




Color Computer

Memory Map By Bob

Russell Version 2.0

10/01/83



This memory map has been created after many hours of research ,investigation and experimentation. It is too bad Radio Shack (RS) does not publish an accurate map like this for our use. I probably have over 650 hrs into this project and I know I am not finished. I plan to offer an update to this document at least yearly. If you are interested in being informed when a new release is available please complete and mail the enclosed postcard. Also, I hope this will become a living growing document that you will also contribute to. I also am aware that there are probably some errors in this map. I hope you will bring them to my attention and also share any other unmapped discoveries you may have made. To encourage you to do so I offer a 40% discount on the next release version you purchase for notification of errors and omissions.

My experience in utilizing Radio Shack information has been often very frustrating. Information in general and especially memory references made by RS's Getting Started With Basic, Going Ahead with Color Basic, and TRS-80 Microcomputer News are frequently incorrect. However RS's Service Manual is much more exact and reliable. (A recommended purchase for the serious Color Computer user). Two other sources of information also have been much more exact and reliable. They are magazines available for your purchase. These magazines are Color Computer News and The Rainbow (also recommended purchases). If you refer to these resources you will most often find much more detailed information (sometimes even a commented listing that can be matched to a disassembled listing to really describe what is going on!!). Throughout the Map I will reference some of these resources. An *S* indicates a ROM or RAM based RS routines that you could potentially call from your own assembly language program. \$nnn is a notation for hexadecimal values.

- | | |
|---|--|
| [1] Getting Started With Color Basic
Copyright 1981 Tandy Corporation
Fort Worth, Texas 76102, U.S.A. | [5] Color Computer News
Remarkable Software
PO Box 1192
Muskegan, Mi. 49443 |
| [2] Going Ahead With Extended Basic
Copyright 1981 Tandy Corporation
Fort Worth, Texas 76102, U.S.A. | [6] The Rainbow - Falsoft, Inc.
5803 Timber Ridge Drive
Prospect Ky. 40059 |
| [3] Service Manual TRS-80 Color Computer
Catalog number 26-3001/3002 | [7] 80 Micro
P.O. Box 981
Farmingdale, NY. 11737 |
| [4] TRS-80 Microcomputer News
(Information published for TRS-80 users)
P.O. Box 2910, Fort Worth, Texas
76113-2910 | [8] Hot CoCo
P.O. Box 975
Farmingdale, NY. 11737 |

Happy Explorations,

Color Computer Memory Map:

While reasonable time and effort has been taken in preparation of this Memory Map to assure its accuracy, R. R. Enterprises assumes no liability resulting from any errors or omissions in this manual, or from the use of the information obtained herein.

Beginning DEC ADDR	Ending HEX ADDR	DEC ADDR	HEX ADDR	Description
00000	0000	32767	7FFF	RAM
00000	0000	01023	03FF	System Use
00003	0003			General Counter
00006	0006			String flag indicating variable type. \$00 = number \$FF = string
00007	0007			Garbage Collection Flag \$00 = not occurred \$FF = occurred
00008	0008			Array assignment Flag \$00 = assignment ok \$FF = assignment not allowed Cleared from \$AD43
00011	000B	00012	000C	Address of next available string descriptor reset at \$AD33
00025	0019	00026	001A	Address of start of BASIC program set by PCLEAR
00027	001B	00028	001C	Address of start of variable storage also address of end of BASIC program + 1. Note, I count the 3 zeros at the end of every program as part of the program. One zero comes from end of the last line in the program and the other 2 zeros act as a flag to signify end of program.
00029	001D	00030	001E	Address of start of arrays (subscripted variable storage) Also address of end of variable storage +1.
00031	001F	00032	0020	Address of beginning of free memory used in NEW command. Also address of end of arrays + 1.
00033	0021	00034	0022	Address of start of string pool (string stack)
00035	0023	00036	0024	Address of Basic limit. Also address of bottom of string stack.
00037	0025	00038	0026	Address of start of last string in string pool
00039	0027	00040	0028	Highest available ram address (end of string pool) set from CLEAR command
00041	0029	00042	002A	Continue line number displayed by execution of "STOP"
00043	002B	00044	002C	Temporary 2 byte storage area

Beginning DEC ADDR	Ending HEX ADDR	DEC ADDR	HEX ADDR	Description
00000	0000	32767	7FFF	RAM
00000	0000	01023	03FF	System Use Cont.
00045	002D	00046	002E	Address of position to return to after execution of CONT.
00047	002F	00048	0030	Address of current Basic program line the interpreter is executing.
00049	0031	00050	0032	Current Data statement line number (after read)
00051	0033	00052	0034	Pointer to comma after last data value read
00055	0037	00056	0038	Name of last variable used
00065	0041	00072	0048	Scratch area used by Tokenizing routine, garbage collection routine, move memory block routine
00065	0041	00072	0048	Data for Block Move
00065	0041	00066	0042	High end destination address
00067	0043	00068	0044	High end origin address
00069	0045	00070	0046	Low end destination address
00071	0047	00072	0048	Low end origin address
00079	004F	00084	0054	Floating point accumulator 1 (FPAC1)
00079	004F			Exponent
00080	0050	00083	0053	Mantissa
00084	0054			Sign
00086	0056			String variable length
00092	005C	00097	0061	Floating point accumulator 2 (FPAC2)
00092	005C			Exponent.
00093	005D	00096	0060	Mantissa
00097	0061			Sign
00098	0062			Sign comparison
00099	0063			Extended precision byte

Beginning DEC ADDR	Ending HEX ADDR	Description
00000	0000 32767 7FFF	RAM
00000	0000 01023 03FF	System Use Cont.
00104	0068 00105 0069	Current line number being interpreted. Will contain \$FFFF if illegal line number or if in command mode
00106	006A 00111 006F	Parameters set by definition of the output device.
00106	006A	Comma field width (defaults to 16)
00107	006B	Tab field width (defaults to 16)
00108	006C	Current column position (value of pos)
00109	006D	Number of characters in output line (if Device is set to 0 for screen this value is set to 32)
00110	006E	Output device number
00111	006F	Device DEVNUM[2] -2=Printer (\$FE) -1=Cassette (\$FF) 0=Screen 1-15=File for Disk BASIC
00112	0070	Input Buffer status flag 0=unpurged (something left) FF=purged (empty)
00113	0071	Warm start flag (RSTFLAG[2]) Ref[5]#22 pp 98-103 \$12= Do warm start \$0 = Condition before cartridge prog. starts created by BASIC \$55= IF RSTVEC[2] points to a NOP \$12 then control is transfered to address RSTVEC else BASIC starts up
00114	0072 00115 0073	Warm start vector address (RSTVEC[2]). At power up this contains \$80C0
00116	0074 00117 0075	Highest physical memory address determined by routine at \$A084
00120	0078	File mode set by OPEN or CLOSE statements 0 = closed 1 = open for input 2 = open for output

Beginning DEC ADDR	Ending DEC ADDR	Beginning HEX ADDR	Ending HEX ADDR	Description
00000	0000	32767	7FFF	RAM
00000	0000	01023	03FF	System Use Cont.
00121	0079			Dynamic changing count of characters remaining in the cassette Block as bytes are moved from the block to the Input/Output Buffer.
00122	007A	00123	007B	Address of the Input/Output Buffer.
00124	007C			Block type BLKTYP[2] 0=file header 1=data \$FF=end of file
00125	007D			Bytes in block 0-255 BLKLEN[2] Z=1,A=CSRERR=0 for no errors Z=0,A=CSRERR=1 for checksum error Z=0,A=CSRERR=2 for memory error
00126	007E	00127	007F	Buffer address CBUFAD[2]. Also program end address + 1 after CLOADM.
00128	0080			Checksum
00129	0081			CSRERR[2] 1 = Checksum Error (Tape Error) 2 = Memory Error (load error)
00130	0082			General bit counter. Pulse counter for cassette input.
00131	0083			Pulse width timing count
00132	0084			Rise/fall flag (0=rise)
00133	0085			Last sine value output voltage
00134	0086			Pixel information stored from RESET routine (\$A8B1)
00135	0087			Last key pressed (character obtained by INKEY\$)
00136	0088	00137	0089	Address of current cursor location
00138	008A	00139	008B	2 bytes always containing zeros
00140	008C			Pitch of sound (frequency)
00141	008D	00142	008E	Duration of sound
00143	008F			Full wave threshold \$12

Beginning DEC ADDR	Ending HEX ADDR	Beginning DEC ADDR	Ending HEX ADDR	Description																																												
00000	0000	32767	7FFF	RAM																																												
00000	0000	01023	03FF	System Use Cont.																																												
00144	0090			Upper allowable pulse width limit. Default = \$18 for cassette input.																																												
00145	0091			Lower allowable pulse width limit. Default = \$0A for cassette input.																																												
00146	0092			Gap length \$80. Controls length of unmodulated carrier preceding I/O Ref. [6]Vol II No. 2																																												
00148	0094			Cursor color flash count down timer (poke 0 to make cursor disappear)																																												
00149	0095	00156	009C	Printer variables																																												
00149	0095	00150	0096	RS-232 baud rate LPTBTD[2]																																												
				<table border="1"> <thead> <tr> <th>HEX</th> <th>MSB,</th> <th>LSB(Decimal)</th> <th>Baud</th> </tr> </thead> <tbody> <tr> <td></td> <td>149</td> <td>150</td> <td></td> </tr> <tr> <td>\$02EB</td> <td>2</td> <td>,235</td> <td>75</td> </tr> <tr> <td>\$01CA</td> <td>1</td> <td>,202</td> <td>120</td> </tr> <tr> <td>\$0173</td> <td>1</td> <td>,115</td> <td>150</td> </tr> <tr> <td>\$00BE</td> <td>0</td> <td>,180</td> <td>300</td> </tr> <tr> <td>\$0057</td> <td>0</td> <td>, 87</td> <td>600 Default</td> </tr> <tr> <td>\$0029</td> <td>0</td> <td>, 41</td> <td>1200</td> </tr> <tr> <td>\$0012</td> <td>0</td> <td>, 18</td> <td>2400</td> </tr> <tr> <td>\$0006</td> <td>0</td> <td>, 6</td> <td>4800</td> </tr> <tr> <td>\$0001</td> <td>0</td> <td>, 1</td> <td>9600</td> </tr> </tbody> </table>	HEX	MSB,	LSB(Decimal)	Baud		149	150		\$02EB	2	,235	75	\$01CA	1	,202	120	\$0173	1	,115	150	\$00BE	0	,180	300	\$0057	0	, 87	600 Default	\$0029	0	, 41	1200	\$0012	0	, 18	2400	\$0006	0	, 6	4800	\$0001	0	, 1	9600
HEX	MSB,	LSB(Decimal)	Baud																																													
	149	150																																														
\$02EB	2	,235	75																																													
\$01CA	1	,202	120																																													
\$0173	1	,115	150																																													
\$00BE	0	,180	300																																													
\$0057	0	, 87	600 Default																																													
\$0029	0	, 41	1200																																													
\$0012	0	, 18	2400																																													
\$0006	0	, 6	4800																																													
\$0001	0	, 1	9600																																													
00151	0097	00152	0098	RS-232 line return delay LPTLND[2]																																												
				<table border="1"> <thead> <tr> <th>HEX</th> <th>MSB,</th> <th>LSB(Decimal)</th> <th>Delay in sec.</th> </tr> </thead> <tbody> <tr> <td></td> <td>151</td> <td>152</td> <td></td> </tr> <tr> <td>\$0001</td> <td>0,</td> <td>1</td> <td>0 Default</td> </tr> <tr> <td>\$4000</td> <td>64,</td> <td>0</td> <td>.288</td> </tr> <tr> <td>\$8000</td> <td>128,</td> <td>0</td> <td>.576</td> </tr> <tr> <td>\$FFFF</td> <td>255,</td> <td>255</td> <td>1.15</td> </tr> </tbody> </table>	HEX	MSB,	LSB(Decimal)	Delay in sec.		151	152		\$0001	0,	1	0 Default	\$4000	64,	0	.288	\$8000	128,	0	.576	\$FFFF	255,	255	1.15																				
HEX	MSB,	LSB(Decimal)	Delay in sec.																																													
	151	152																																														
\$0001	0,	1	0 Default																																													
\$4000	64,	0	.288																																													
\$8000	128,	0	.576																																													
\$FFFF	255,	255	1.15																																													
00153	0099			RS-232 comma field width \$10 OR 16 DEFAULT																																												
00154	009A			RS-232 last comma field width \$70 OR 112 default																																												

Beginning DEC ADDR	Ending HEX ADDR	Beginning DEC ADDR	Ending HEX ADDR	Description																		
00000	0000	32767	7FFF	RAM																		
00000	0000	01023	03FF	System Use Cont.																		
00149	0095	00156	009C	Printer variables Cont.																		
00155	009B			RS-232 line printer width. It appears that Radioshack has a bug in BASIC 1.1 and extended BASIC 1.0 in that you can not set the line printer width by poking the width into 9B. The routine does delay for a carriage return but does not execute a carriage return.																		
				<table border="1"> <thead> <tr> <th>HEX</th> <th>(Decimal)</th> <th>char/line</th> </tr> </thead> <tbody> <tr> <td>\$10</td> <td>16</td> <td>16</td> </tr> <tr> <td>\$20</td> <td>32</td> <td>32</td> </tr> <tr> <td>\$40</td> <td>64</td> <td>64</td> </tr> <tr> <td>\$84</td> <td>132</td> <td>132 Default</td> </tr> <tr> <td>\$FF</td> <td>255</td> <td>255</td> </tr> </tbody> </table>	HEX	(Decimal)	char/line	\$10	16	16	\$20	32	32	\$40	64	64	\$84	132	132 Default	\$FF	255	255
HEX	(Decimal)	char/line																				
\$10	16	16																				
\$20	32	32																				
\$40	64	64																				
\$84	132	132 Default																				
\$FF	255	255																				
00156	009C			Line printer head position LPTPOS[2]																		
00157	009D	00158	009E	Transfer or execution address after CLOADM. Address jumped to by execution of EXEC command. Ref. [6]Vol. II No 2.																		
00159	009F	00168	00A8	*S* Get next character routine. A JSR \$009F will get the next byte. JSR \$00A5 will get the same byte. Ref [5] #12																		
00166	00A6	00167	00A7	Next character pointer																		
00168	00A8	00170	00AA	Vector to 43376. Contains \$7EAA1A (jump AA1A) to print "OK"																		
00171	00AB	00174	00AE	Extended product area 32 bits of a 64																		
00175	00AF			Trace Flag 0 = TRON 76 = TROFF																		
00176	00B0	00177	00B1	Address of start of DEFUSR table. \$013E																		
00178	00B2	00179	00B3	Foreground Color																		
00180	00B4			Color																		
00181	00B5	00219	00DB	Graphics data/constants																		
00181	00B5			Current color; if 4 color mode legal values are \$00,\$55,\$AA,\$FF; if 2 color mode legal values are \$00 or \$FF																		

Beginning DEC ADDR	Ending DEC ADDR	Beginning HEX ADDR	Ending HEX ADDR	Description
00000	0000	32767	7FFF	RAM
00000	0000	01023	03FF	System Use Cont.
00181	00B5	00219	00DB	Graphics data/constants cont.
00182	00B6			Current PMODE
00183	00B7	00184	00B8	Address of end of graphics page
00185	00B9			Number of bytes per line
00186	00BA			Address of top of graphics page
00188	00BC			Graphics page base address. Contains \$0E if Disk system else \$06. Non disk system can start graphics at \$0600 but a disk system uses page 1 & 2 and thus graphic must start after \$0E00.
00189	00BD	00202	00CA	Line Data
00189	00BD	00190	00BE	X1 line starting X coordinate.
00191	00BF	00192	00C0	Y1 line starting Y coordinate.
00193	00C1			Current CSS mask for control of PIA register bits at \$FF22. Value is either 0 or 8.
00194	00C2			Value is either 1 (indicating PSET or 0 indicating PRESET).
00195	00C3	00196	00C4	X2 line ending X coordinate.
00197	00C5	00198	00C6	Y2 line ending Y coordinate.
00199	00C7	00200	00C8	Previous line ending X coordinate.
00201	00C9	00202	00CA	Previous line ending Y coordinate.
00203	00CB	00218	00DA	Circle data
00203	00CB	00204	00CC	Center X coordinate.
00205	00CD	00206	00CE	Center Y coordinate.
00207	00CF	00208	00D0	Radius
00209	00D1	00210	00D2	H/W ratio
00217	00D9	00218	00DA	End point of a circle section
00219	00DB			Pixel Change flag

Beginning DEC ADDR	Ending DEC ADDR	Beginning HEX ADDR	Ending HEX ADDR	Description
00000	0000	32767	7FFF	RAM
00000	0000	01023	03FF	System Use Cont.
00222	00DE	00229	00E5	PLAY command data/constants
00222	00DE			Octave. "O" parameter-1. Default value is 2 or "O3".
00223	00DF	00224	00E0	Volume. "Vn" parameter.
00223	00DF			MSB of "V" (126+n*4). Default value is 186 or V15 (186=126+15*4)
00224	00E0			LSB of "V" (126-n*4). Default value is 66 or V15 (126-15*4).
00225	00E1			Note length "L". Default value is 4 or L4
00226	00E2			Tempo "T". Default value is 2 or T2. Must be value 0-255
00227	00E3			Note Timer
00229	00E5			Number of Dots after note length. Default value is 0.
00230	00E6			Baud rate constant
00231	00E7			Input timeout constant
00232	00E8			Angle Parameter of DRAW command
00233	00E9			Scale parameter of DRAW command
00234	00EA	00239	00EF	Disk I/O variables [5] #9
00234	00EA			Disk operation code (0-3)
00235	00EB			Drive number (0-3)
00236	00EC			Track number (0-34)
00237	00ED			Sector number (1-18)
00238	00EE	00239	00EF	Buffer address
00240	00F0			Disk status returned. I/O error codes

Beginning		Ending		Description
DEC	HEX	DEC	HEX	
ADDR	ADDR	ADDR	ADDR	
00000	0000	32767	7FFF	RAM
00000	0000	01023	03FF	System Use Cont.
00256	0100	00273	0111	Interrupt vectors
00256	0100	00258	0102	SWI3 - Software Interrupt 3 called from \$FFF2 Execution of a SWI3 instruction (\$113F) will stack registers and jump here.
00259	0103	00261	0105	SWI2 - Software Interrupt 2 called from \$FFF4 Execution of a SWI2 instruction (\$103F) will stack registers and jump here.
00262	0106	00264	0108	SWI1 - Software Interrupt 1 called from \$FFFA Execution of a SWI1 instruction (\$3F) will stack registers and jump here.
00265	0109	00267	010B	NMI - Non-Maskable Interrupt called from \$FFFC Set to \$7ED7AE (JMP \$D7AE) by initialization of Disk Operating System.
00268	010C	00270	010E	IRQ - Interrupt Request called from \$FFF8 Set to \$7EA9B3 (JMP \$A9B3) by initialization of Basic. Set to \$7E894C (JMP \$894C) by initialization of Extended Basic. Set to \$7ED7BC (JMP \$D7BC) by initialization of the Disk Operating System. This is primarily a timing interrupt and is used by SOUND & TIMER commands and to turn off Disk Motor when disk has not been accessed for several seconds.
00271	010F	00273	0111	FIRQ - Fast Interrupt Request called from \$FFF6. Set to \$7EA0F6 (JMP \$A0F6) by initialization of Basic. This causes a jump to the cartridge port in the expansion interface.
00274	0112	00276	0114	*S* Execution of "USR" Basic function (Token \$FF83) USRJMP jump to basic's usr routine[1] also 274-275 high and low order bytes of TIMER value Ref. [6] Vol II No 2
00278	0116	00281	0119	Seed for RND function Ref. [6] Vol II No 2
00281	0119	00282	0120	Address of next file buffer that could be allocated
00282	011A			Keyboard alpha lock (upper or lower case selection) [1] 0 = not locked (lower case) 255 = locked (upper case)

Beginning DEC ADDR	Ending DEC ADDR	HEX DEC ADDR	HEX DEC ADDR	Description
00000	0000	32767	7FFF	RAM
00000	0000	01023	03FF	System Use Cont.
00283	011B	00284	011C	Keyboard delay constant[1]
00285	011D	00287	011F	Exponentiation operation Vector to 45509. Contains \$7E8489 (JMP \$8489)
00288	0120	00316	013C	Token Table Directory Ref. [5] No.7 and Ref. [7] December Anniversary Issue
00288	0120	00292	0124	Basic Commands
00288	0120			Number of BASIC command keywords in token table located at \$AA66-\$AB19 (\$35)
00289	0121	00290	0122	Address of BASIC Command Keyword Table (\$AA66)
00291	0123	00292	0124	Address of table of pointers to BASIC Command subroutine entry addresses (\$AB67)
00293	0125	00297	0129	Basic Functions
00293	0125			Number of BASIC function keywords in token table located at \$AB1A-\$AB66 (\$14)
00294	0126	00295	0127	Address of BASIC function Keyword table (\$AB1A)
00296	0128	00297	0129	Address of table of pointers to BASIC Function subroutine entry addresses (\$AA29)
00298	012A	00302	012E	Extended Basic Commands
00298	012A			Number of Extended BASIC Command keywords in token table located at \$8183-\$81EF (\$19)
00299	012B	00300	012C	Address of Extended Basic Command keyword table (\$8183)
00301	012D	00302	012E	Address of a subroutine that determines subroutine entry addresses for tokens \$B5 to \$CB and \$FF90 to \$FF9F. Can also be used as hook to add new commands to Extended or Disk Basics. Set to \$813C by initialization of Extended Basic. Set to \$CE2E by initialization of Disk Basic.

Beginning DEC ADDR	Ending HEX ADDR	Beginning DEC ADDR	Ending HEX ADDR	Description
00000	0000	32767	7FFF	RAM
00000	0000	01023	03FF	System Use Cont.
00288	0120	00316	013C	Token Table Directory Ref. [5] No.7 and Ref. [7] December Anniversary Issue Cont.
00303	012F	00307	0133	Extended Basic Functions
00303	012F			Number of Extended BASIC Function keywords in token table located at \$821E-\$8272 (\$E)
00304	0130	00305	0131	Address of Extended Basic Function keyword table (\$821E)
00306	0132	00307	0133	Address of a subroutine that determines subroutine entry addresses for tokens \$FF94 to \$FF9F with the exception of \$FF90 and \$FF9F. (\$8168)
00308	0134	00312	0138	Disk Basic Command
00308	0134			Number of tokens
00309	0135	00310	0136	Address of Keyword Table
00311	0137	00312	0138	Address of where to go to execute a token
00313	0138	00317	013C	Disk Basic Function
00313	0138			Number of tokens
00314	0139	00315	013A	Address of Keyword Table
00316	013B	00317	013C	Address of where to go to execute a token
00318	013E	00337	0151	DEFUSR Table. Definition of entry point for USR functions 0-9
00318	013E	00319	013F	Address of DEFUSR 0
00320	0140	00321	0141	Address of DEFUSR 1
00322	0142	00323	0143	Address of DEFUSR 2
00324	0144	00325	0145	Address of DEFUSR 3
00326	0146	00327	0147	Address of DEFUSR 4
00328	0148	00329	0149	Address of DEFUSR 5
00330	014A	00331	014B	Address of DEFUSR 6
00332	014C	00333	014D	Address of DEFUSR 7
00334	014E	00335	014F	Address of DEFUSR 8
00336	0150	00337	0151	Address of DEFUSR 9

Beginning	Ending			
DEC	HEX	DEC	HEX	
ADDR	ADDR	ADDR	ADDR	Description

00000 0000 32767 7FFF RAM

00000 0000 01023 03FF System Use Cont.

00338 0152 00345 0159 Keyboard rollover table [4] Oct 1981

	Bit 7	6	5	4	3	2	1	0	
00338 0152		ENTER	8	0	X	P	H	@	
00339 0153		CLEAR	9	1	Y	Q	I	A	
00340 0154		BREAK	:	2	Z	R	J	B	
00341 0155			;	3	~	S	K	C	~= UP ARROW
00342 0156			,	4	~	T	L	D	~= DN ARROW
00343 0157			-	5	~	U	M	E	~= LT ARROW
00344 0158			.	6	~	V	N	F	~= RT ARROW
00345 0159		SHIFT	/	7	~	W	O	G	~= BLANK

Dec val of contents 191 223 239 247 251 253 254
of 338-345 if key in
col. is pressed

1 1 1 1 1 1 1 1
Is the binary value or \$FF or 255
when no key is pressed

If right joystick fire button is pressed:
1 1 1 1 1 1 1 0
Is the binary value or \$FE or 254 at all
locations \$152-\$159 or 338-345

If left joystick fire button is pressed:
1 1 1 1 1 1 0 1
is the binary value or \$FD OR 253 at all
locations \$152-\$159 or 338-345

When a key or mutiple keys are
simultaneously pressed the 1's toggle to
zero bit at col positions corresponding
to table above

Example: 'A' key is pressed
1 1 1 1 1 1 1 0
is the binary val or \$FE OR 254 at
location \$153 or 339

Example: 'd' and '4' keys are pressed
simultaneously
1 1 1 0 1 1 1 0
is the binary val or \$EE OR 238 at
location \$156 OR 342

Beginning DEC ADDR	Ending HEX ADDR	Beginning DEC ADDR	Ending HEX ADDR	Description
00000	0000	32767	7FFF	RAM
00000	0000	01023	03FF	System Use Cont.
00346	015A	00349	015D	Joystick pot values[1]
00346	015A			Joystick 0 x position left
00347	015B			Joystick 0 y position left
00348	015C			Joystick 1 x position right
00349	015D			Joystick 1 y position right
00350	015E	00423	01A8	Hooks to Ram area from Roms. Unless otherwise defined assume Basic initializes hooks with \$39 (RTS) Ref. [5] #11
00350	015E			Hook to allow device numbers 1-16 to be opened to disk files. Called from \$A5F7. Set to \$7EC426 (JMP \$C426) by initialization of Disk Operating System. Comes here before OPEN
00353	0161	00255	0163	Hook to allow device numbers greater than 0 for disk. Called from \$A5B9. Set to \$7EC838 (JMP \$C838) by initialization of Disk Operating System. Comes here after device number is obtained.
00356	0164	00358	0166	Hook to return device parameters for disk files such as current position in record, tab length, etc. Called from A35F. Set to \$7EC843 (JMP \$C843) by initialization of Disk Operating System. Comes here before output parameters are obtained.
00359	0167	00361	0169	Hook to allow user interface of other types of terminals or printers with different protocols. This location is referenced during each print. Extended Basic changes graphic screen back to text and ignores output to device #3 used in DLOAD. Disk operating system enables output to disk files. Set to \$39 (RTS) by initialization of Basic. Set to \$7E8273 (JMP \$8273) by initialization of Extended Basic. Set to \$7ECB4A (JMP \$7ECB4A) by Disk Operating System. User may poke or load a JMP to user defined modified RS232 driver then exiting driver and executing the proper JMP depending on system being run. Comes here before printing. Ref. [5] #6 & #11

Beginning DEC ADDR	Ending HEX ADDR	Beginning DEC ADDR	Ending HEX ADDR	Description
00000	0000	32767	7FFF	RAM
00000	0000	01023	03FF	System Use Cont.
00350	015E	00423	01A8	Hooks to Ram area from Roms. Unless otherwise defined assume Basic initializes hooks with \$39 (RTS) Ref. [5] #11 Cont.
00362	016A	00364	016C	Hook to keyboard input. Set to \$7E8CF1 by initialization of Extended Basic. Comes here before input.
00365	016D	00367	016F	Hook to make sure that a device number is open for input. Called from \$A3ED. Set to \$7EC818 (JMP \$C818) by initialization of Disk Operating System.
00368	0170	00370	0172	Hook to Make sure that a device is open for output.
00371	0173	00373	0175	Hook to allow closing of all open files. Called from \$A426. Set to \$7ECA3B (JMP \$CA3B) by initialization of Disk Operating System. Called before CLOSE is performed.
00374	0176	00376	0178	Hook to allow closing of the file opened to a specified device number. Called from \$A42D. Set to \$7ECA4B (JMP \$CA4B) by initialization of Disk Operating System. Set to \$7E8286 by initialization of Extended Basic.
00377	0179	00379	017B	Hook to enable PRINTUSING statement. Called from \$B918. Set to \$7E8E90 (JMP \$8E90) by initialization of Extended Basic. Comes here during print.
00380	017C	00382	017E	Hook to enable INPUT statements from disk. Called from \$B061. Set to \$7ECC5B (JMP \$CC5B) by initialization of Disk Operating System. Comes here after input.
00383	017F	00385	0181	Hook to enable/disable BREAK key. BREAK key is disabled during disk output. Called from \$A549. Set to \$7EC859 (JMP \$C859) by initialization of Disk Operating System. Comes here before BREAK.

Beginning DEC ADDR	Ending HEX ADDR	Beginning DEC ADDR	Ending HEX ADDR	Description
00000	0000	32767	7FFF	RAM
00000	0000	01023	03FF	System Use Cont.
00350	015E	00423	01A8	Hooks to Ram area from Roms. Unless otherwise defined assume Basic initializes hooks with \$39 (RTS) Ref. [5] #11 Cont.
00386	0182	00388	0184	Hook to enable line input. Called from \$A390. Set to \$39 (RTS) by initialization of Disk Operating System. Comes here before keyboard input.
00389	0185	00391	0187	Hook to enable cleanup after loading an ASCII file. Called from \$A4BF. Set to \$7ECA36 (JMP \$CA36) by initialization of Disk Operating System.
00392	0188	00394	018A	Hook to enable check before end of disk file. Set to \$393939 by initialization of Extended Basic. Set to \$7EC860 (JMP \$C860) by initialization of Disk Operating System. Called from \$A5CE.
00395	018B	00397	018D	Hook to evaluate Extended Basic Operands and Functions. Called from \$B223. Set to \$7E8846 (JMP \$8846) by initialization of Extended Basic. Set to \$7ECDF6 (JMP \$CDF6) by initialization of Disk Operating System.
00398	018E	00400	0190	Hook to allow a user program to trap errors. Called from \$AC46. Set to \$39 (RTS) by initialization of Extended Basic. Set to \$7EC70D by initialization Disk Operating System.
00401	0191	00403	0193	Hook to print Extended and Disk Basic Error messages. Also closes files upon errors. Called from \$AC49. Set to \$7E88F0 (JMP \$88F0) by initialization of Extended Basic. Set to \$7EC24D (JMP \$C24D) by initialization of Disk Operating System.
00404	0194	00406	0196	Hook to set up Graphics parameters for Extended Basic. Also enables RUN "filename" for Disk Basic. Called from \$AE75. Set to \$7E829C (JMP \$829C) by initialization of Extended Basic. Set to \$7EC990 (JMP \$C990) by initialization of Disk Operating System.

Beginning DEC ADDR	Ending HEX ADDR	Beginning DEC ADDR	Ending HEX ADDR	Description
00000	0000	32767	7FFF	RAM
00000	0000	01023	03FF	System Use Cont.
00350	015E	00423	01A8	Hooks to Ram area from Roms. Unless otherwise defined assume Basic initializes hooks with \$39 (RTS) Ref. [5] #11 Cont.
00407	0197	00409	0199	Hook to enable Hex (&H) and Octal (&O) conversions. Called from \$AD9E. Set to \$7E87E5 (JMP \$87E5) by initialization of Extended Basic.
00410	019A	00412	019C	Hook to allow user control of extended Basic interpreter. Called from \$AD9E after interpreting each Basic statement. Set to \$7E82B9 (JMP \$82B9) by initialization of Extended Basic. Purpose of Hook to Extended Basic is to enable TRACE function. It can be used to disable the BREAK key or SHIFT @ key. This speeds up Basic. Ref. [5] #6
00413	019D	00415	019F	Hook for Graphics. Called from \$A8C4.
00416	01A0	00418	01A2	Hook to allow Get and Put routines to access disk records. Called from \$8162,\$8AFA,\$975C,\$A910, and \$C29A. Set to \$7EC29A (JMP \$C29A).
00419	01A3	00421	01A5	Hook for Extended Basic interpreter tokenization Called from \$B821. Set to \$7E8304 (JMP \$8304) by initialization of Extended Basic.
00422	01A6	00424	01A8	Hook for Extended Basic interpreter untokenize. Called from \$B7C2.

Beginning		Ending		Description
DEC	HEX	DEC	HEX	
ADDR	ADDR	ADDR	ADDR	
00000	0000	32767	7FFF	RAM
00000	0000	01023	03FF	System Use Cont.
00465	01D1	00488	01E6	File I/O data
00465	01D1			Length of filename requested.
00466	01D2	00473	01D9	Cassette filename requested (8 bytes max)
00474	01DA	00731	02D8	Cassette buffer 255 bytes.
00474	01DA	00481	01E1	Filename found
00482	01E2			File Type 0 = Tokenized Basic 1 = Data 2 = Binary
00483	01E3	00484	01E4	File descriptor block
00483	01E3			0 = Data
00484	01E4			0 = Tokenized Basic or assembly language <>0 = Ascii
00485	01E5	00486	01E6	Transfer address used in CSAVEM command. Address of where an assembly language program begins execution.
00487	01E7	00488	01E8	Start address or load address of an assembly language program
00733	02DD	00988	03DC	Keyboard buffer 255 bytes (input terminated by 0). When data is tokenized it is stored starting at 00732 (\$02DC)
00737	02E1	00827	033B	Screen Buffer (90 bytes)

Beginning	Ending			
DEC	HEX	DEC	HEX	
ADDR	ADDR	ADDR	ADDR	Description

00000	0000	32767	7FFF	RAM
01024	0400	01535	05FF	Text screen memory (normal video display)[1]
01536	0600	13823	35FF	GRAPHICS PAGE AREA[2]
01536	0600	03071	0BFF	Page 1
03072	0C00	04607	11FF	Page 2
04608	1200	06143	17FF	Page 3
06144	1800	07679	1DFF	Page 4
07680	1E00	09215	23FF	Page 5
09216	2400	10751	29FF	Page 6
10752	2A00	12287	2FFF	Page 7
12288	3000	13823	35FF	Page 8
01536	0600	06350	18CE	Disk buffers, variable storage, and DOS variable storage (NOTE CONFLICT WITH GRAPHICS PAGES 1 - 4) Buffer space normally defaults to 256 bytes at startup but may be changed using the FILES command. File control blocks (FCB's) default to 843 bytes total at startup but can also be reset by the FILES command. Req. FCB bytes = (FILES+1)*281. Ref. [5] #9; [6] July 83 pp71-80
01536	0600	01791	06FF	Disk buffer for Disk I/O - 1 sector long - 256 bytes are reserved at startup but may be changed using FILES command.
01792	0700	02047	07FF	Disk buffer for Disk I/O - 1 sector long - 256 bytes are reserved at startup but may be changed using FILES command.
02048	0800	02343	0927	Disk drive table - 74 bytes data per drive
02048	0800	02121	0849	Drive 0 status table
02122	084A	02195	0893	Drive 1 status table
02196	0894	02269	08DD	Drive 2 status table
02270	08DE	02343	0927	Drive 3 status table

Beginning DEC ADDR	Ending HEX ADDR	Beginning DEC ADDR	Ending HEX ADDR	Description
00000	0000	32767	7FFF	RAM
01536	0600	06350	18CE	Disk buffers, variable storage, and DOS variable storage (NOTE CONFLICT WITH GRAPHICS PAGES 1 - 4) Buffer space normally defaults to 256 bytes at startup but may be changed using the FILES command. File control blocks (FCB's) default to 843 bytes total at startup but can also be reset by the FILES command. Req. FCB bytes = (FILES+1)*281. Ref. [5] #9:[6] July 83 pp71-80 Cont.
02344	0928	02349	029D	File buffer table
	02344	0928	02345 0929	Address of file buffer #1
	02346	092A	02347 092B	Address of file buffer #2
	02348	029C	02349 029D	Address of file buffer #3
02376	0948	02378	094A	Random access buffer
02380	094C			Disk filename storage area
02391	0957	02392	0958	File type \$0000 Basic Program \$00FF Basic Program ASCII format \$0100 Binary Data \$01FF ASCII Data \$0200 Machine language program \$0300 Binary Text \$03FF ASCII Text
02394	095A			Current Default drive number
02395	095B			Number of user allocated file buffers
02396	095C			Non 0 while doing DISKINI or BACKUP
02397	095D			Non 0 while doing LOAD
02399	095F			Disk Basic USR function Vectors
02426	097A			Count of Gran Tabl changes before rewrite

Beginning DEC ADDR	Ending HEX ADDR	Beginning DEC ADDR	Ending HEX ADDR	Description
00000	0000	32767	7FFF	RAM
01536	0600	06350	18CE	Disk buffers, variable storage, and DOS variable storage (NOTE CONFLICT WITH GRAPHICS PAGES 1 - 4) Buffer space normally defaults to 256 bytes at startup but may be changed using the FILES command. File control blocks (FCB's) default to 843 bytes total at startup but can also be reset by the FILES command. Req. FCB bytes = (FILES+1)*281. Ref. [5] #9:[6] July 83 pp71-80 Cont.
02430	097E	02438	0986	Disk variable storage
02430	097E			Start of table of current tracks
02434	0982			NMI in use flag (non 0 if NMI in use)
02435	0983			Address of return after NMI interrupt
02437	0985			Motor shutoff counter
02438	0986			Current Latch data - status of 1793 register in RAM
02459	0987			Verify Status 1 for on 0 for off
02460	0989			Random Access record buffer initially set to 256 bytes.
02978	0BA2	06350	18CE	Disk Buffers. For every file opened Disk BASIC allocates a 281 byte buffer. Last
02978	0BA2	03259	0CBB	1 file open.
03260	0CBC	03540	0DD4	2 files open.
03541	0DD5	03821	0EED	3 files open.
03822	0EEF	04102	1006	4 files open.
04103	1007	04383	111F	5 files open.
04384	1120	04664	1238	6 files open.
04665	1239	04945	1351	7 files open.
04946	1352	05226	146A	8 files open.
05227	146B	05507	1583	9 files open.
05508	1584	05788	169C	10 files open.
05789	169D	06069	17B5	11 files open.
06070	17B6	06350	18CE	12 files open.

Beginning		Ending		Description
DEC	HEX	DEC	HEX	
ADDR	ADDR	ADDR	ADDR	
00000	0000	32767	7FFF	RAM
13824	1536	32767	7FFF	Program and variable storage (program and variable storage could actually begin anywhere from 1536 to 13824 depending on number of graphics pages used; and if Disk Rom is in used, more storage is required for disk variables and disk buffers. The more files concurrently opened ,the less room for program and variable storage.

Beginning		Ending		Description
DEC	HEX	DEC	HEX	
ADDR	ADDR	ADDR	ADDR	
32768	8000	40959	9FFF	EXTENDED BASIC ROM[1]
32768	8000	32769	8001	Flag to Identify Extended Basic Rom
32770	8002	32957	80BD	Extended BASIC cold start routine
32960	80C0	32973	80CD	Extended BASIC warm start routine
32976	80D0	32989	80DD	Unused routine to try to set up 32k compatibility prior to release of updated BASIC ROM 1.1 Ref. [5] Aug 23, 1983.
32990	80DE			Number of Extended BASIC commands.
32991	80DF	32992	80E0	Address of start of Extended BASIC keyword - token table. (\$8183)
32993	80E1	32994	80E2	Address of Extended BASIC command processor.
32995	80E3			Number of Extended BASIC functions.
32996	80E4	32997	80E5	Address of start of Extended BASIC keyword - Function table. (\$821E)
32998	80E6	32999	80E7	Address of Extended BASIC function processor.
33000	80E8	33083	813B	Extended BASIC copyright notice.

Beginning Ending
 DEC HEX DEC HEX
 ADDR ADDR ADDR ADDR

Description

33155 8183 33263 81EF				Extended BASIC Command keyword - token table - ASCII Characters Ref. [7] December 1982 Anniversary Issue		
				CONTAINS	CONVERTED TO TOKEN	
				STRING	HEX	DEC
33155	8183	33157	8185	DEL	\$B5	181
33158	8186	33161	8189	EDIT	\$B6	182
33162	818A	33165	818D	TRON	\$B7	183
33166	818E	33170	8192	TROFF	\$B8	184
33171	8193	33173	8195	DEF	\$B9	185
33174	8196	33176	8198	LET	\$BA	186
33177	8199	33180	819C	LINE	\$BB	187
33181	819D	33184	81A0	PCLS	\$BC	188
33185	81A1	33188	81A4	PSET	\$BD	189
33189	81A5	33194	81AA	PRESET	\$BE	190
33195	81AB	33200	81B0	SCREEN	\$BF	191
33201	81B1	33206	81B6	PCLEAR	\$C0	192
33207	81B7	33211	81BB	COLOR	\$C1	193
33212	81BC	33217	81C1	CIRCLE	\$C2	194
33218	81C2	33222	81C6	PAINT	\$C3	195
33223	81C7	33225	81C9	GET	\$C4	196
33226	81CA	33228	81CC	PUT	\$C5	197
33229	81CD	33232	81D0	DRAW	\$C6	198
33233	81D1	33237	81D5	PCOPY	\$C7	199
33238	81D6	33242	81D7	PMODE	\$C8	200
33243	81D8	33246	81DE	PLAY	\$C9	201
33247	81DF	33251	81E3	DLOAD	\$CA	202
33252	81E4	33256	81E8	RENUM	\$CB	203
33257	81E9	33258	81EA	FN	\$CC	204
33259	81EB	55263	81EF	USING	\$CD	205

Note that FN and USING
 have tokens but are not in
 the table of pointers
 \$81F0-821D

Beginning DEC	Ending HEX	Beginning DEC	Ending HEX	Description
ADDR	ADDR	ADDR	ADDR	
32768	8000	40959	9FFF	EXTENDED BASIC ROM Cont. [1]
33088	81F0	33309	821D	Table of Pointers to subroutine entry addresses associated with Extended BASIC Command Keyword Table defined at \$8183 to \$81EF. Ref. [7] December 1982 Anniversary Issue.

	Contents	For Command
33088 81F0	\$8970	DEL
33090 81F2	\$8533	EDIT
33092 81F4	\$86A7	TRON
33094 81F6	\$86A8	TROFF
33096 81F8	\$8871	DEF
33098 81FA	\$AF89	LET
33100 81FC	\$93BB	LINE
33102 81FE	\$9532	PCLS
33104 8200	\$9361	PSET
33106 8202	\$9365	PRESET
33108 8204	\$9670	SCREEN
33110 8206	\$968B	PCLEAR
33112 8208	\$9546	COLOR
33114 820A	\$9E9D	CIRCLE
33116 820C	\$98EC	PAINT
33118 820E	\$9755	GET
33120 8210	\$9758	PUT
33122 8212	\$9CB6	DRAW
33124 8214	\$9723	PCOPY
33126 8216	\$9621	PMODE
33128 8218	\$9A22	PLAY
33130 821A	\$8C18	DLOAD
33132 821C	\$8A09	RENUM

33310	821E	33366	8256	Extended BASIC Function keyword - token table - ASCII characters. Ref. [7] December 1982 Anniversary Issue.
-------	------	-------	------	---

	CONTAINS	CONVERTED TO TOKEN		
	STRING	HEX	DEC(MSB)	DEC(LSB)
33310 821E 33312 8221	ATN	\$FF94	255	148
33313 8222 33315 8223	COS	\$FF95	255	149
33316 8224 33318 8226	TAN	\$FF96	255	150
33319 8227 33321 8229	EXP	\$FF97	255	151
33322 822A 33324 822C	FIX	\$FF98	255	152
33325 822D 33327 822F	LOG	\$FF99	255	153
33328 8230 33330 8232	POS	\$FF9A	255	154
33331 8233 33333 8235	SQR	\$FF9B	255	155
33334 8236 33337 8239	HEX\$	\$FF9C	255	156
33338 823A 33343 823F	VARPTR	\$FF9D	255	157
33344 8240 33348 8244	INSTR	\$FF9E	255	158
33349 8245 33353 8249	TIMER	\$FF9F	255	159
33354 824A 33359 824F	PPOINT	\$FFA0	255	160
33360 8250 33366 8256	STRINGS	\$FFA1	255	161

Beginning DEC ADDR	Ending HEX ADDR	Beginning DEC ADDR	Ending HEX ADDR	Description																														
32768	8000	40959	9FFF	EXTENDED BASIC ROM Cont. [1]																														
33367	8257	33394	8272	Table of Pointers to subroutine entry addresses associated with Extended BASIC Function Keyword Table defined at \$821E to \$8256. Ref. [7] Dec 1982 Anniversary Issue.																														
				<table border="1"> <thead> <tr> <th>Contents</th> <th>For Function</th> </tr> </thead> <tbody> <tr> <td>\$83B0</td> <td>ATN</td> </tr> <tr> <td>\$8378</td> <td>COS</td> </tr> <tr> <td>\$8381</td> <td>TAN</td> </tr> <tr> <td>\$84F2</td> <td>EXP</td> </tr> <tr> <td>\$8524</td> <td>FIX</td> </tr> <tr> <td>\$8446</td> <td>LOG</td> </tr> <tr> <td>\$86AC</td> <td>POS</td> </tr> <tr> <td>\$8480</td> <td>SQR</td> </tr> <tr> <td>\$8BDD</td> <td>HEXS</td> </tr> <tr> <td>\$86BE</td> <td>VARPTR</td> </tr> <tr> <td>\$877E</td> <td>INSTR</td> </tr> <tr> <td>\$8968</td> <td>TIMER</td> </tr> <tr> <td>\$9339</td> <td>PPOINT</td> </tr> <tr> <td>\$874E</td> <td>STRINGS</td> </tr> </tbody> </table>	Contents	For Function	\$83B0	ATN	\$8378	COS	\$8381	TAN	\$84F2	EXP	\$8524	FIX	\$8446	LOG	\$86AC	POS	\$8480	SQR	\$8BDD	HEXS	\$86BE	VARPTR	\$877E	INSTR	\$8968	TIMER	\$9339	PPOINT	\$874E	STRINGS
Contents	For Function																																	
\$83B0	ATN																																	
\$8378	COS																																	
\$8381	TAN																																	
\$84F2	EXP																																	
\$8524	FIX																																	
\$8446	LOG																																	
\$86AC	POS																																	
\$8480	SQR																																	
\$8BDD	HEXS																																	
\$86BE	VARPTR																																	
\$877E	INSTR																																	
\$8968	TIMER																																	
\$9339	PPOINT																																	
\$874E	STRINGS																																	
33395	8273			*S* Reset of video display registers to stay in text mode and check for DLOAD																														
33436	829C	33464	82B8	*S* Execution of Extended BASIC RUN setup. Set default play and draw values in RAM																														
33436	829C			Set up maximum volume for play command																														
33439	829F			Set value to mid volume																														
33441	82A1			Set Tempo to 2																														
33445	82A5			Set Octive to 2																														
33447	82A7			Set note length to 4																														
33450	82AA			Set dotted length to 0																														
33452	82AC			Set Angle and scale for draw to 0																														
33456	82B0			Set starting X,Y coordinates for DRAW, LINE,CIRCLE,etc to center screen (128,96)																														
33465	82B9	33566	831E	*S* Routine to look for BREAK or shift @ and do some byte fetching after interpreter is done with a BASIC statement Ref. [5] #6																														

Beginning DEC ADDR	Ending HEX ADDR	DEC ADDR	HEX ADDR	Description
32768	8000	40959	9FFF	EXTENDED BASIC ROM Cont. [1]
33467	82BB	33538	8302	*S* Interpret loop to enable trace function
33502	82DE	33521	82F1	*S* Trace Routine Ref. [8] Vol 1 No. 1 pp 63-64
33656	8378			*S* Execution of "COS" Extended BASIC function (Token \$FF95)
33665	8381			*S* Execution of "TAN" Extended BASIC function (Token \$FF96)
33712	83B0			*S* Execution of "ATN" Extended Basic function (Token \$FF94)
33862	8446			*S* Execution of "LOG" Extended BASIC function (Token \$FF99)
33920	8480			*S* Execution of "SQR" Extended BASIC function (Token \$FF9B)
34034	84F2			*S* Execution of "EXP" Extended BASIC function (Token \$FF97)
34084	8524			*S* Execution of "FIX" Extended BASIC function (Token \$FF98)
34099	8533			*S* Execution of "EDIT" Extended BASIC command (Token \$B6)
34115	8543			*S* Edit Routine
34471	86A7			*S* Execution of "TRON" Extended BASIC command (Token \$B7)
34472	86A8			*S* Execution of "TROFF" Extended BASIC command (Token \$B8)
34476	86AC			*S* Execution of "POS" Extended BASIC function (Token \$FF9A)
34494	86BE			*S* Execution of "VARPTR" Extended BASIC function (Token \$FF9D)
34638	874E			*S* Execution of "STRING\$" Extended BASIC function (Token \$FFA1)
34686	877E			*S* Execution of "INSTR" Extended BASIC function (Token \$FF9E)
34929	8871			*S* Execution of "DEF" Extended BASIC command (Token \$B9)

Beginning DEC ADDR	Ending HEX ADDR	Description
32768	8000 40959 9FFF	EXTENDED BASIC ROM Cont. [1]
35056	88F0 35080 8908	*S* Extended BASIC's error processing routines Ref. [5] #21 p126-127
35083	890B	Extended BASIC's error code table.
35176	8968	*S* Execution of "TIMER" Extended BASIC function (Token \$FF9F)
35184	8970	*S* Execution of "DEL" Extended BASIC command (Token \$B5)
35337	8A09	*S* Execution of "RENUM" Extended BASIC command (Token \$CB)
35386	8A3A	*S* Entry to RENUM when all parameters are set
35805	8BDD	*S* Execution of "HEX\$" Extended BASIC function (Token \$FF9C)
35864	8C18 36282 8DBA	*S* Execution of "DLOAD" Extended BASIC command (Token \$CA). DLOADM Ref [5] #13. Note experience has indicated DLOADM does not work, DLOAD does, will try to determine error later.
36284	8DBC	*S* Input RS232 character Ref. [5] Vol 1 #4
36326	8DE6	*S* Get bit or timeout
36343	8DF7	*S* Delay one bit time
36358	8E06	*S* Send RS232 character
37519	928F 37772 938C	*S* Get line data and setup Graphics. Ref. [5] #8
37519	928F	Find byte/bit routine
37542	92A6	Set up for Pmodes 0,2,or 4
37570	92C2	Setup for Pmodes 1,or 3
37597	92DD	Table for 1 bit/point
37605	92E5	Table for 2 bit/point
37689	9339	*S* Execution of "PPOINT" Extended BASIC function (Token \$FFA0)
37729	9361	*S* Execution of "PSET" Extended BASIC command (Token \$BD)

Beginning		Ending		Description
DEC	HEX	DEC	HEX	
ADDR	ADDR	ADDR	ADDR	
32768	8000	40959	9FFF	EXTENDED BASIC ROM Cont. [1]
37733	9365			*S* Execution of "PRESET" Extended BASIC command (Token \$BE)
37819	93BB			*S* Execution of "LINE" Extended BASIC command (Token \$BB)
37956	9444			*S* Draw Horizontal line. Ref. [5] #8
37996	946C			*S* Draw Vertical line. Ref. [5] #8
38049	94A1	38177	9521	*S* Draw sloped line routine. Ref. [5] #8
	38114	94E2		Draw line loop
	38150	9506		Move right
	38157	950D		Move up
	38164	9514		Move left
	38171	951B		Move down
38194	9532			*S* Execution of "PCLS" Extended BASIC command (Token \$BC)
38214	9546			*S* Execution of "COLOR" Extended BASIC command (Token \$C1)
38316	95AC			*S* Returns normal text screen from graphics
38433	9621			*S* Execution of "PMODE" Extended BASIC command (Token \$C8)
38512	9670			*S* Execution of "SCREEN" Extended BASIC command (Token \$BF)
38539	968B			*S* Execution of "PCLEAR" Extended BASIC command (Token \$C0)

Beginning DEC	Ending HEX	Beginning DEC	Ending HEX	Description
ADDR	ADDR	ADDR	ADDR	
32768	80C0	40959	9FFF	EXTENDED BASIC ROM Cont. [1]
38630	96E6	38661	9705	Initialize default RAM system values
38630	96E6			Set program start pointer. PCLEAR 4
38634	96EA			Start graphics page 1 at \$0600
38638	96EE			Set PMODE 0,1. (Start graphics on Page 1 and set PMODE = 0.
38643	96F3			Set screen width to 16 bytes.
38647	96F7			Set Foreground color to 3.
38651	96FB			Set end of graphics page 1 at \$BFF
38655	96FF			Set up user program space
38657	9701			Set up \$0 for initial end of line
38672	9710			*S* Compare 2 point. Ref. [5] #8
38691	9723			*S* Execution of "PCOPY" Extended BASIC command (Token \$C7)
38741	9755			*S* Execution of "GET" Extended BASIC command (Token \$C4)
38744	9758			*S* Execution of "PUT" Extended BASIC command (Token \$C5)
39148	98EC			*S* Execution of "PAINT" Extended BASIC command (Token \$C3)
39458	9A22			*S* Execution of "PLAY" Extended BASIC command (Token \$C9)
40118	9CB6			*S* Execution of "DRAW" Extended BASIC command (Token \$C6)
40605	9E9D			*S* Execution of "CIRCLE" Extended BASIC command (Token \$C2)

Beginning DEC ADDR	Ending HEX ADDR	DEC ADDR	HEX ADDR	Description
40960	A000	49151	BFFF	BASIC ROM[1]
40961	A000			Address of routine to poll keyboard for a character POLCAT[2] (\$A1C1) Z=1,A=0 NO KEY Z=0,A=KEY FOR KEY
40962	A002			Address of routine to Output char in REG A to current output device (all registers but CC are preserved) CHROUT[1] (\$A2A7)
40964	A004			Address of routine to start cassette-turns on motor and read header/gaps (gets into bit sync) CSRDON[2] (\$A77C)
40966	A006			Address of routine to read a block from cassette into tape buffer BLKIN[2], cassette must be on and in bit sync X=CBUFAD[2]+BLKLEN[2] interrupts are masked U and Y are preserved all others modified (\$A70B)
40968	A008			Address of routine to write a block to cassette from tape buffer BLKOUT[2] X=CBUFAD[2]+BLKLEN[2] all registers modified (\$A7F4)
40970	A00A			Address of routine to sample joystick pots and store their values in \$15A (346) to \$15D (349). Y is preserved all others modified JOYIN[1] (\$A9DE)
40972	A00C			Address of routine to turn cassette on and write leader/gap/ \$55'S WRTLDR[2] (\$A7D8)
40974	A00E	41328	A170	Re-start routines. Ref.[5]#22pp98-102
40974	A00E	40998	A026	*S* Secondary restart routine enables the cartridge interrupt then determines whether cold or warm start is specified. A warm start is performed if the restart flag at \$71 contains \$55 and if the the value of the byte pointed to by the address at \$72-73 is \$12(NOP).
40999	A027	41069	A06D	*S* Primary restart routine. Performs reset when reset button is pressed. PIA and SAM are reinitialized and then a jump to the secondary reset routine \$A00E is executed. Function of reset button can be emulated in software by POKE 113,0:EXEC 40999

Beginning DEC ADDR	Ending HEX ADDR	Beginning DEC ADDR	Ending HEX ADDR	Description
40960	A000	49151	BFFF	BASIC ROM Cont. [1]
41076	A074	41191	A0E7	*S*Cold start routine
41076	A074	41125	A0A5	Initializes low Ram Scratch pad memory \$0000-\$3FFF, determines upper limit of RAM and stores this limit at \$74-\$75, does equivalent of CLEAR 200, and sets up BASIC's next line pointer.
41126	A0A6			Check for extended Disk ROM and branch; Basic ends up here after Extended Basic initializes and reconfigures the system to include Extended Basic functions. Then a check is made to see if Disk Basic ROM is present; if \$C000 contains \$44 and \$C001 contains \$4B then jump is made to \$C002
41163	A0CB			Check for extended Basic ROM and branch; a check is made to see if Extended Basic ROM is present; if \$8000 contains \$45 and \$8001 contains \$58 then jump is made to \$8002.
41175	A0D7			*S* Displays version of BASIC ROM \$A000-\$BFFF.
41181	A0DD	41191	A0E7	*S* set restart vector and flag for warm re-start go to main command mode idle loop
41192	A0E8			*S* Warm restart (pointers not reinitialized)
41203	A0F3			*S* Return to BASIC
41206	A0F6			*S* Default FIRQ handler
41224	A108			*S* Cartridge secondary restart routine
41229	A10D	41256	A128	BASIC Initialization data down loaded to Ram addresses \$008F-\$00AA.
41257	A129	41286	A145	BASIC Initialization data down loaded to RAM addresses \$010C-\$0129
41286	A146	41328	A170	Copyright notice Logo text
41329	A171			*S* Get character from keyboard or cassette and put into REG. A. Set bit 8=0 and parity off
41334	A176			*S* Get character from device specified by \$6F and put that character into REG. A. no modification
41350	A186			*S* Get character from cassette and put into REG. A

Beginning DEC ADDR	Ending HEX ADDR	Beginning DEC ADDR	Ending HEX ADDR	Description
40960	A000	49151	BFFF	BASIC ROM Cont. [1]
41369	A199			*S* Display cursor
41393	A1B1	41410	A1C0	*S* Display cursor and wait for input character from keyboard and put character into REG. A.
41409	A1C1	41414	A1C6	*S* Check for character at keyboard and flag status (8 bit =0 if character is not there; 8 bit =1 if character is there). Ref. [5] Vol 1 #3
41416	A1C8	41452	A1EC	*S* Check for character at keyboard and decode character. (called from \$A1C3)
41453	A1ED	41516	A22C	*S* Decode character input from keyboard. (called from \$A1E1)
41517	A22D	41527	A237	*S* check for shift key depressed. (Called from \$A203)
41528	A238	41581	A26D	Check keyboard (called from \$A21B)
41582	A26E	41601	A281	Keyboard Special character table
41582	A26E			\$5E (Up Arrow)
41583	A26F			\$5F (Shift Up Arrow)
41584	A270			\$0A (Down Arrow)
41585	A271			\$5B (Shift Down Arrow)
41586	A272			\$08 (Left Arrow)
41587	A273			\$15 (Shift Left Arrow)
41588	A274			\$09 (Right Arrow)
41589	A275			\$5D (Shift Right Arrow)
41590	A276			\$20 (Space)
41591	A277			\$20 (Space)
41592	A278			\$30 (Zero)
41593	A279			\$12 (Shift Zero)
41594	A27A			\$0D (Enter)
41595	A27B			\$0D (Shift Enter)
41596	A27C			\$0C (Clear)
41597	A27D			\$5C (Shift Clear)
41598	A27E			\$03 (Break)
41499	A27F			\$03 (Shift Break)
41500	A280			\$40 (@)
41501	A281			\$13 (Shift @)
41602	A282	41639	A2A7	*S* Character output routine determines current output device specified by \$6F, then outputs the character in reg A to that device.
41640	A2A8	41662	A2BE	Set parameters \$79-\$7E and perform cassette output

Beginning DEC ADDR	Ending HEX ADDR	Beginning DEC ADDR	Ending HEX ADDR	Description
40960	A000	49151	BFFF	BASIC ROM Cont. [1]
41606	A2BF	41722	A2FA	*S* Send character in REG. A. to current output device. Called as subroutine from \$A282. Ref. [5] Vol 1 #4 and [6] Vol 2 #8 p122-123
41723	A2FB	41737	A309	*S* Send bit (called from \$A2DD)
41730	A302			*S* Delay half bit time
41738	A30A	41822	A35E	*S* Screen Display routine. Ref.[6] April 83 p50
41823	A35F	41859	A383	*S* Routine to set output parameters for selected device in RAM memory \$6A to \$6F
41860	A384	41868	A38C	*S* Routine to set up parameters for cassette output
41869	A38D	41964	A3EC	*S* Subroutine clears screen then does line input to buffer.
41872	A390			*S* Line input entry point (no clear screen)
41965	A3ED	41989	A405	*S* Routine to assure proper file mode is set and file has properly been opened
41990	A406	42005	A415	*S* Routine to assure when device is set for cassette output that a file is open for output and the mode is correct
42006	A416	42044	A43C	*S* Execution of "CLOSE" BASIC command (Token \$9A)
42045	A43D	42059	A44B	*S* Assure any remaining characters in the output buffer are purged from the buffer and written to the output device (called by Close from \$A43B)
42060	A44C	42135	A497	*S* Execution of "CSAVE" BASIC command (Token \$98)
42136	A498	42237	A4FD	*S* Execution of "CLOAD" BASIC command (Token \$97)
42232	A4F8			*S* Display of "I/O Error"
42238	A4FE	42301	A53D	*S* Execution of "CLOADM" Called by CLOAD Command from \$A49C
42302	A53E	42323	A553	*S* Execution of "EXEC" BASIC command (Token \$A2)
42324	A554	42339	A563	Position cursor for PRINT@
42340	A564	42359	A577	*S* Execution of "INKEY\$" BASIC function (Token \$FF92)

Beginning DEC ADDR	Ending HEX ADDR	DEC ADDR	HEX ADDR	Description
40960	A000	49151	BFFF	BASIC ROM Cont. [1]
42360	A576	42401	A5A1	*S* Parse out filename from current position in input buffer. Used in conjunction with search for filename routine at \$A648
42394	A59A	42401	A5A1	*S* Transfer block X register holds starting address, B register holds number of bytes to move, U holds destination address.
42402	A5A2	42436	A5C4	*S* Parse out device number
42437	A5C5	42445	A5CD	*S* Parse out filename called by CLOAD from SA4A0 and by SKIPF from \$A5EC
42446	A5CE	42475	A5EB	*S* Execution of "EOF" Basic function (Token \$FF8C)
42476	A5EC	42485	A5F5	*S* Execution of "SKIPF" command (Token SA3)
42486	A5F6	42531	A623	*S* Execution of "OPEN" command (Token \$99)
42532	A624	42567	A647	*S* Open a tape file for input
42549	A635			*S* Read an ASCII file from tape
42568	A648	42583	A657	*S* Search tape for a filename
42584	A658	42624	A680	*S* Open a tape file for output
42625	A681	42762	A70A	*S* Find a file specified by filename \$1D2 - \$1D9
42750	A6FE	42762	A70A	*S* Blink Corner of the screen
42753	A701			*S* Read one record from tape with motor control (sync, read gap, read block)
42763	A70B	42824	A748	*S* Plain read block from cassette only; motor must be on and in bit sync; \$7C holds block type, \$7D holds blocksize; U and Y are preserved; if no errors Z is 1, A is 0, X = buffer start + block length; if checksum error Z is 0, A is 1, X points beyond bad address; if memory error Z is 0 and A is 2.
42825	A749	42836	A754	*S* Read one byte
42837	A755	42844	A75C	*S* Read one bit
42845	A75D	42859	A76B	Determine wavelength of cassette input
42860	A76C	42875	A77B	Time cassette input by watching for 0 crossover

Beginning DEC	Ending HEX	Beginning DEC	Ending HEX	Description
ADDR	ADDR	ADDR	ADDR	
40960	A000	49151	BFFF	BASIC ROM Cont. [1]
42876	A77C	42902	A796	*S* Turn on tape motor synchronize data (read gap/\$55's); U and Y are preserved; FIRQ and IRQ are masked
42903	A797	42924	A7AC	Calculate Time Till next falling edge. Called from \$A786
42925	A7AD	42937	A7B9	Check that pulse width is between allowable limits. Called from \$A784
42938	A7BA	42940	A7BC	Clear pulse counter. Called from \$A7B5.
42941	A7BD	42967	A7D7	*S* Execution of "MOTOR" command (Token \$9F)
42954	A7CA			*S* Turn on tape motor
42961	A7D1	42967	A7D7	*S* 500 millisecond delay
42963	A7D3	42967	A7D7	*S* Entry point to the delay routine where you can set the delay by the magnitude of a value you pass in the X register. An RTS will be processed when countdown to 0 is reached.
42968	A7D8	42980	A7E4	*S* Turn on tape motor and write leader/gap/\$55's
42981	A7E5	42995	A7F3	*S* Write one record to tape with motor control
42985	A7E9	42995	A7F3	*S* Turn off tape motor
42996	A7F4	43046	A826	*S* Write Block; tape should be up to speed and leader of \$55's already written in 1st block; \$7E contains the buffer address; \$7C contains the block type; \$7D contains number of data bytes in block; X= buffer address + number of data bytes; all registers modified
43050	A82A			*S* Write one byte. Called by Write Block Routine from \$A80F,\$A813,\$A817,\$A81E

Beginning DEC ADDR	Ending HEX ADDR	DEC ADDR	HEX ADDR	Description
40960	A000	49151	BFFF	BASIC ROM Cont. [1]
43100	A85C			Sine table for CSAVE.
				Hex Decimal
43100	A85C			\$82 130
43101	A85D			\$92 146
43102	A85E			\$AA 170
43103	A85F			\$BA 186
43104	A860			\$CA 202
43105	A861			\$DA 218
43106	A862			\$EA 234
43107	A863			\$F2 242
43108	A864			\$FA 250
43109	A865			\$FA 250
43110	A866			\$FA 250
43111	A867			\$F2 242
43112	A868			\$EA 234
43113	A869			\$DA 218
43114	A86A			\$CA 202
43115	A86B			\$BA 186
43116	A86C			\$AA 170
43117	A86D			\$92 146
43118	A86E			\$7A 122
43119	A86F			\$6A 106
43120	A870			\$52 82
43121	A871			\$42 66
43122	A872			\$32 50
43123	A873			\$22 34
43124	A874			\$12 18
43125	A875			\$0A 10
43126	A876			\$02 2
43127	A877			\$02 2
43128	A878			\$02 2
43129	A879			\$0A 10
43130	A87A			\$12 18
43131	A87B			\$22 10
43132	A87C			\$32 50
43133	A87D			\$42 66
43134	A87E			\$52 82
43135	A87F	43184	A8B0	*S* Execution of "SET" command (Token \$9C)
43185	A8B1	43252	A8F4	*S* Execution of "RESET" command (Token \$9D)
43253	A8F5	43279	A90F	*S* Execution of "POINT" Basic function (Token \$FF91)
43280	A910	43318	A936	*S* Execution of "CLS n " command (Token \$9E)
43304	A928	43318	A936	*S* Clear screen to blanks and home cursor Ref [6] April 83 p50

Beginning DEC ADDR	Ending HEX ADDR	Description
40960	A000 49151 BFFF	BASIC ROM Cont. [1]
43339	A94B 43318 A973	*S* Execution of "SOUND" command (Token \$A0)
43350	A956 43318 A973	*S* Generate a sound determined by pitch at \$8C and duration at \$8D
43380	A974 43396 A984	*S* Execution of "AUDIO OFF" called from \$A996
43397	A985 43407 A98F	Timed Output through D/A and level setting routines called by SOUND command \$A94B
43408	A990 43425 A9A1	*S* Execution of "AUDIO" command (Token \$A1)
43426	A9A2 43442 A9B2	*S* Select joystick routine. Sets or resets SEL1 and SEL2 on port A of PIA U8
43443	A9B3 43461 A9C5	IRQ
43462	A9C6 43545 AA19	*S* Execution of "JOYSTK" Basic function (Token \$FF8D)
43486	A9DE	*S* Read positions of joysticks
43546	AA1A 43560 AA28	*S* ROM part of get next character routine called from \$00A8. Clears carry flag if ':' Sets carry flag if ASCII 0-9. Skips blanks. Ref. [5] #12.
43561	AA29 43600 AA50	Table of Pointers to subroutine entry addresses associated with Basic Function Keyword Table defined at \$AB1A to AB66. Ref. [7] Dec 1982 Anniversary Issue.
		Contents For Function
43561	AA29	\$BC7A SGN
43563	AA2B	\$BCEE INT
43565	AA2D	\$BC93 ABS
43567	AA2F	\$0112 USR
43569	AA31	\$BF1F RND
43571	AA33	\$BF78 SIN
43573	AA35	\$B750 PEEK
43575	AA37	\$B681 LEN
43577	AA39	\$B4FD STR\$
43579	AA3B	\$B716 VAL
43581	AA3D	\$B6A0 ASC
43583	AA3F	\$B68C CHR\$
43585	AA41	\$A5CE EOF
43587	AA43	\$A9C6 JOYSTK
43589	AA45	\$B6AB LEFT\$
43591	AA47	\$B6C8 RIGHT\$
43593	AA49	\$B6CF MID\$
43595	AA4B	\$A8F5 POINT
43597	AA4D	\$A564 INKEY\$
43599	AA4F	\$B4EE MEM

Beginning DEC ADDR	Ending HEX ADDR	DEC ADDR	HEX ADDR	Description
40960	A000	49151	BFFF	BASIC ROM Cont. [1]
43622	AA66	43801	AB19	BASIC Command keyword - token table - ASCII Characters Ref. [7] December 1982 Anniversary Issue
				CONTAINS STRING
				CONVERTED TO TOKEN HEX DECIMAL
43622	AA66	43624	AA68	FOR \$80 128
43625	AA69	43626	AA6A	GO \$81 129
43627	AA6B	43629	AA6D	REM \$82 130
43630	AA6E			' \$83 131
43631	AA6F	43634	AA72	ELSE \$84 132
43635	AA73	43636	AA74	IF \$85 133
43637	AA75	43640	AA78	DATA \$86 134
43641	AA79	43645	AA7D	PRINT \$87 135
43646	AA7E	43647	AA7F	ON \$88 136
43648	AA80	43652	AA84	INPUT \$89 137
43653	AA85	43655	AA87	END \$8A 138
43656	AA88	43659	AA8B	NEXT \$8B 139
43660	AA8C	43662	AA8E	DIM \$8C 140
43663	AA8F	43666	AA92	READ \$8D 141
43667	AA93	43669	AA95	RUN \$8E 142
43670	AA96	43676	AA9C	RESTORE \$8F 143
43677	AA9D	43682	AAA2	RETURN \$90 144
43683	AAA3	43686	AAA6	STOP \$91 145
43687	AAA7	43690	AAA	POKE \$92 146
43691	AAAB	43694	AAAE	CONT \$93 147
43695	AAAF	43698	AAB2	LIST \$94 148
43699	AAB3	43703	AAB7	CLEAR \$95 149
43704	AAB8	43706	AABA	NEW \$96 150
43707	AABB	43711	AABF	CLOAD \$97 151
43712	AAC0	43716	AAC4	CSAVE \$98 152
43717	AAC5	43720	AAC8	OPEN \$99 153
43721	AAC9	43725	AACD	CLOSE \$9A 154
43726	AACE	43730	AAD2	LLIST \$9B 155
43731	AAD3	43733	AAD5	SET \$9C 156
43734	AAD6	43738	AADA	RESET \$9D 157
43739	AADB	43741	AADD	CLS \$9E 158
43742	AADE	43746	AAE2	MOTOR \$9F 159
43747	AAE3	43751	AAE7	SOUND \$A0 160
43752	AAE8	43756	AAEC	AUDIO \$A1 161
43757	AAED	43760	AAF0	EXEC \$A2 162
43761	AAF1	43765	AAF5	SKIPF \$A3 163
43766	AAF6	43769	AAF9	TAB(\$A4 164
43770	AAFA	43771	AAFB	TO \$A5 165
43772	AAFC	43774	AAFE	SUB \$A6 166
43775	AAFF	43778	AB02	THEN \$A7 167
43779	AB03	43781	AB05	NOT \$A8 168

Beginning Ending

DEC HEX DEC HEX

ADDR ADDR ADDR ADDR

Description

40960 A000 49151 BFFF BASIC ROM Cont. [1]

43622 AA60 43801 AB19 BASIC Command keyword - token table - ASCII Characters Ref. [7] December 1982 Anniversary Issue

				CONTAINS	CONVERTED TO TOKEN	
				STRING	HEX	DECIMAL
43782	AB06	43785	AB09	STEP	\$A9	169
43786	AB0A	43788	AB0C	OFF	\$AA	170
43789	AB0D			+	\$AB	171
43790	AB0E			-	\$AC	172
43791	AB0F			*	\$AD	173
43792	AB10			/	\$AE	174
43793	AB11			^	\$AF	175
43794	AB12	43896	AB14	AND	\$B0	176
43797	AB15	43898	AB16	OR	\$B1	177
43799	AB17			>	\$B2	178
43800	AB18			=	\$B3	179
43801	AB19			<	\$B4	180

43802 AB1A 43878 AB66 BASIC Function keyword - token table - ASCII Characters Ref. [7] December 1982 Anniversary Issue

				CONTAINS	CONVERTED TO TOKEN		
				STRING	HEX	DEC(MSB)	DEC(LSB)
43802	AB1A	43804	AB1C	SGN	\$FF80	255	128
43805	AB1D	43807	AB1F	INT	\$FF81	255	129
43808	AB20	43810	AB22	ABS	\$FF82	255	130
43811	AB23	43813	AB25	USR	\$FF83	255	131
43814	AB26	43816	AB28	RND	\$FF84	255	132
43817	AB29	43819	AB2B	SIN	\$FF85	255	133
43820	AB2C	43823	AB2F	PEEK	\$FF86	255	134
43824	AB30	43826	AB32	LEN	\$FF87	255	135
43827	AB33	43830	AB36	STR\$	\$FF88	255	136
43831	AB37	43833	AB39	VAL	\$FF89	255	137
43834	AB3A	43836	AB3C	ASC	\$FF8A	255	138
43837	AB3D	43840	AB40	CHR\$	\$FF8B	255	139
43841	AB41	43843	AB43	EOF	\$FF8C	255	140
43844	AB44	43849	AB49	JOYSTK	\$FF8D	255	141
43850	AB4A	43854	AB4E	LEFT\$	\$FF8E	255	142
43855	AB4F	43860	AB54	RIGHT\$	\$FF8F	255	143
43861	AB55	43864	AB58	MID\$	\$FF90	255	144
43865	AB59	43869	AB5D	POINT	\$FF91	255	145
43870	AB5E	43875	AB63	INKEY\$	\$FF92	255	146
43876	AB64	43878	AB66	MEM	\$FF93	255	147

Beginning	Ending			
DEC	HEX	DEC	HEX	
ADDR	ADDR	ADDR	ADDR	Description

40960 A000 49151 BFFF BASIC ROM[1] Cont.

43879 AB67 43950 ABAE Table of Pointers to subroutine entry addresses associated with BASIC Command Keyword Table defined at \$AA66 to \$AB19 Ref. [7] December 1982 Anniversary Issue.

		Contents	For Command
43879	AB67	\$AD47	FOR
43881	AB69	\$AE86	GO
43883	AB6B	\$AEE3	REM
43885	AB6D	\$AEE3	'
43887	AB6F	\$AEE3	ELSE
43889	AB71	\$AF14	IF
43891	AB73	\$AEE0	DATA
43893	AB75	\$B8F7	PRINT
43895	AB77	\$AF42	ON
43897	AB79	\$AFF5	INPUT
43899	AB7B	\$AE02	END
43901	AB7D	\$B0F8	NEXT
43903	AB7F	\$B34E	DIM
43905	AB81	\$B046	READ
43907	AB83	\$AE75	RUN
43909	AB85	\$ADE4	RESTORE
43911	AB87	\$AEC0	RETURN
43913	AB89	\$AE09	STOP
43915	AB8B	\$B757	POKE
43917	AB8D	\$AE30	CONT
43919	AB8F	\$B764	LIST
43921	AB91	\$AE41	CLEAR
43923	AB93	\$AD17	NEW
43925	AB95	\$A498	CLOAD
43927	AB97	\$A44C	CSAVE
43929	AB99	\$A5F6	OPEN
43931	AB9B	\$A416	CLOSE
43933	AB9D	\$B75E	LLIST
43935	AB9F	\$A880	SET
43937	ABA1	\$A8B1	RESET
43939	ABA3	\$A910	CLS
43941	ABA5	\$A7BD	MOTOR
43943	ABA7	\$A94B	SOUND
43945	ABA9	\$A990	AUDIO
43947	ABAB	\$A53E	EXEC
43949	ABAD	\$A5EC	SKIPF

Beginning DEC ADDR	Ending HEX ADDR	Beginning DEC ADDR	Ending HEX ADDR	Description
40960	AC00	49151	BFFF	BASIC ROM Cont. [1]
43951	ABAF	44000	ABE0	BASIC Error Code Table ASCII text
43951	ABAF	43952	ABB0	NF
43953	ABB1	43954	ABB2	SN
43955	ABB3	43956	ABB4	RG
43957	ABB5	43958	ABB6	OD
43959	ABB7	43960	ABB8	FC
43961	ABB9	43962	ABBA	OV
43963	ABBB	43964	ABBC	OM
43965	ABBD	43966	ABBE	UL
43967	ABBF	43968	ABC0	RS
43969	ABC1	43970	ABC2	DD
43971	ABC3	43972	ABC4	/0
43973	ABC5	43974	ABC6	ID
43975	ABC7	43976	ABC8	TM
43977	ABC9	43978	ABCA	OS
43979	ABCB	43980	ABCC	LS
43981	ABCD	43982	ABCE	ST
43983	ABCF	43984	ABD0	CN
43985	ABD1	43986	ABD2	FD
43987	ABD3	43988	ABD4	AO
43989	ABD5	43990	ABD6	DN
43991	ABD7	43992	ABD8	IO
43993	ABD9	43994	ABDA	FM
43995	ABDB	43996	ABDC	NO
43997	ABDD	43998	ABDE	IE
43999	ABDF	44000	ABE0	DS
44001	ABE1	44006	ABE6	ASCII text 'ERROR'
44007	ABE7	44008	ABE8	\$0 THEN ASCII Space
44009	ABE9	44010	ABEA	ASCII text 'IN'
44011	ABEB	44012	ABEC	ASCII Space then \$0
44013	ABED	44017	ABF1	CR ASCII text 'OK' CR THEN \$0
44018	ABF2	44024	ABF8	CR ASCII TEXT 'BREAK' THEN \$0
44038	AC37			*S* Check if room under stack
44064	AC20			*S* Routine to move a block of memory; \$41-\$42 holds destination top address; \$43-\$44 holds source top address; \$45-\$46 holds destination bottom address; \$47-\$48 holds source bottom address.
44102	AC46	44146	AC72	*S* Error Processing routine. Ref. [5] #21 pl26-127

Beginning DEC ADDR	Ending HEX ADDR	Beginning DEC ADDR	Ending HEX ADDR	Description
40960	A000	49151	BFFF	BASIC ROM Cont. [1]
44147	AC73	44189	AC9D	*S* Entry to command level of BASIC. This is the idle loop. Print "OK", close files, tokenize line, then JMP \$ADCO. Ref. [5] #12
44156	AC7C			*S* Tokenize ASCII inputs
44271	ACEF			*S* Execution of a routine that relinks line link pointers (The first 2 bytes in a line point to the address of the start of the next line.)
44313	AD17	44357	AD45	*S* Execution of "NEW" BASIC command (Token \$96) Ref [5] #21 p126-127
44321	AD21			*S* Execution from this point clears all variables and moves the stack
44326	AD26			*S* Execution from this point clears variables only
44359	AD47			*S* Execution of "FOR" BASIC command (Token \$80)
44446	AD9E	44484	ADC4	*S* Interpret loop Ref. [5] #12
44486	ADC6	44542	ADFE	*S* Execute line Ref. [5] #12
44516	ADE4	44522	ADEA	*S* Execution of "RESTORE" BASIC command (Token \$8F) Ref. [5] #21 p 126-127
44539	ADFB			*S* Exec 44539 has performs equivalent of : 1 A\$=INKEY\$:IFA\$=""THEN1
44546	AE02			*S* Execution of "END" BASIC command (Token \$8A)
44553	AE09			*S* Execution of "STOP" BASIC command (Token \$91)
44592	AE30			*S* Execution of "CONT" BASIC command (Token \$93)
44609	AE41			*S* Execution of "CLEAR" BASIC command (Token \$95)
44661	AE75			*S* Execution of "RUN" BASIC command (Token \$8E)
44678	AE86			*S* Execution of "GO" BASIC command (Token \$81)
44736	AEC0			*S* Execution of "RETURN" BASIC command (Token \$90)
44768	AEE0			*S* Execution of "DATA" BASIC command (Token \$86)
44771	AEE3			*S* Execution of "REM or '" BASIC command (Token \$83)

Beginning DEC ADDR	Ending HEX ADDR	DEC ADDR	HEX ADDR	Description
40960	A000	49151	BFFF	BASIC ROM Cont. [1]
				S Execution of "ELSE" BASIC command (Token \$84)
44820	AF14			*S* Execution of "IF" BASIC command (Token \$85)
44866	AF42			*S* Execution of "ON" BASIC command (Token \$88)
44903	AF67			*S* Get next line number and place it in temporary scratch area \$2B-2C
44937	AF89			*S* Execution of "LET" Extended BASIC command (Token \$BA)
45045	AFF5			*S* Execution of "INPUT" BASIC command (Token \$89)
45126	B046			*S* Execution of "READ" BASIC command (Token \$8D)
45304	B0F8			*S* Execution of "NEXT" BASIC command (Token \$8B)
45677	B26D			Check for "," syntax
45679	B26F			Check for "A" syntax
45607	B277			*S* Get operand
45902	B34E			*S* Execution of "DIM" BASIC command (Token \$8C)
46052	B3E4			Put cursor position in D register (called by PRINT@ routine from \$B3E4)
46061	B3ED			*S* INTCNV[2] convert floating point number to a 16 bit 2's compliment integer; if number exceeds + or - 32768 an overflow error will occur and a return to Basic.
46154	B44A			*S* Display of "FC ERROR" message.
46324	B4F4			*S* Returns an integer to BASIC from D loaded with 2's compliment
46333	B4FD			*S* Execution of "STR\$" BASIC function (Token \$FF88)

Beginning DEC ADDR	Ending DEC ADDR	Hex DEC ADDR	Description
40960	A000	49151 BFFF	BASIC ROM Cont. [1]
46445	B56D	46605 B60C	Garbage collection routines Ref. [5] #10
46445	B56D		Allocate String in string buffer
46481	B591		Garbage Collection
46552	B5D8		Process a descriptor
46575	B5EF		Compact a string
46721	B681		*S* Execution of "LEN" BASIC function (Token \$FF87)
46732	B68C		*S* Execution of "CHR\$" BASIC function (Token \$FF8B)
46763	B6AB		*S* Execution of "LEFT\$" BASIC function (Token \$FF8E)
46764	B6AC		*S* Execution of "ASC" BASIC function (Token \$FF8A)
46792	B6C8		*S* Execution of "RIGHT\$" BASIC function (Token \$FF8F)
46799	B6CF		*S* Execution of "MID\$" BASIC function (Token \$FF90)
46859	B70B		*S* Execution of routine to evaluate argument of PCLEAR command. Argument should be present in B register.
46870	B716		*S* Execution of "VAL" BASIC function (Token \$FF89)
46904	B738		*S* Parse for ',' then return integer <256
46909	B73D		Determine offset load value for CLOADM command
46928	B750		*S* Execution of "PEEK" BASIC function (Token \$FF86)
46935	B757		*S* Execution of "POKE" BASIC command (Token \$92)
46942	B75E		*S* Execution of "LLIST" BASIC command (Token \$9B)
46948	B764		*S* Execution of "LIST" BASIC command (Token \$94)

Beginning DEC ADDR	Ending HEX ADDR	Beginning DEC ADDR	Ending HEX ADDR	Description
40960	A000	49151	BFFF	BASIC ROM Cont. [1]
47042	B7C2	47349	B8F5	Tokenize & Untokenize Routines Ref. [5] #7
47042	B7C2	47136	B820	*S* Untokenize Routine
47042	B7C2			RAM Hook from untokenize routine
47078	B7E6			Untokenize one token
47137	B821	47349	B8F5	*S* Tokenize Routine
47137	B821			RAM Hook from tokenize routine
47250	B892			Tokenize one word
47351	B8F7			*S* Execution of "PRINT" BASIC command (Token S87)
47532	B9AC			*S* Print a space
47545	B9B9			*S* Floating point subtract FPAC1=[X]-FPAC1
47554	B9C2			*S* Floating point add FPAC1=[X]+FPAC1
47818	BACA			*S* Floating point multiply FPAC1=[X]*FPAC1
48015	BB8F			*S* Floating point divide FPAC1=[X]/FPAC1
48250	BC7A			*S* Execution of "SGN" Basic function (Token \$FF80)
48275	BC93			*S* Execution of "ABS" Basic function (Token \$FF82)
48366	BCEE			*S* Execution of "INT" Basic function (Token \$FF81)
48588	BDCC			*S* Display decimal value in D
48927	BF1F			*S* Execution of "RND" Basic function (Token \$FF84)
49016	BF78			*S* Execution of "SIN" Basic function (Token \$FF85)
49138	BFF2	49151	BFFF	Interrupt and reset vectors
49138	BFF2	49139	BFF3	SWI3 Vector
49140	BFF4	49141	BFF5	SWI2 Vector
49142	BFF6	49143	BFF7	FIRQ Vector
49144	BFF8	49145	BFF9	IRQ Vector
49146	BFFA	49147	BFFB	SWI1 Vector
49148	BFFC	49149	BFFD	NMI Vector
49150	BFFE	49151	BFFF	RESET Vector

Beginning DEC	Ending HEX	Beginning DEC	Ending HEX	Description
ADDR	ADDR	ADDR	ADDR	
49152	C000	65279	FEEF	Cartridge ROM[1]/ Disk Operating System (DOS ends at \$D800)
49152	C000	49153	C001	Flag to identify Disk ROM is present. ASCII 'DK'
49156	C004			Address of DSKCON (\$D66C)
49158	C006			Address of DSKCON parameters
49160	C008			Cold start Disk Basic. (Actually starts at C002 and continues here)
49364	C0D4	49370	C0DA	Warm start Disk Basic
49373	C0DD	49397	C0F5	Reset Disk Drive routine
49446	C126	49534	C17E	Disk Basic Copyright notice
49535	C17F	49630	C1DA	Disk BASIC Command keyword - token table - ASCII Characters Ref. [7] December 1982 Anniversary Issue

				CONTAINS STRING	CONVERTED TO TOKEN HEX DECIMAL
49535	C17F	49537	C181	DIR	\$CE 206
49538	C182	49542	C186	DRIVE	\$CF 207
49543	C187	49547	C18B	FIELD	\$D0 208
49548	C18C	49552	C190	FILES	\$D1 209
49553	C191	49556	C194	KILL	\$D2 210
49557	C195	49560	C198	LOAD	\$D3 211
49561	C199	49564	C19C	LSET	\$D4 212
49565	C19D	49569	C1A1	MERGE	\$D5 213
49570	C1A2	49575	C1A7	RENAME	\$D6 214
49576	C1A8	49579	C1AB	RSET	\$D7 215
49580	C1AC	49583	C1AF	SAVE	\$D8 216
49584	C1B0	49588	C1B4	WRITE	\$D9 217
49589	C1B5	49594	C1BA	VERIFY	\$DA 218
49595	C1BB	49600	C1C0	UNLOAD	\$DB 219
49601	C1C1	49606	C1C6	DSKINI	\$DC 220
49607	C1C7	49612	C1CC	BACKUP	\$DD 221
49613	C1CD	49616	C1D0	COPY	\$DE 222
49617	C1D1	49621	C1D5	DSKI\$	\$DF 223
49622	C1D6	49626	C1DA	DSKO\$	\$E0 224

Beginning DEC ADDR	Ending HEX ADDR	Beginning DEC ADDR	Ending HEX ADDR	Description
49152	C000	65279	FEEF	Cartridge ROM[1]/ Disk Operating System (DOS ends at \$D800) Cont.

49627 C1DB 49665 C200 Table of Pointers to subroutine entry addresses associated with Disk BASIC Command Keyword Table defined at \$C17F to \$C200 Ref. [7] December 1982 Anniversary Issue.

	Contents	For Command
49627 C1DB	\$CBCF	DIR
49629 C1DF	SCDE9	DRIVE
49631 C1E1	\$CFEC	FIELD
49633 C1E3	\$D080	FILES
49635 C1E5	\$C6C2	KILL
49637 C1E7	SC99A	LOAD
49639 C1E9	\$DC26	LSET
49641 C1EB	\$C98B	MERGE
49643 C1ED	\$CF3F	RENAME
49645 C1EF	\$D025	RSET
49647 C1F1	\$C932	SAVE
49649 C1F3	\$CF8A	WRITE
49651 C1F5	\$D65B	VERIFY
49653 C1F7	\$D146	UNLOAD
49655 C1F9	\$D4AB	DSKINI
49657 C1FB	\$D175	BACKUP
49659 C1FD	\$D2CC	COPY
49661 C1FF	\$D3FF	DSKIS
49663 C200	\$D474	DSKOS

49665 C201 49683 C213 Disk BASIC Function keyword - token table - ASCII Characters

	CONTAINS	CONVERTED TO TOKEN		
	STRING	HEX	DEC(MSB)	DEC(LSB)
49665 C201 49667 C203	CVN	\$FFA2	255	162
49668 C204 49671 C207	FREE	\$FFA3	255	163
49672 C208 49674 C20A	LOC	\$FFA4	255	164
49675 C20B 49677 C20D	LOF	\$FFA5	255	165
49678 C20E 49681 C211	MKN\$	\$FFA6	255	166
49682 C212 49683 C213	AS	\$FFA7	255	167

(Note that AS is tokenized but is not in the table of pointers \$C214-C21D.

49684 C214 49693 C21D Table of Pointers to subroutine entry addresses associated with Disk BASIC Function Keyword Table defined at \$C201 to \$C213

	Contents	For Function
49684 C214	\$CD1A	CVN
49686 C216	\$CDC0	FREE
49688 C218	\$CD36	LOC
49690 C21A	\$CD5B	LOF
49692 C21C	\$CD28	MKN\$

Beginning DEC ADDR	Ending HEX ADDR	Beginning DEC ADDR	Ending HEX ADDR	Description
49152	C000	65279	FEEF	Cartridge ROM[1]/ Disk Operating System (DOS ends at \$D800) Cont.
49741	C24D	49783	C277	*S* Disk BASIC's error processing routine. Ref. [5] #21 pl27-128
49784	C278			Disk BASIC's error code table.
50280	C468			*S* OPEN file subroutine. Before calling this routine the filename should be processed by the routine at \$C8A4, the A reg should contain the file mode (\$49=ASCII "I" for input or \$4F=ASCII "O" for output), the B reg should contain the device number. Also \$957-958 should contain the file type.
50882	C6C2			*S* Execution of "KILL" Disk BASIC command (Token \$D2)
	50885	C6C5		*S* Entry point for KILL when you have set the filename you want to kill by executing the routine at C84A.
51364	C8A4			*S* Subroutine to process a filename. Prior to calling this routine \$EB should be set with the default drive number, and \$957 should contain the filename.
51506	C932			*S* Execution of "SAVE" Disk BASIC command (Token \$D8)
51595	C98B			*S* Execution of "MERGE" Disk BASIC command (Token \$D5)
51610	C99A			*S* Execution of "LOAD" Disk BASIC command (Token \$D3)
51771	CA3B			*S* performs close of a disk file
51795	CA53			*S* entry point for close when Device # at \$6F is set.
52175	CBCF			*S* Execution of "DIR" Disk BASIC command (Token \$CE)
52506	CD1A			*S* Execution of "CVN" Disk BASIC function (Token \$FFA2)
52520	CD28			*S* Execution of "MKN\$" Disk BASIC function (Token \$FFA6)
52534	CD36			*S* Execution of "LOC" Disk BASIC function (Token \$FFA4)

Beginning		Ending		Description
DEC	HEX	DEC	HEX	
ADDR	ADDR	ADDR	ADDR	
49152	C000	65279	FEEF	Cartridge ROM[1]/ Disk Operating System (DOS ends at \$D800) Cont.
52571	CD5B			*S* Execution of "LOF" Disk BASIC function (Token \$FFA5)
52672	CDC0			*S* Execution of "FREE" Disk BASIC function (Token \$FFA3)
52713	CDE9			*S* Execution of "DRIVE" Disk BASIC command (Token \$CF)
52876	CE8C			*S* Execution of "SAVEM" Disk BASIC command (tokenized to \$D84D Note \$4D is the letter "M")
52965	CEE5			*S* Execution of "LOADM" Disk BASIC command (tokenized to \$D34D Note \$4D is the letter "M")
53055	CF3F			*S* Execution of "RENAME" Disk BASIC command (Token \$D6)
53130	CF8A			*S* Execution of "WRITE" Disk BASIC command (Token \$D9)
53216	CFE0			*S* Execution of "FIELD" Disk BASIC command (Token \$D0)
53285	D025			*S* Execution of "RSET" Disk BASIC command (Token \$D7)
53286	D026			*S* Execution of "LSET" Disk BASIC command (Token \$D4)
53376	D080			*S* Execution of "FILES" Disk BASIC command (Token \$D1)
53574	D146			*S* Execution of "UNLOAD" Disk BASIC command (Token \$DB)
53621	D175			*S* Execution of "BACKUP" Disk BASIC command (Token \$DD)
53733	D1E5	53774	D20E	*S* Cleanup after BACKUP, DSKINI, or LOAD command Ref. [5] #21 p126-127
53964	D2CC			*S* Execution of "COPY" Disk BASIC command (Token \$DE)
54271	D3FF			*S* Execution of "DSKIS" Disk BASIC command (Token \$DF)

Beginning		Ending		Description
DEC	HEX	DEC	HEX	
ADDR	ADDR	ADDR	ADDR	
49152	C000	65279	FEEF	Cartridge ROM[1]/ Disk Operating System (DOS ends at \$D800) Cont.
54388	D474			*S* Execution of "DSKOS" Disk BASIC command (Token \$E0)
54443	D4AB			*S* Execution of "DSKINI" Disk BASIC command (Token \$DC)
54875	D65B			*S* Execution of "VERIFY" Disk BASIC command (Token \$DA)
54892	D66C	55258	D7DA	DSKCON Read or write a 256 byte sector. Ref [5] #9
54981	D6C5	55005	D6DD	Restore
55006	D6DE	55036	D6FC	Get Status
55037	D6FD	55044	D704	Delay 78 mSEC
55045	D705	55201	D7A1	Read/write a sector
55202	D7A2	55209	D7A9	Command address table
55202	D7A2	55203	D7A3	Address of restore
55204	D7A4	55205	D7A5	Address of NOP (RTS)
55205	D7A6	55207	D7A7	Address of Read sector
55208	D7A8	55290	D7A9	Address of Write sector
55210	D7AA	55213	D7AD	Bit table for drives
55214	D7AE	55227	D7BB	NMI Handler
55228	D7BC	55258	D7DA	IRQ Handler

Beginning DEC ADDR	Ending HEX ADDR	Description
65280	FF00 65535 FFFF	INPUT/OUTPUT AREA[3]
65280	FF00 65283 FF03	PIA U8
65280	FF00	BIT 0 = Keyboard row 1 and right joystick switch BIT 1 = Keyboard row 2 and left joystick switch BIT 2 = Keyboard row 3 BIT 3 = Keyboard row 4 BIT 4 = Keyboard row 5 BIT 5 = Keyboard row 6 BIT 6 = Keyboard row 7 BIT 7 = Joystick comparison input Decimal value is 255 or 127 if no fire buttons are pressed. Decimal value is 126 or 254 if right joystick fire button is pressed. Decimal value is 125 or 253 if left joystick fire button is pressed.
65281	FF01	BIT 0 Control of horizontal 0 = IRQ to cpu enabled 1 = IRQ to cpu disabled BIT 1 Sync clock (63.5 usec) 0 = Flag set on falling edge of hs 1 = Flag set on rising edge of hs BIT 2 Normally 1 data I/O register is addressed 0 = Changes FF00 to the data dir register BIT 3 SEL 1: lsb of the two analog mux lines BIT 4 1 Always BIT 5 1 always BIT 6 Not used BIT 7 Horizontal sync interrupt flag
65282	FF02	BIT 0 = keyboard column 1 BIT 1 = keyboard column 2 BIT 2 = keyboard column 3 BIT 3 = keyboard column 4 BIT 4 = keyboard column 5 BIT 5 = keyboard column 6 BIT 6 = keyboard column 7 BIT 7 = keyboard column 8

Beginning Ending				Description
DEC	HEX	DEC	HEX	
ADDR	ADDR	ADDR	ADDR	
65280	FF00	65535	FFFF	INPUT/OUTPUT AREA[3] Cont.
65280	FF00	65283	FF03	PIA U8 Cont.
65283	FF03			BIT 0 = Control of Sync Clock 0 = IRQ To CPU disabled 1 = IRQ To CPU enabled BIT 1 = 16.667 MS Interrupt input 0 = Sets Flag on falling edge FS 1 = Sets Flag on rising edge FS BIT 2 = Normally 1 Data I/O Register is addressed 0 = Changes F002 to Data Direction Register BIT 3 = SEL2: (MSB of the two analog Mux lines) BIT 4 = 1 Always BIT 5 = 1 Always BIT 6 = Not used BIT 7 = Field Sync Interrupt Flag
65312	FF20	65315	FF23	PIA U4[3]
65312	FF20			BIT 0 = Cassette data input BIT 1 = RS232 data output BIT 2-7 = 6 BIT D/A (.25-4.75 volts out)
65313	FF21			BIT 0 = control of CD 0 = FIRQ to CPU disabled 1 = FIRQ to CPU enabled BIT 1 = RS232 Status input 0 = Set Flag on falling edge CD 1 = Set Flag on rising edge CD BIT 2 = Normally 1 Data I/O Register is addressed 0 = Changes FF20 to Data Direction Register BIT 3 = Cassette motor control 0 = Off 1 = On BIT 4 = 1 Always BIT 5 = 1 Always BIT 6 = Not used BIT 7 = CD Interrupt Flag
65314	FF22			BIT 0 = RS-232 Data input BIT 1 = Single Bit sound output BIT 2 = Ram size input 0 = 4K 1 = 16K BIT 3 = VDG Control output CSS Color Set Control BIT 4 = VDG Control output GM0 & NOT(INT)/EXT BIT 5 = VDG Control output GM1 BIT 6 = VDG Control output GM2 BIT 7 = VDG Control output NOT(A)/G

Beginning DEC ADDR	Ending HEX ADDR	Description
65280	FF00 65535 FFFF	INPUT/OUTPUT AREA[3] Cont.
65312	FF20 65315 FF23	PIA U4[3] Cont.
65315	FF23	Poke 54 to disable auto execute from cartridge Poke 55 to enable auto execute from cartridge BIT 0 = Control of the cartridge 0 = FIRQ to CPU disabled 1 = FIRQ to CPU enabled BIT 1 = Interrupt input 0 = Sets Flag on falling edge of cart 1 = Sets Flag on rising edge of cart BIT 2 = Normally 1 0 = Changes F22 to the Data Direction Reg. BIT 3 = Six Bit sound enable BIT 4 = 1 Always BIT 5 = 1 Always BIT 6 = Not used BIT 7 = Cartridge interrupt flag
65344	FF40	Disk Output latch (8 bit output port) Ref. [5] #9 BIT 0 = Drive select 0 BIT 1 = Drive select 1 BIT 2 = Drive select 2 BIT 3 = Motor on BIT 4 = Precomp BIT 5 = Double density BIT 6 = Drive select 3 BIT 7 = Halt enable
65352	FF48 65355 FF4B	Disk Control Registers for 1793 Disk controller chip Ref. [5] #9
65352	FF48	Disk status
65353	FF49	Disk track number
65354	FF4A	Disk sector number
65355	FF4B	Disk data

Beginning DEC ADDR	Ending HEX ADDR	Description
--------------------------	-----------------------	-------------

65280 FF00 65535 FFFF INPUT/OUTPUT AREA[3] Cont.

65376 FF60 65378 FF62 Control registers for special I/O (used for Radio Shack Digitizer 26-1195)

65376 FF60 X coordinate 0-255

65377 FF61 Y coordinate 0-191

65378 FF62 Pen Status

Bit 0 = Pen down

Bit 1 = Proximity (pen is within writing range of digitizer)

Bit 2 = X margin

Bit 3 = Y margin

65472 FFC0 65503 FFDF VDG Control Registers for 6883 SAM Address Multiplexer

65472 FFC0 65477 FFC5 VDG Mode Register contains 3 pairs of addresses (V0-V2). Poking any value to even addresses Set bit OFF (0) in VDG circuitry. Poking value to odd addresses set bit ON (1) in 6847 VDG circuitry[3]

65472 FFC0 65473 FFC1 Control of bit 0 (V0)

65474 FFC2 65475 FFC3 Control of bit 1 (V1)

65476 FFC4 65477 FFC5 Control of bit 2 (V2)

V0 V1 V2

1 1 1 DMA

1 0 1 G6R Graphics 6R (PMODE 4)

G6C Graphics 6C (PMODE 3)

1 1 0 G3R Graphics 3R (PMODE 2)

1 0 0 G3C Graphics 3C (PMODE 1)

0 1 1 G2R Graphics 2R (PMODE 0)

0 0 1 G2C Graphics 2C

0 1 0 G1C Graphics 1C

0 0 0 AL Alphanumeric Internal

AE Alphanumeric External

S4 Semigraphic 4 (Characters generated by SET instruction)

S6 Semigraphic 6

Beginning DEC ADDR	Ending HEX ADDR	Description
65280	FF00 65535 FFFF	INPUT/OUTPUT AREA[3]
65472	FFC0 65503 FFDF	VDG Control Registers for 6883 SAM Address Multiplexer Cont.
65478	FFC6 65491 FFD3	Page Select Register contains 7 pairs of addresses. (F0-F6) Controls display starting address (binary). Address of upper left most display element =0000+1/2*OFFSET. Poking any Value to even addresses sets bit OFF (0) in Page Select Circuitry. Poking any value to ODD addresses sets bit ON (1) in Page Select Circuitry. Also, Basepage is set by converting binary value of F (BITS F0-F6) to decimal and multiplying this decimal number by 512
65478	FFC6 65479 FFC7	Control of bit 0 F0
65480	FFC8 65481 FFC9	Control of bit 1 F1
65482	FFCA 65483 FFCE	Control of bit 2 F2
65484	FFCC 65485 FFCD	Control of bit 3 F3
65486	FFCE 65487 FFCF	Control of bit 4 F4
65488	FFD0 65489 FFD1	Control of bit 5 F5
65490	FFD2 65491 FFD3	Control of bit 6 F6
65492	FFD4 65493 FFD5	Page #1 P1 control of bit 7 F7 0 Normal 1
65494	FFD6 65497 FFD9	Clock speed (R0-R1). Poking any value to even addresses sets bit OFF (0). Poking any value to odd addresses sets bit ON (1)
65494	FFD6 65495 FFD7	Control of bit R0
65496	FFD8 65497 FFD9	Control of bit R1 R0 R1 1 1 0 1 1.8 All memory fast mode (RAM & ROM fast). Note getting all memory fast works as expected - 1 0 0.9/1.8 MHZ split memory (essentially ROM fast & RAM slow. 0 0 0.9 MHZ ONLY (all memory slow mode default)

Beginning		Ending		Description
DEC	HEX	DEC	HEX	
ADDR	ADDR	ADDR	ADDR	
65280	FF00	65535	FFFF	INPUT/OUTPUT AREA[3] Cont.
65472	FFC0	65503	FFDF	VDG Control Registers for 6883 SAM Address Multiplexer Cont.
65498	FFDA	65501	FFDD	Memory size (M0-M1) Poking any value to even addresses sets bit OFF (0). Poking any value to odd addresses sets bit ON (1)
65498	FFDA	65499	FFDB	Control of bit M0
65500	FFDC	65501	FFDD	Control of bit M1
				M0 M1
				1 1 64K static memory configuration
				0 1 64K dynamic memory configuration
				1 0 16K dynamic memory configuration
				0 0 4K dynamic memory configuration
65502	FFDE	65503	FFDF	Register (TY) Map type. Poking any value to even addresses sets bit OFF (0). Poking any value to odd addresses sets bit ON (1)
				TY
				0 Normal (ROM selected)
				1 ROM deselected
65522	FFF2	65523	FFF3	SWI3 Vector mapped to 49138 BFF2 - 49139 BFF3
65524	FFF4	65525	FFF5	SWI2 Vector mapped to 49140 BFF4 - 49141 BFF5
65526	FFF6	65527	FFF7	FIRQ Vector mapped to 49142 BFF6 - 49143 BFF7
65528	FFF8	65529	FFF9	IRQ Vector mapped to 49144 BFF8 - 49145 BFF9
65530	FFFA	65531	FFFB	SWI1 Vector mapped to 49146 BFFA - 49147 BFFB
65532	FFFC	65533	FFFD	NMI Vector mapped to 49148 BFFC - 49149 BFFD
65534	FFFE	65535	FFFF	RESET Vector mapped to 49150 BFFE - 49151 BFFF

Beginning		Ending		Description
DEC	HEX	DEC	HEX	
ADDR	ADDR	ADDR	ADDR	
				Map of areas that have changed between Basic 1.0 and 1.1
40988	A01C			Address offset changed
40996	A024			Address offset changed
40999	A027	41154	A0C2	Revised reset routine 1.0 ROM checked bit 2 of PIA to determine memory size (4/16k); 1.1 ROM checks bit 2 of PIA and also starts at \$600 and looks for ROM if no ROM is present 1.1 basic allows you to access 40K of Contiguous RAM
41217	A101	41220	A104	Register change
41301	A155			Logo change
41409	A1C1	41581	A26D	Revised keyboard routine to fix old bug where joystick fire buttons used to be input to the keyboard input routine
41669	A2C5	41722	A2FA	Fix to printer driver for 8 bit ascii versus 7 bit
42047	A43F	42048	A440	Fix to close file to output partially filled buffer without junk
42730	A6EA	42731	A6EB	FIX to OPEN

Beginning		Ending	
DEC	HEX	DEC	HEX
ADDR	ADDR	ADDR	ADDR

Description

				Map of areas that have changed between Basic 1.2 and 1.1
40961	A001			New start address for keyboard poll routine - POLCAT
41236	A114			Change to Basic initialization value of default Baud rate.
41301	A155			Change to copyright notice to replace 0 in 1.0 by 1 to obtain 1.1
41310	A15E			Change to year of copyright notice from 0 in 1980 to 3 in 1983.
41398	A1B6			New address for keyboard strobe routine.
41409	A1C1	41490	A212	Changes to POLCAT
41494	A216			"
41501	A21D			"
41505	A221			"
41517	A22D			"
41524	A234	41581	A26D	"
41667	A2C3	41722	A2FA	Changes to RS-232 Output Byte routine.
42346	A56A			Change to INKEY\$ routine.
44541	ADFD			"
45631	B23F			?
45966	B38E			?
46061	B3ED	46118	B426	Changes to INTCNV routine
47574	B9D6			Mod to floating point add.

Beginning DEC	Ending HEX	Beginning DEC	Ending HEX	Description
Map of areas that have changed between Extended Basic 1.1 and 1.0				
32976	80D0	32989	80DD	
33023	80FF			
33042	8112			
34555	86FB			
35867	8C1B	35874	8C22	Apparent fix to DLOAD command
35921	8C51			"
37241	9179			
37245	917D			
38444	962C	38445	962D	Fix to PMODE command.
38563	96A3	38579	96B3	Fixes to PCLEAR command to eliminate ?SN error when executing PCLEAR in start of a program.
\$				