Slot Allocation:

Slot	Function	Slot	Function
0	Language card	1	Printer serial port
2	Modem serial port, 3rd and 4th hard drives	3	80 column card
4	Mouse (//c)	5	"Easter Egg" in early ROMs, external floppy/HDD later ROMs
6	Internal and external floppy drives	7	Mouse (//c Plus)

Memory Map:

$0000 - $BFFF	RAM
$0000 - $00FF	Zero Page
$0100 - $01FF	Stack
$0400 - $07FF	Text/Low Resolution Graphics Video Page 1
$0800 - $0BFF	Text/Low Resolution Graphics Video Page 2
$2000 - $3FFF	High Resolution Graphics Video Page 1
$4000 - $5FFF	High Resolution Graphics Video Page 2
$C000 - $CFFF	I/O
$C000 - $C0FF	Soft Switches and Status Locations
$C100 - $C7FF	Peripheral Card Memory
$C800 - $CFFF	Extended Memory for Peripheral Card in Use
$D000 - $FFFF	ROM/Bank-Switched RAM
$D000 - $DFFF	Bank-Switched RAM (2 Banks RAM, 1 Bank ROM)
$E000 - $FFFF	Bank-Switched RAM (1 Bank RAM, 1 Bank ROM)

Common Addresses:

Hex	Decimal	Function
$0020	32	Left margin of text window
$0021	33	Text window width
$0022	34	Top margin of text window
$0023	35	Bottom margin of text window
$0024	36	Cursor column
$0025	37	Cursor row
$004C, 4D	76,77	Integer Basic HIMEM
$0069, 6A	105, 106	Applesoft and Integer Basic LOMEM
$0073,74	115,116	Applesoft HIMEM
$C000	-16384	Read to get keyboard character (>127 means key pressed)
$C00E	-16370	Read to set primary character set
$C00F	-16369	Read to set secondary character set

Hex	Decimal	Function
$C010	-16368	Write to clear keyboard strobe
$C01A	-16358	Read for mode (<128 graphics, >127 text)
$C01B	-16357	Read for text window (<128 absent, >127 present)
$C01C	-16356	Read for screen page (<128 page 1, >127 page 2)
$C01D	-16355	Read for graphics mode (<128 low-res, >127 hi-res)
$C01E	-16354	Read for character set status (<128 primary, >127 alternate)
$C020	-16352	Read to toggle cassette output (not on Apple //c)
$C030	-16336	Read to toggle speaker
$C050	-16304	Read to set graphics mode
$C051	-16303	Read to set text mode
$C052	-16302	Read to set full screen graphics
$C053	-16301	Read to set graphics plus text
$C054	-16300	Read to set graphics page 1
$C055	-16299	Read to set graphics page 2
$C056	-16298	Read to set low resolution graphics
$C057	-16297	Read to set high resolution graphics
$C058	-16296	Read to set annunciator 0 off
$C059	-16295	Read to set annunciator 0 on
$C05A	-16294	Read to set annunciator 1 off
$C05B	-16293	Read to set annunciator 1 on
$C05C	-16292	Read to set annunciator 2 off
$C05D	-16291	Read to set annunciator 2 on
$C05E	-16290	Read to set annunciator 3 off
$C05F	-16289	Read to set annunciator 3 on
$C060	-16288	Read to get cassette tape input level (Not on Apple //c)
$C061	-16287	Read for pushbutton 0 status (<127 not pressed, >127 pressed)
$C062	-16286	Read for pushbutton 1 status (<127 not pressed, >127 pressed)
$C063	-16285	Read for pushbutton 2 status (<127 not pressed, >127 pressed)
$C064	-16284	Read for pushbutton 3 status (<127 not pressed, >127 pressed)
$C070	-16272	Read/write to trigger game control connector strobe output once/twice
$F666	-2458	Enter Mini-Assembler (ROM version 0 and later)
$FF69	-151	Enter monitor

Common Keyboard Commands:

Key	Function
<Control>-<Reset>	Halt current program
<Control>-<Open Apple>-<Reset>	Halt and reboot
<Esc><Control>-Q	Leave 80-column mode
<Control>-X	Cancel line
<Esc>E	Delete to end of line
<Esc>F	Delete to end of screen
<Esc>@	Clear screen

BASIC Keywords: (I=Integer BASIC only, A=Applesoft only, D=DOS, P=ProDOS)

_P	?A	APPENDD,P	AUTOI	BLOADD,P	BRUND,P	BSAVED,P
CALL	CATP	CATALOGD,P	CHAIND,P	CLEARA	CLOSED,P	CLRI
COLOR	CONI	CONTA	CREATEP	DATA	DEFFNA	DEL
DELETED,P	DIM	DRAWA	DSPI	END	EXECD,P	FLASHA
FLUSHP	FOR	FPI,D	FREP	GETA	GOSUB	GOTO
GR	HCOLORA	HGRA	HGR2A	HLIN	HMEM:	HOMEA
HPLOTA	HTABA	IF-THEN	IN#	INITD	INPUT	INVERSEA
LET	LIST	LOADD,P	LOCKD,P	LOMEM:	MANI	MAXFILESD
MOND	NEW	NEXT	NODSPI	NOTRACE	NOMOND	NORMALA
ONA	ONERRA	OPEND,P	PDL	PEEK	PLOT	POKE
POPA	POSITIOND,P	PR#	PREFIXP	PRINT	READD,P	RECALLA
REM	RENAMED,P	RESTOREP	RESUMEA	RETURN	RTA	RUND,P
SAVED,P	SCALEA	SHLOADA	SPEEDA	STOPA	STOREP	TABI
TEXT	TRACE	UNLOCKD,P	USR	VERIFYD	VLIN	VTAB
WAITA	WRITED,P	XDRAWA				

BASIC Functions:

ABS	ASC	ATNA	CHR$A	COSA	EXPA	FNA
FREA	INTA	LEFT$A	LEN	LOGA	MID$A	PDL
PEEK	POSA	RIGHT$A	RND	SCRN	SGN	SINA
SPCA	SQRA	STR$A	TABA	USRA	VALA	

Monitor Commands:

<Control>-C	Return (e.g. to BASIC)
<*slot*><Control>-P	Send output to/activate device in slot <*slot*>
<slot><Control>-K	Accept input from device in slot <slot>

<*a*>	Display contents of address <*a*>
<Return>	Display contents of next 8 bytes of memory
<*a*>.<*b*>	Display range of memory from <*a*> to <*b*>
<Control>-E	Examine registers. Can then change values using : *a x y p s*
<*a*>:<*b*>...	Modify memory locations starting at address <*a*>. Omit <*a*> to use next address. Prefix data with ' and follow by space to enter a character.
<*start*>.<*end*>W	Write memory to tape (not on Apple //c)
<*start*>.<*end*>R	Read memory from tape (not on Apple //c)
<*dest*><<*start*>.<*end*>M	Move (copy) memory
<a>S	Step one instruction from address <a> and display registers. Omit <a> to step next instruction. (later ROMs only)
<a>T	Trace instructions from address <a> until BRK instruction or Closed-Apple key pressed. Press Open-Apple key to slow down speed of trace. (later ROMs only)
<*dest*><<*start*>.<*end*>V	Verify memory
<*a*>G	Go to address <*a*>
<*a*>L	Disassemble code from address <*a*>
I	Set inverse video
N	Set normal video
<*n1*>+<*n2*>	Add hex numbers
<*n1*>-<*n2*>	Subtract hex numbers
<Control>-Y	Jump to user-defined routine at $03F8
!	Enter mini-assembler (later ROMs only)

Mini-Assembler Commands:

$<*monitor command*>	Run monitor command (early ROM only)
<*mnem*> <*operands...*>	Assemble instruction at current address
<*addr*>:<*mnem*> <*operands...*>	Assemble instruction at specific address
<Return>	Empty line returns to monitor

Common 80 Column Card Control Characters:

7	Beep speaker	8	Cursor left	10	Cursor down
11	Clear screen from cursor	12	Clear screen	14	Set normal video
15	Set inverse video	17	Set active-40 mode	18	Set active-80 mode
21	Leave 80 column mode	22	Scroll down	23	Scroll up
24	Deactivate Mousetext	25	Cursor home	26	Clear line
27	Activate Mousetext	28	Cursor right	29	Clear to end of line

Prompts:

>	Integer BASIC	]	Applesoft	*	Monitor	!	Mini-Assembler

Lo-Res Graphics Colors:

0	Black	1	Magenta	2	Dark Blue	3	Purple
4	Dark Green	5	Grey 1	6	Medium Blue	7	Light Blue
8	Brown	9	Orange	10	Grey 2	11	Pink
12	Light Green	13	Yellow	14	Aqua	15	White

Hi-Res Graphics Colors:

0	Black	1	Green	2	Purple	3	White
4	Black	5	Orange	6	Blue	7	White

AppleCommander Command Line Options:

-i <imagename> [<imagename>] display information about image(s).
-ls <imagename> [<imagename>] list brief directory of image(s).
-l <imagename> [<imagename>] list directory of image(s).
-ll <imagename> [<imagename>] list detailed directory of image(s).
-e <imagename> <filename> [<output>] export file from image to stdout or to an output file.
-x <imagename> [<directory>] extract all files from image to directory.
-g <imagename> <filename> [<output>] get raw file from image to stdout or to an output file.
-p <imagename> <filename> <type> [[$|0x]<addr>] put stdin in filename on image, using file type and address [0x2000].
-d <imagename> <filename> delete file from image.
-k <imagename> <filename> lock file on image.
-u <imagename> <filename> unlock file on image.
-n <imagename> <volname> change volume name (ProDOS or Pascal).
-cc65 <imagename> <filename> <type> put stdin with cc65 header in filename on image
-geos <imagename> interpret stdin as a GEOS conversion file and place it on image (ProDOS only).
-dos140 <imagename> create a 140K DOS 3.3 image.
-pro140 <imagename> <volname> create a 140K ProDOS image.
-pro800 <imagename> <volname> create an 800K ProDOS image.
-pas140 <imagename> <volname> create a 140K Pascal image.
-pas800 <imagename> <volname> create an 800K Pascal image.
-convert <filename> <imagename> [<sizeblocks>] uncompress a ShrinkIt or Binary II file or convert a DiskCopy 4.2 image into a ProDOS disk image.

Using ADT Pro with Apple //c:

Serial speed: 9600bps, 8N1, no h/w or s/w handshaking.
Initialize serial port: IN#2 <Control>A 14 B
Serial file transfer from Linux: ascii-xfer -s -p 50 -c 10 *filename* >/dev/tty/USB0