV. Stores result in BASL/BASH.
\$CB24 (52004) [TESTCARD] %SEN	Monitor S/R to calculate base address for screen line using CV. Checks for 40/80 column mode and if IRQ is enabled and not in Pascal, uses SNIFFIRO to check for interrupts.
\$CB51 (52049) [BASCALC] %SEN	Monitor S/R to process command control characters. Char in A to process, returns BCC if executed, BCS if not control command.
\$CB54 (52052) [BASCALC2] %SEN	Monitor routine to push CTLADH and CTLADL onto stack for control routine address and execute a fake RTS.
\$CB99 (52121) [CTLCHAR] %SEN	Monitor S/R to beep speaker, same as FB: BELL1
\$CB86 (52150) [CTLXFER] %LN	Monitor S/R to wait depending on A. Same as FB: WAIT
\$CBC8 (52156) [X_BELL] %SEN	Monitor S/R to execute a backspace
\$CBCF (52175) [WAIT] %SEN	Monitor S/R to execute a carriage return
\$CBDB (52187) [X_B5] %SEN	Monitor S/R to execute HOME
\$CBEC (52204) [X_CR] %SEN	Monitor S/R to execute clear line
\$CC00 (52237) [X_EM] %SEN	Monitor S/R to execute forward space
\$CC1A (52250) [X_SUB] %SEN	Monitor S/R to execute a reverse linefeed
\$CC26 (52262) [X_FS] %SEN	Monitor S/R to execute "normal video"
\$CC34 (52276) [X_US] %SEN	Monitor S/R to execute "inverse video"
\$CC49 (52287) [X_S0] %SEN	Table of low byte addresses for control characters subroutines: 0 = invalid
\$CC52 (52368) [X_S1] %SEN	Table of high byte addresses for control character subroutines: 0 = invalid
\$CC5F (52319) [CTLADL] %P24%	Monitor S/R to execute linefeed
\$CC78 (52344) [CTLADH] %P24%	Monitor S/R to scroll the screen up one line
\$CC91 (52369) [X_LF] %SEN	Monitor S/R to scroll the screen down one line
\$CCA4 (52388) [SCROLLUP] %SEN	Monitor routine to check for 40/80 columns
\$CCAA (52394) [SCROLLDN] %SEN	Monitor routine to scroll 40 columns
\$CCAE (52398) [SCROLL1] %LN	Monitor routine to scroll the other 40 columns
\$CCBC (52408) [SCROLL2] %LN	Monitor S/R to scroll only 40 column active window
\$CCCB (52416) [SCROLL80] %LN	Monitor routine to clear top or bottom line (depending on scroll up or down)
\$CCD1 (52433) [SCRLSUB] %SEN	Return to user via BASCALC.
\$CCD5 (52437) [X_SRLRET] %LN	Monitor S/R to clear to end of page
\$CD03 (52515) [X_VT] %SEN	Monitor S/R to home the cursor. Returns via X.VT to clear screen.
\$CD02 (52546) [X_FF] %SEN	Monitor S/R to clear to end of line
\$CD48 (52552) [X_G5] %SEN	Monitor S/R to clear entire line
\$CD4E (52558) [X_G5EOL2] %SEN	Monitor S/R to set 40 column mode
\$CD59 (52569) [X_DC1] %SEN	Monitor S/R to set 80 column mode
\$CD77 (52599) [X_DC2] %SEN	Monitor S/R to quit 80 column card
\$CD90 (52624) [X_NAK] %SEN	Monitor S/R to set full 80 column window parameters
\$CD9B (52635) [FULL80] %SEN	Monitor S/R to restore 40 column window, convert 80 to 40 if needed, set cursor at bottom left corner, reset video and keyboard to old mode
\$CDA4 (52650) [QUIT] %SEN	

\$CDBB (52699) [SCRNB4] \N\$N Monitor S/R to convert 80 column screen to 40 column screen. Moves leftmost 40 characters to TXTPAGE1  
 \$CEBA (52746) [ATEFOR] \N\$N Monitor S/R to convert one line from 80 to 40 columns  
 \$CE22 (52776) [GETB4] \N\$N Monitor S/R to move one character from 80 window to 40 window  
 \$CE32 (52786) [SCRNB4] \N\$N Monitor S/R to convert 40 column screen to 80 column screen. Moves whole 40 character screen to left most 40 positions on 80 column screen  
 \$CEB3 (52835) [FORATE] \N\$N Monitor S/R to convert one line from 80 to 40 columns  
 \$CEB1 (52881) [CLRHALF] \N\$N Monitor S/R to clear right half of both screen pages  
 \$CEA3 (52899) [D048] \N\$N Monitor S/R to move one character from 80 to 40 columns  
 \$CEAF (52919) [SETCH] \N\$N Monitor S/R to set CURCH and CH. In 40 column mode sets to A value. In 80 column mode, sets to 0 unless less than 8 from end of line, in which case moves up near right  
 \$CEED (52957) [INVERT] \N\$N Monitor S/R to invert the character at the current screen location: CH,CV  
 \$CEF2 (52978) [STORCHAR] \N\$N Monitor S/R to store character in A at screen horizontal position Y  
 \$CF01 (52993) [PICK] \N\$N Monitor S/R to read the character at screen position Y = horizontal, returns with character  
 \$CF06 (52998) [SCREENIT] \N\$N Monitor S/R to either store character on screen or read character from screen. V clear for pick, V set for store, character in A for store, Y = CH position. Saves Y and checks for mode. 40 branches to SCREEN40, 80 falls through to SCREEN80  
 \$CF0E (53006) [SCREEN80] \N\$N Monitor routine to calculate which page, and if V set, branch to STOR80, otherwise read the character from the screen and return.  
 \$CF2A (53034) [STOR80] \N\$N Monitor routine to store the character on the screen.  
 \$CF37 (53047) [SCREEN40] \N\$N Monitor routine to get cursor position, and if V set, branch to STOR40, otherwise read the character from the screen and return.  
 \$CF4A (53066) [STOR40] \N\$N Monitor routine to store the character on the screen.  
 \$CF52 (53074) [ESCON] \N\$N Monitor S/R to save current character in CHAR and put inverse "\*" on screen. Returns via ESCRET.  
 \$CF65 (53093) [ESCOFF] \N\$N Monitor S/R to restore original character back on the screen that was saved in CHAR. Falls through to ESCRET.  
 \$CF8E (53102) [ESCRET] \N\$N Monitor routine to put character on screen and return.  
 \$CF78 (53112) [COPYROM] \N\$N Monitor S/R to copy the FB ROM to the language card. Destroys X and Y. Uses CWSL/CWSH (which it saves) as hook for transfer. Sets ROM/RAM banks for transfer, moves the bytes, and resets the language card to it's previous state before returning.  
 \$CF08 (53192) [PSETUP] \N\$N Monitor S/R to set up zero page for Pascal operation. Checks 40-80 columns, sets INWFLG, and updates BASL/BASH before returning.  
 \$CFEA (53226) [F.TABLE] \P\$N Table of addresses for ESCAPE functions in 40 column mode. Entries at \$CFF0-1 are used by SCROLL (Label = PLUSMINUS1).  
 \$CFF3 (53235) [B.TABLE] \P\$N Table of addresses for ESCAPE functions in 80 column mode. Entries at \$CFF9-A are used by SCROLL (Label = WNDTAB).

## Changes in the F800 ROM

\$7FF (63487) [?] was \$D7, is now \$78, appears to be unused  
 \$7A75-\$7A7A (64117-64122) [RESET] A change in the RESET code to allow for the presence of an 80 column card. Does a JSR to GOTOCX Y=5  
 \$FB0A-\$FB0D (64226-64269) [TITLE] APPLE II [-> Apple II  
 \$FB51-\$FB54 (64337-64340) [SETWND] A change in the SETWND code to allow for the presence of an 80 column card. Does a branch to GOTOCX Y=8  
 \$FB3A (64419) [ESCONW] A change in the ESCONW code to allow for i,j,k,m and arrow keys. Does JSR to RDESC which is the old KEYIN2  
 \$FB83 (64435) [VERSION] ID code for check on which kind of Apple it is //e=\$06 [f]=\$EA [g]=\$38  
 \$FB84-\$FB8D (64436-64448) [GOTOCX] Formerly NOPs, now code to save current ROM states, set interrupts, turn on CX00 ROMs and JMP to C100; new code for 80 cols. Requires function code to be in Y Reg.  
 \$FC42-\$FC45 (64578-64581) [CLREOP] Changed to branch to GOTOCX Y=0  
 \$FC46-\$FC57 (64582-64598) [COPYRT] Notice of copyright "(C) 1981-82, APPLE"  
 \$FC58-\$FC5B (64600-64603) [HOME] Changed to branch to GOTOCX Y=1  
 \$FC5C-\$FC61 (64604-64609) [AUTHOR1] "RICK A" for Rick Auricchio  
 \$FC70-\$FC71 (64624-64625) [SCROLL] Changed to jump to GOTOCX Y=2  
 \$FC72-\$FC74 (64626-64628) [XGOTOCX] A JMP to GOTOCX for long branching purposes  
 \$FC75-\$FC98 (64629-64667) [SNIFFIRQ] IRQ Sniffer for Video Code: A new routine to check the current video mode, CXROM usage, and check for interrupts  
 \$FC9C-\$FC9D (64668-64669) [CLREOL] Changed to branch to GOTOCX Y=3  
 \$FC9E-\$FCA7 (64670-64679) [CLREOLZ] Changed to branch to GOTOCX Y=4  
 \$FD10-\$FD20 (64878-64880) [KEYIN] Changed to jump to GOTOCX Y=6 KEYIN no longer falls through to KEYIN2.  
 \$FD21-\$FD28 (64881-64888) [RDESC] Formerly KEYIN2, changed to jump to GOTOCX Y=7  
 \$FD29-\$FD2D (64889-64813) [FUNCEXIT] Return from GOTOCX here: A new routine that restores the CXROM bank and the IRQ before an RTS to the calling routine.  
 \$FD30 (64816) [ESC] A change to JSR to RDESC instead of RKEYY  
 \$FD42-\$FD43 (64834-64835) [NOTCR] A change to NOPs of the cursor inverse mode. No longer needed now that the cursor is a standard character  
 \$FD83 (64899) [CAPST] \P\$N A change in the input AND mask that used to convert lower case input to upper case  
 \$FEAF (85199) [CKSUMFIX] \P\$N Correct CKSUM at create time.  
 \$FEC5-\$FEC9 (65221-65225) [AUTHOR2] "Bryan" for Bryan Stearns.

? (634B7) [\$F7FF] Monitor S/R to convert one line from 80 to 40 columns

ATEFOR (52746) [\$CE0A] N\$EN "RICK A" for Rick Auricchio

AUTHOR1 (64604-64609) [\$FCE5-\$FCE1] "Bryan" for Bryan Stearns.

AUTHOR2 (65221-65225) [\$FCE5-\$FCE9] Monitor routine to cancel literal mode

B.CANLIT (51714) [\$CA02] N\$L Monitor routine to check for cancelling literal mode

B.CHKCAN (51679) [\$C90F] N\$L Entry point for monitor routine to clear to end of line

B.CLREOL (49819) [\$C103] N\$EN Entry point for monitor routine to clear entire line

B.CLROLZ (49825) [\$C109] N\$EN Entry point for monitor routine to clear to end of page

B.CLREOP (49833) [\$C11E] N\$EN Entry point for monitor routine to clear to end of page

B.ESCFIX (49774) [\$C20E] N\$EN Monitor S/R to map i,j,k,m and (-, ", ", and V into I,J,K,M for cursor movement

B.INPUT (51461) [\$C905] N\$EN Monitor routine to inverse char at current position, get a char from the keyboard, remove cursor, and process char, including ESCapes. If not ESC then JMP to NOESC.

B.FIXCHAR (51722) [\$CA0A] N\$L Monitor routine to up/shift the character in non-literal or restrict mode

B.FLIP (51703) [\$C9F7] N\$L Monitor routine to switch the literal mode

B.FUNC (49408) [\$C100] N\$EN Entry point for all routines with code in Y. Check first for KEYIN Y=6

B.FUNC1 (49681) [\$C211] N\$EN Pushes \$C1 on stack, and low byte address of the function -1 by looking up in B.TABLE indexed by Y. Then does fake RTS to routine.

B.FUNCE (49422) [\$C10E] N\$EN Test for card, if present, use the new routines, if not, old routines

B.FUNCK (49415) [\$C107] N\$EN Check for Escape-fix Y=7

B.FUNCO (49572) [\$C1A1] N\$EN Monitor routine to new routines. Sets the IRQ mode and screen holes, Y reg.

B.GETCH (49678) [\$C20E] N\$EN Save CH in screenhole

B.INRET (51748) [\$CA24] N\$L Monitor routine to return to caller from input

B.KEYIN (49800) [\$C28B] N\$EN Monitor routine to read a key with new additions to save CX bank status, check interrupt status, put new cursor ASC"FFF" on screen, JSR to KEYLD (old RKEY)

B.OLDFUNC (49439) [\$C11F] N\$EN \$C1 on stack, and low byte address of the function -1 by looking up in F.TABLE indexed by Y. Then does fake RTS to routine.

B.RESEX (49716) [\$C234] N\$EN Monitor routine to reset system, checks for "Apple" keys for cold start, else does warm restart without diagnostics, blasts memory from BFFX down to stack, checks B0 col board to see if CX ROM needs resetting, and returns

B.SCROLL (49613) [\$C1CD] N\$EN Entry point for monitor routine to scroll up one line

B.SETND (49639) [\$C1E7] N\$EN Entry point for monitor routine to set text window

B.SETWDX (49689) [\$C219] N\$EN Monitor S/R to set normal text window 40/80 columns

B.TABLE (53255) [\$CFF3] N\$P\$N Table of addresses for ESCape functions in 80 column mode. Entries at \$CFF9-A are used by SCROLL (Label = WNDTAB).

B.VECTOR (49663) [\$C1FF] N\$EN Monitor S/R to check on 80 col use and get current Cursor Horizontal position (CH)

BASCALC (52049) [\$CB51] N\$EN Monitor S/R to calculate base address for screen line using OIRCV.

BASCALCZ (52052) [\$CB54] N\$EN Stores result in BASL/BASH.

Monitor S/R to calculate base address for screen line using CV. Checks for 40/80 column mode and if IRQ is enabled and not in Pascal, uses SNIFFIRQ to check for interrupts.

BASIC (49943) [\$C317] N\$EN BASIC I/O entry point, saves CHAR, A, Y, X, and P, pulls P from stack, checks IRQ status, and returns appropriately.

BASICENT2 (49974) [\$C336] N\$EN Turns off any slots using C8 area, sets C8SLOT to \$C3, checks INIT flag, and jumps to warm or cold BASIC in C8 ROM

DIAGS (49761) [\$C2B1] N\$EN Entry point for monitor S/R diagnostics

D048 (52899) [\$CEA3] N\$L Monitor routine to move one character from 80 to 40 columns

ESC (64816) [\$FD30] A change to JSR to RDESC instead of RKEY

ESCAPING (51480) [\$C918] N\$EN Monitor routine to process Escape command sequences. Places Escape cursor on screen, Gets a command key, puts lower case into upper, checks the ESCTAB for a valid character. If the char is there, load A with the Y index into ESCCHAR, and "print" the control character, if its not, check for "I", "R" and "CTRL-Q" special functions and process, if its not, return to caller.

ESCCHAR (51587) [\$C983] N\$P1\$N Table of corresponding control codes-high bit set for "remain in Escape mode"

ESCLN (49792) [\$C280] N\$P4\$N Table of arrow keys

ESCONV (64419) [\$FBA3] A change to ESCCHW code to allow for i,j,k,m and arrow keys. Does JSR to RDESC which is the old KEYIN2

ESCOFF (53093) [\$CF65] N\$EN Monitor S/R to replace original character back on the screen that was saved in CHAR. Falls through to ESCRET.

ESCON (53074) [\$CF52] N\$EN Monitor S/R to save current character in CHAR and put inverse "" on screen. Returns via ESCRET.

ESCOUT (49796) [\$C284] N\$P4\$N "J,K,M,I" translations for arrows

ESCRET (53102) [\$CF6E] N\$L Monitor routine to put character on screen and return.

ESCTAB (51570) [\$C972] N\$P1\$N Table of ESCape codes

F.CLREOL (49533) [\$C17D] N\$EN Monitor S/R to clear to end of line.

F.CLROLZ (49564) [\$C19C] N\$EN Monitor S/R to clear entire line.

F.CLREOP (49449) [\$C129] N\$EN Monitor S/R to clear from the cursor to the end of page.

F.GORET (49569) [\$C1A1] N\$L Exit routine to F.RETURN

F.HOME (49475) [\$C143] N\$EN Clear scroll window to blanks. Set cursor to top left corner.

F.RETURN (49899) [\$C2EB] N\$EN Monitor routine to exit from CX ROM routines either leaving I/O disabled or enabling it if it was on entry

F.SCROLL (49485) [\$C14D] N\$EN Monitor S/R to scroll up one line.

F.SETWND (49546) [\$C18A] N\$EN Monitor S/R to set normal low-resolution graphics window, cursor bottom left.

F.TABLE (53226) [\$CFEA] N\$P9\$N Table of addresses for ESCape functions in 40 column mode. Entries at \$CFF0-1 are used by SCROLL (Label = PLUSMINUS).

FORATE (52835) [\$CEB3] N\$EN Monitor S/R to convert one line from 80 to 40 columns

FULL80 (52835) [\$CEB8] N\$EN Monitor S/R to set full 80 column window parameters

FUNCXINT (64809-64813) [\$FD29-\$FD2D] Return from GOTOX here: A new routine that restores the CXROM bank and the IRQ before an RTS to the calling routine.

GET84 (52770) [\$CE22] N\$EN Monitor S/R to move one character from 80 window to 40 window

GETKEY (51989) [\$CB15] N\$EN Monitor S/R to read the keyboard, incrementing the random locations while waiting, load the char into A, clear the keyboard strobe and return

GETPRIOR (51751) [\$CA27] N\$EN Monitor S/R to read the keyboard, incrementing the random locations while waiting, load the char into A, clear the keyboard strobe and return

TEMP1; outputs BEG if character is double quote \$NE if not. Used for changing literal mode if backspacing over a double quote.

GOTOXC (64436-64448) [F8B4-F8BC0] Formerly NOPs, now code to save current ROM states, set interrupts, turn on CX00 ROMs and JMP to C100; new code for 80 cols. Requires function code to be in Y Reg. Changed to branch to GOTOXC Y=1

HOME (64600-64603) [F5C8-F5C5B] Monitor S/R to invert the character at the current screen location: CH,CV

INVERT (52957) [F5CEDD] VSE

JPINIT (49995) [F3C4B] VSE

JPREAD (50001) [F3C51] VSE

JPREAT (50013) [F3C5D] VSE

JPWRITE (50007) [F3C57] VSE

KEYDLV (49862) [F2C2C] VSE

KEYIN (64795-64800) [F4D1B-F4D20] Formerly NOPs, now code to save current ROM states, set interrupts, turn on CX00 ROMs and JMP to C100; new code for 80 cols. Requires function code to be in Y Reg. Changed to branch to GOTOXC Y=1

MODE (1275) [F4FB] VPI

MOVE (50019) [F3C63] VSE

NOESC (51639) [F3C87] VSE

NOCTR (64834-64835) [F4D42-F4D43] Changed to jump to GOTOXC Y=6 KEYIN no longer falls through to KEYIN2. Current operating mode according bits set.

OLDASH (2043) [F7FB] VPI

OLDRASL (1915) [F77B] VPI

OLDCH (1147) [F47B] VPI

128KJMP (49937) [F3C11] VPI

OURCH (1403) [F57B] VPI

OURCV (1531) [F5FB] VPI

PASPT (49931) [F3C0B] VPI

PHOOK (51622) [F3C96] VSE

PICK (52993) [F3C01] VSE

PINIT (51791) [F3CA4] VSE

PINIT1 (51200) [F3C80] VSE

PINIT1.0 (51786) [F3CA4] VSE

PINIT2 (51793) [F3CA5] VSE

PJUMPS (49995-50018) [F3C4B-F3C62] Monitor S/R to move memory across memory banks. Call with A1 = Source start, A2 = Source end, A4 = Destination start. Carry set for Main to Card, Carry clear for Card to Main.

PREAD (51828) [F3A74] VSE

PREAD1.0 (51272) [F3C8B] VSE

PSETUP (53192) [F3CFC] VSE

PSTATUS (51604) [F3C99A] VSE

BASIC/INIT (51203) [F3C803] VSE

BASIC/IN (49920) [F3C30] VSE

BASIC/OUT (49927) [F3C37] VSE

BINIT1 (51222) [F3C816] VSE

BINIT2 (51280) [F3C850] VSE

BINPUT (51446) [F3C8F6] VSE

BIPRET (51426) [F3C8E2] VSE

BOUT (51350) [F3C866] VSE

BPNCPL (51404) [F3C8CC] VSE

BPRINT (51361) [F3C8A1] VSE

CB2 (51316) [F3C874] VSE

CB3 (51326) [F3C87E] VSE

CB4 (51344) [F3C890] VSE

CBASIC (51302) [F3C886] VSE

CAPTST (64899) [F4D83] VPI

CHAR (1659) [F67B] VPI

CKSUMF (X (65199) [F5EAF] VPI

CLEARIT (51293) [F3C850] VSE

CLR80COL (49152) [F3C000] VSE

CLR80VID (49164) [F3C00C] VSE

CLRALTCHAR (49166) [F3C00E] VSE

CREOL (64668-64669) [F3C9C-F3C9D] VSE

CREOLZ (64670-64679) [F3C9E-F3CA7] VSE

CREOP (64578-64581) [F3C42-F3C45] VSE

CRHALF (52881) [F3C71] VSE

COPYROM (53112) [F3E9F] VSE

COPYRT (64582-64599) [F3C46-F3C57] Notice of copyright "(C) 1981-82, APPLE"

CTLADH (52344) [F3C78] VPI

CTLADL (52319) [F3C5F] VPI

CTLCHAR (51211) [F3C899] VSE

CTLXFER (52150) [F3C886] VSE

CX00ROM (49408-53247) [F3100-F3FFF] Monitor S/R to process command control characters. Char in A to process, returns BCC if executed, BCS if not control command.

PWRITE (51854) [SCABE] \SEN  
 Pascal output-Set zero page, turn cursor off, check GOTOXY Mode and process if necessary, check if GOTOXY and start if true, else store it on screen, increment cursor horizontal, check if transparent mode and do carriage return/line feed if necessary, replace the cursor and return.

QUIT (52650) [SCDAA] \SEN  
 Monitor 5/R to restore 40 column window, convert 80 to 40 if needed, set cursor at bottom left corner, reset video and keyboard to old mode

RD08COL (49176) [SC018] \H1N  
 Reads SET80COL  
 RD08VID (49183) [SC01F] \H1N  
 Reads SET80VID  
 RDCARDRAM (49155) [SC003] \H1N  
 Read RAM on card  
 RDESC (64801-64808) [F021-F028]  
 Reads language card bank 2  
 RDCBK8 (49169) [SC011] \H1N  
 Reads language card RAM enable  
 RDLCRAM (49170) [SC012] \H1N  
 Read RAM on mainboard  
 RDMAINRAM (49154) [SC002] \H1N  
 Reads page 1/2 status  
 RDAPAGE2 (49180) [SC01C] \H1N  
 Reads RAM/READ state  
 RDRAMRPT (49171) [SC013] \H1N  
 Reads RAM/READ state  
 RDRAMWRT (49172) [SC014] \H1N  
 Reads Text mode  
 RDTEXT (49178) [SC01A] \H1N  
 Reads VBL signal  
 RDVBLBAR (49177) [SC019] \H1N  
 A change in the RESET code to allow for the presence of an 80 column card. Does a JSR to GOTOXC Y=5

RESET (64117-64122) [FA75-FA7A]  
 Monitor routine to get cursor position, and if V set, branch to STOR40, otherwise read the character from the screen and return.  
 Monitor routine to calculate which page, and if V set, branch to STOR80, otherwise read the character from the screen and return.  
 Monitor 5/R to either store character on screen or read character from screen. V clear for pick, V set for store, character in A, 80 falls through to SCREEN80  
 Saves Y and checks for mode, 40 branches to SCREEN40, 80 falls through to SCREEN80  
 Monitor 5/R to scroll only 40 column active window  
 Monitor 5/R to convert 40 column screen to 80 column screen. Moves whole 40 character screen to left most 40 positions on 80 column screen  
 Monitor 5/R to convert 80 column screen to 40 column screen. Moves leftmost 40 characters to TXTPAGE1

SCREEN40 (53047) [FCF37] \LN  
 Changed to jump to GOTOXC Y=2  
 SCREEN80 (53006) [FCF0E] \LN  
 Monitor routine to scroll 40/80 columns  
 SCREENIT (52998) [FCF06] \SEN  
 Monitor routine to scroll 40 columns  
 Monitor routine to scroll 40 columns  
 Monitor routine to scroll the other 40 columns  
 Monitor 5/R to scroll the screen down one line  
 Monitor 5/R to scroll the screen up one line

SCRLSUB (52433) [SC001] \SEN  
 Enable 80 column store  
 SCRN48 (52786) [CE32] \SEN  
 Enable 80 column video  
 SCRN84 (52699) [SCD0B] \SEN  
 Normal/inverse lower case, no flash  
 Set alternate zero page/stack  
 Setup IRQ C800 protocol. Stores IC3 in CBSLOT.  
 Monitor 5/R to set OVRCH and CH. In 40 column mode sets to A value. In 80 column mode, sets to 0 unless less than 8 from end of line, in which case moves up near right  
 Set internal CX00 ROM  
 Enable C300 slot ROM  
 Set standard zero page/stack  
 A change in the SETWND code to allow for the presence of an 80 column card. Does a branch to GOTOXC Y=8  
 IRQ Sniffer for Video Code: A new routine to check the current video mode!  
 CXROM usage and interrupt status  
 Monitor routine to store the character on the screen.  
 Monitor routine to store the character on the screen.  
 Monitor 5/R to store character in A at screen horizontal position Y.  
 A temporary storage location  
 Monitor 5/R to test for presence of 80 column card, destroys A,Y; returns BEQ if card is there, BNE if not.  
 APPLE -> Apple  
 ID code for check on which kind of Apple it is //e-\$08 +\$EA =38  
 Monitor 5/R to wait depending on A. Same as FB: WAIT  
 Write RAM on card  
 Write RAM on mainboard  
 Monitor 5/R to beep speaker, same as FB: BELL1  
 Monitor 5/R to execute a backspace  
 Monitor 5/R to execute a carriage return  
 Monitor 5/R to set 40 column mode  
 Monitor 5/R to set 80 column mode  
 Monitor 5/R to execute HOME  
 Monitor 5/R to home the cursor. Returns via X.VT to clear screen.  
 Monitor 5/R to execute a forward space  
 Monitor 5/R to clear to end of line  
 Monitor 5/R to clear entire line  
 Monitor 5/R to execute linefeed  
 Monitor 5/R to quit 80 column card  
 Monitor routine to clear top or bottom line (depending on scroll up or down)  
 Return to user via BASCALC.  
 Monitor 5/R to execute "inverse video"  
 Monitor 5/R to execute "normal video"  
 Monitor 5/R to execute clear line  
 Monitor 5/R to execute a reverse linefeed  
 Monitor 5/R to clear to end of page  
 X coordinate in GOTOXY routine  
 Transfer program control from main board to card or vice versa. \$3ED-\$3EE is address to be executed upon transfer, carry set means transfer to card, carry clear means transfer to main board, V flag clear means use standard zero page/stack, V flag set means use alternate zero page/stack. Also uses \$3ED-\$3EE in destination bank. Enter via JMP not JSR.  
 A JMP to GOTOXC for long branching purposes  
 Temporary storage for the Y register

SCROLL (64624-64625) [FC70-FC71]  
 SCROLL1 (52399) [SCA1] \LN  
 Monitor routine to scroll 40/80 columns  
 SCROLL2 (52408) [SCCB8] \LN  
 Monitor routine to scroll 40 columns  
 SCROLL80 (52416) [SCCC0] \LN  
 Monitor routine to scroll the other 40 columns  
 SCROLLDN (52394) [SCCAA] \SEN  
 Monitor 5/R to scroll the screen down one line  
 SCROLLUP (52388) [SCCA4] \SEN  
 Monitor 5/R to scroll the screen up one line

SET80COL (49153) [SC001] \H1N  
 Enable 80 column store  
 SET80VID (49165) [SC004] \H1N  
 Enable 80 column video  
 SETALTCHAR (49187) [SC00F] \H1N  
 Normal/inverse lower case, no flash  
 SETALTZP (49161) [SC009] \H1N  
 Set alternate zero page/stack  
 SETCB (50155) [SC3EB] \SEN  
 Setup IRQ C800 protocol. Stores IC3 in CBSLOT.  
 SETCH (52911) [CEAF1] \SEN  
 Monitor 5/R to set OVRCH and CH. In 40 column mode sets to A value. In 80 column mode, sets to 0 unless less than 8 from end of line, in which case moves up near right  
 Set internal CX00 ROM  
 Enable C300 slot ROM  
 Set standard zero page/stack  
 A change in the SETWND code to allow for the presence of an 80 column card. Does a branch to GOTOXC Y=8  
 IRQ Sniffer for Video Code: A new routine to check the current video mode!  
 CXROM usage and interrupt status  
 Monitor routine to store the character on the screen.  
 Monitor routine to store the character on the screen.  
 Monitor 5/R to store character in A at screen horizontal position Y.  
 A temporary storage location  
 Monitor 5/R to test for presence of 80 column card, destroys A,Y; returns BEQ if card is there, BNE if not.  
 APPLE -> Apple  
 ID code for check on which kind of Apple it is //e-\$08 +\$EA =38  
 Monitor 5/R to wait depending on A. Same as FB: WAIT  
 Write RAM on card  
 Write RAM on mainboard  
 Monitor 5/R to beep speaker, same as FB: BELL1  
 Monitor 5/R to execute a backspace  
 Monitor 5/R to execute a carriage return  
 Monitor 5/R to set 40 column mode  
 Monitor 5/R to set 80 column mode  
 Monitor 5/R to execute HOME  
 Monitor 5/R to home the cursor. Returns via X.VT to clear screen.  
 Monitor 5/R to execute a forward space  
 Monitor 5/R to clear to end of line  
 Monitor 5/R to clear entire line  
 Monitor 5/R to execute linefeed  
 Monitor 5/R to quit 80 column card  
 Monitor routine to clear top or bottom line (depending on scroll up or down)  
 Return to user via BASCALC.  
 Monitor 5/R to execute "inverse video"  
 Monitor 5/R to execute "normal video"  
 Monitor 5/R to execute clear line  
 Monitor 5/R to execute a reverse linefeed  
 Monitor 5/R to clear to end of page  
 X coordinate in GOTOXY routine  
 Transfer program control from main board to card or vice versa. \$3ED-\$3EE is address to be executed upon transfer, carry set means transfer to card, carry clear means transfer to main board, V flag clear means use standard zero page/stack, V flag set means use alternate zero page/stack. Also uses \$3ED-\$3EE in destination bank. Enter via JMP not JSR.  
 A JMP to GOTOXC for long branching purposes  
 Temporary storage for the Y register

SETINTCXROM (49159) [SC007] \H1N  
 Set internal CX00 ROM  
 SETSLOT3ROM (49163) [SC00B] \H1N  
 Enable C300 slot ROM  
 SETSTDZP (49160) [SC008] \H1N  
 Set standard zero page/stack  
 SETWND (64337-64340) [FBF51-FBF54]  
 A change in the SETWND code to allow for the presence of an 80 column card. Does a branch to GOTOXC Y=8  
 IRQ Sniffer for Video Code: A new routine to check the current video mode!  
 CXROM usage and interrupt status  
 Monitor routine to store the character on the screen.  
 Monitor routine to store the character on the screen.  
 Monitor 5/R to store character in A at screen horizontal position Y.  
 A temporary storage location  
 Monitor 5/R to test for presence of 80 column card, destroys A,Y; returns BEQ if card is there, BNE if not.  
 APPLE -> Apple  
 ID code for check on which kind of Apple it is //e-\$08 +\$EA =38  
 Monitor 5/R to wait depending on A. Same as FB: WAIT  
 Write RAM on card  
 Write RAM on mainboard  
 Monitor 5/R to beep speaker, same as FB: BELL1  
 Monitor 5/R to execute a backspace  
 Monitor 5/R to execute a carriage return  
 Monitor 5/R to set 40 column mode  
 Monitor 5/R to set 80 column mode  
 Monitor 5/R to execute HOME  
 Monitor 5/R to home the cursor. Returns via X.VT to clear screen.  
 Monitor 5/R to execute a forward space  
 Monitor 5/R to clear to end of line  
 Monitor 5/R to clear entire line  
 Monitor 5/R to execute linefeed  
 Monitor 5/R to quit 80 column card  
 Monitor routine to clear top or bottom line (depending on scroll up or down)  
 Return to user via BASCALC.  
 Monitor 5/R to execute "inverse video"  
 Monitor 5/R to execute "normal video"  
 Monitor 5/R to execute clear line  
 Monitor 5/R to execute a reverse linefeed  
 Monitor 5/R to clear to end of page  
 X coordinate in GOTOXY routine  
 Transfer program control from main board to card or vice versa. \$3ED-\$3EE is address to be executed upon transfer, carry set means transfer to card, carry clear means transfer to main board, V flag clear means use standard zero page/stack, V flag set means use alternate zero page/stack. Also uses \$3ED-\$3EE in destination bank. Enter via JMP not JSR.  
 A JMP to GOTOXC for long branching purposes  
 Temporary storage for the Y register

STOR40 (53066) [FCFAA] \LN  
 Monitor routine to store the character on the screen.  
 STOR80 (53034) [FCF2A] \LN  
 Monitor routine to store the character on the screen.  
 STORCHAR (52978) [CEFE2] \SEN  
 Monitor 5/R to store character in A at screen horizontal position Y.  
 TEMP1 (1144) [5478] \P1N  
 A temporary storage location  
 TESTCARD (52004) [CB24] \SEN  
 Monitor 5/R to test for presence of 80 column card, destroys A,Y; returns BEQ if card is there, BNE if not.  
 APPLE -> Apple  
 ID code for check on which kind of Apple it is //e-\$08 +\$EA =38  
 Monitor 5/R to wait depending on A. Same as FB: WAIT  
 Write RAM on card  
 Write RAM on mainboard  
 Monitor 5/R to beep speaker, same as FB: BELL1  
 Monitor 5/R to execute a backspace  
 Monitor 5/R to execute a carriage return  
 Monitor 5/R to set 40 column mode  
 Monitor 5/R to set 80 column mode  
 Monitor 5/R to execute HOME  
 Monitor 5/R to home the cursor. Returns via X.VT to clear screen.  
 Monitor 5/R to execute a forward space  
 Monitor 5/R to clear to end of line  
 Monitor 5/R to clear entire line  
 Monitor 5/R to execute linefeed  
 Monitor 5/R to quit 80 column card  
 Monitor routine to clear top or bottom line (depending on scroll up or down)  
 Return to user via BASCALC.  
 Monitor 5/R to execute "inverse video"  
 Monitor 5/R to execute "normal video"  
 Monitor 5/R to execute clear line  
 Monitor 5/R to execute a reverse linefeed  
 Monitor 5/R to clear to end of page  
 X coordinate in GOTOXY routine  
 Transfer program control from main board to card or vice versa. \$3ED-\$3EE is address to be executed upon transfer, carry set means transfer to card, carry clear means transfer to main board, V flag clear means use standard zero page/stack, V flag set means use alternate zero page/stack. Also uses \$3ED-\$3EE in destination bank. Enter via JMP not JSR.  
 A JMP to GOTOXC for long branching purposes  
 Temporary storage for the Y register

TITLE (64226-64269) [FB0A-FB0D]  
 VERSION (64435) [FBF83]  
 WAIT (52175) [CBCF1] \SEN  
 Monitor 5/R to wait depending on A. Same as FB: WAIT  
 Write RAM on card  
 Write RAM on mainboard  
 Monitor 5/R to beep speaker, same as FB: BELL1  
 Monitor 5/R to execute a backspace  
 Monitor 5/R to execute a carriage return  
 Monitor 5/R to set 40 column mode  
 Monitor 5/R to set 80 column mode  
 Monitor 5/R to execute HOME  
 Monitor 5/R to home the cursor. Returns via X.VT to clear screen.  
 Monitor 5/R to execute a forward space  
 Monitor 5/R to clear to end of line  
 Monitor 5/R to clear entire line  
 Monitor 5/R to execute linefeed  
 Monitor 5/R to quit 80 column card  
 Monitor routine to clear top or bottom line (depending on scroll up or down)  
 Return to user via BASCALC.  
 Monitor 5/R to execute "inverse video"  
 Monitor 5/R to execute "normal video"  
 Monitor 5/R to execute clear line  
 Monitor 5/R to execute a reverse linefeed  
 Monitor 5/R to clear to end of page  
 X coordinate in GOTOXY routine  
 Transfer program control from main board to card or vice versa. \$3ED-\$3EE is address to be executed upon transfer, carry set means transfer to card, carry clear means transfer to main board, V flag clear means use standard zero page/stack, V flag set means use alternate zero page/stack. Also uses \$3ED-\$3EE in destination bank. Enter via JMP not JSR.  
 A JMP to GOTOXC for long branching purposes  
 Temporary storage for the Y register

XGOTOXC (64626-64628) [FC72-FC74]  
 YSAV1 (31) [51F] \P1N