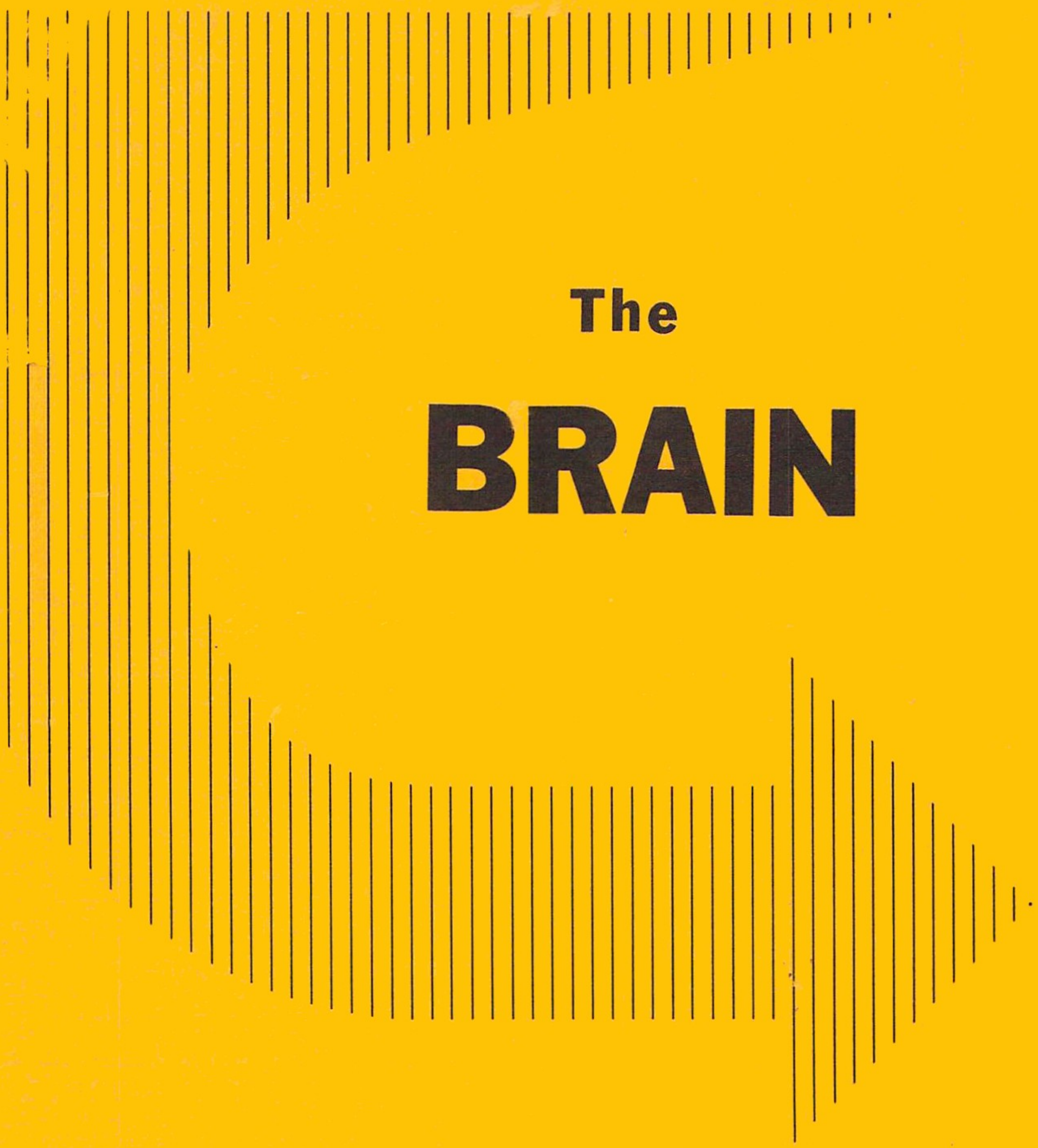
Unlock the Secrets of the COMMODORE 64



The BRAIN

A Programmer's Assistant

By
MAPS T.M.

INTRODUCING

THE BRAIN

Robert & Dell

Taylor

Distributed by VALUE SOFT
9513 S.W. Barbur Boulevard
Portland, Oregon 97219

<--- Super Help --->

Welcome to the concept of programming aides for the computer user. These tools have been written by an avid computerist, with the idea of placing help charts within finger tip reach. Stop hunting every time you need to know a printer command, just grab KISS. For those of you that are not familiar with the term kiss, it is a slang term for 'Keep it simple stupid'. Nothing intended, but that is what this package is all about, SIMPLE. Simple to use, that is.

THE BRAIN

The main body of the brain booklet is a memory chart. At first glance you might think that it is just another copy of a public domain memory map. Several months were spent to cross more than twelve existing maps, plus printing memory dumps of additional sections of the C-64 memory. It became a game to identify as many locations as possible. The end result is this map. It is a valuable tool to the BASIC or the machine language programmer. Another interesting use, is when your desire to examine someone elses code. Just where does that SYS command go, what does it do. The map is a perfect companion to the book 'Protection Revealed'.

A memory map is truly a map into the brain of the computer. Like a street map, the memory map identifies locations within the memory of the computer. Many of the locations may be used with PEEKs and POKEs.

EXAMPLE: When writing a game that uses a joystick, you might wish to disable the keyboard. To do so POKE 649,0. The default value is 649,10. On page 18 in KISS, you find a list of some of the other interesting POKEs. Experiment, try other POKES and PEEKS. You can't hurt anything. The explanations of memory locations were purposely kept condensed for quick reference. Every time I need to compare two addresses, they end up one on one side of a page, the other on the back. For this reason we only printed on one side. Don't feel bad if you can not understand all you read, some of the addresses are for the advanced programmer. Keep experimenting, you will soon be one yourself.

THE WALL MAP

The wall map is an outgrowth of desperation in trying to locate a given address on the screen. Just try to do that with the usual map that gives only the beginning and ending addresses. My first attempt was done with a felt pen on a large piece of cardboard. Everyone that saw it cried 'make me one too'. Mine had taken three days to make, obviously the idea did not appeal. Finally the bells began to ring, if my friends wanted one, why not print them.

The upper numbers in each square are the screen memory. The lower numbers are the color memory. By POKEing to a specific address, you have command of every square inch of the screen. At the bottom of the chart are the poke values of sixteen colors contained in the C-64.

KISS

KISS is a set of reference cards (cheat cards). They have been spiral bound so that they can be stood by your computer in easy view. Stop hunting through a dozen books to find the command you want. The index inside the front cover, lists all the charts contained within.

We sincerely hope you will find SUPER HELP as handy as we do. This packet can become a powerful tool to programmers at any level.

PACKET CONTENTS

1..Poster size screen and color memory map.

2..The Brain, a comprehensive manual of important charts and memory locations.

3..KISS, spiral bound set of reference cards.

THE BRAIN INDEX

This is truly the BRAIN of the C-64, a trip to the special places inside the operating system of the computer. This trip will hopefully add new features to the programs you write, making them fun and exciting.

GOOD LUCK.................

*C-64 is a registered trademark of Commodore Business Machines.

SHORT-HAND FOR COMMANDS

KEYWORD	SHORT-HAND	SCREEN	KEYWORD	SHORT-HAND	SCREEN
ABS	A-SHIFT-B	A │	RIGHT$	R-SHIFT-I	R ╮
AND	A-SHIFT-N	A ╱	RND	R-SHIFT-N	R ╱
ASC	A-SHIFT-S	A ♥	RUN	R-SHIFT-U	R ╭
ATN	A-SHIFT-T	A ▏	SAVE	S-SHIFT-A	S ♠
CHR$	C-SHIFT-H	C ▕	SGN	S-SHIFT-G	S ▏
CLR	C-SHIFT-L	C └	SIN	S-SHIFT-I	S ╮
CMD	C-SHIFT-M	C ╲	SPC(	S-SHIFT-P	S ┐
CONT	C-SHIFT-O	C ┌	SQR	S-SHIFT-Q	S ●
DATA	D-SHIFT-A	A ♠	STOP	S-SHIFT-T	S ▏
DEF	D-SHIFT-E	D ▔	SYS	S-SHIFT-Y	S ▕
DIM	D-SHIFT-I	D ╮	TAB	T-SHIFT-A	T ♠
END	E-SHIFT-N	E ╱	THEN	T-SHIFT-H	T ▕
EXP	E-SHIFT-X	E ♣	USR	U-SHIFT-S	U ♥
FOR	F-SHIFT-O	F ┌	VAL	V-SHIFT-A	V ♠
FRE	F-SHIFT-R	F ▁	VERIFY	V-SHIFT-E	V ▔
GET	G-SHIFT-E	G ▔	WAIT	W-SHIFT-A	W ♠
GOTO	G-SHIFT-O	G ┌			
INPUT#	I-SHIFT-N	I ╱			
LET	L-SHIFT-T	L ▏	CLOSE	CL-SHIFT-O	CL ┌
LIST	L-SHIFT-I	L ╮	GOSUB	GO-SHIFT-S	GO ♥
LOAD	L-SHIFT-O	L ┌	LEFT$	LE-SHIFT-F	LE ▁
MID$	M-SHIFT-I	M ╮	RESTORE	RE-SHIFT-S	RE ♥
NEXT	N-SHIFT-E	N ▔	RETURN	RE-SHIFT-T	RE ▏
NOT	N-SHIFT-O	N ┌	STEP	ST-SHIFT-E	ST ▔
OPEN	O-SHIFT-P	O ┐	STR$	ST-SHIFT-R	ST ▁
PEEK	P-SHIFT-E	P ▔			
POKE	P-SHIFT-O	P ┌			
PRINT#	P-SHIFT-R	P ▁	PRINT	?	?
READ	R-SHIFT-E	R ▔			

HEX TO DECIMAL CONVERSION

HEX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

SCREEN CODES

POKE	CHARACTER	POKE	CHARACTER	POKE	CHARACTER	POKE	CHARACTER
0	@	33	!	66	│	99	▔
1	A	34	"	67	─	100	▁
2	B	35	#	68	⎻	101	▏
3	C	36	$	69	⎺	102	▒
4	D	37	%	70	⎼	103	▕
5	E	38	&	71	▏	104	▄
6	F	39	'	72	▕	105	◤
7	G	40	(	73	╮	106	▐
8	H	41	)	74	╰	107	├
9	I	42	*	75	╯	108	▗
10	J	43	+	76	⌞	109	└
11	K	44	,	77	╲	110	┐
12	L	45	-	78	╱	111	▁
13	M	46	.	79	⌜	112	┌
14	N	47	/	80	⌝	113	┴
15	O	48	0	81	●	114	┬
16	P	49	1	82	⎽	115	┤
17	Q	50	2	83	♥	116	▎
18	R	51	3	84	▏	117	▍
19	S	52	4	85	╭	118	▐
20	T	53	5	86	╳	119	▔
21	U	54	6	87	○	120	▀
22	V	55	7	88	♣	121	▃
23	W	56	8	89	▕	122	┘
24	X	57	9	90	♦	123	▖
25	Y	58	:	91	┼	124	▝
26	Z	59	;	92	▒	125	┘
27	[	60	<	93	│	126	▘
28	£	61	=	94	π	127	▚
29	]	62	>	95	◥		
30	↑	63	?	96	SPACE		
31	←	64	─	97	▌		
32	SPACE	65	♠	98	▄		

ASCII CHARACTER SET

HEX	MSB	0	1	2	3	4	5	6	7
LSB	BIN	000	001	010	011	100	101	110	111
0	0000	NUL	DLE	SPACE	0	@	P	-	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	HT	EM	)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[	K	
C	1100	FF	FS	,	<	L		l	--
D	1101	CR	GS	-	=	M	]	m	
E	1110	SO	RS	.	>	N		n	
F	1111	SI	US	/	?	O	-	o	DEL

HEX	DECIMAL	DESCRIPTION
0000	0	CHIP DIRECTIONAL REGISTER
0001	1	CHIP INPUT/OUTPUT; MEMORY AND TAPE CONTROL
0002	2	UNUSED
0003-0004	3-4	FLOATER-FIXED VECTOR
0005-0006	5-6	FIXED-FLOAT VECTOR
0007	7	SEARCH CHAR. FOR CHAR. ":" OR END OF LINE
0008	8	SCAN FOR QUOTES FLAG AT END OF STRING
0009	9	TAB COLUMN SAVE
000A	10	VERIFY FLAG: 0=LOAD, 1=VERIFY
000B	11	INPUT BUFFER POINTER/ # SUBSCRIPT
000C	12	DEFAULT DIM FLAG
000D	13	TYPE: FF=STRING, 00=NUMRIC
000E	14	TYPE: 80=INTEGER, 00=FLOATING POINT
000F	15	DATA SCAN/LIST QUOTE/MEMORY FLAG
0010	16	SUBSCRIPT/FNX FLAG
0011	17	0=INPUT; $40=GET; $98=READ
0012	18	TAN SIGN/COMPARISON EVAL FLAG
0013	19	CURRENT I/O PROMPT FLAG
0014-0015	20-21	BASIC INTERGER ADDRESS (FOR SYS - GOTO ETC)
0016	22	POINTER: TEMPORARY STRING STACK
0017-0018	23-24	LAST TEMPORARY STRING VECTOR
0019-0021	25-33	STACK FOR TEMPORARY STRINGS
0022	34-35	POINTER FOR NUMBER TRANSFER
0024	36-37	MISC. NUMBER POINTER
0026-002A	38-42	FLOATING POINT AREA OF MULT. & DIVIDE
0028-002C	43-44	POINTER: START OF BASIC
002B-0020	45-46	POINTER: START OF BASIC VARIABLES
002F-0030	47-48	POINTER: START OF BASIC ARRAYS
0031-0032	49-50	POINTER: END OF BASIC ARRAYS
0033-0034	51-52	POINTER: BOTTOM OF STRING STORAGE
0035-0036	53-54	UTILITY STRING POINTER
0037-0038	55-56	POINTER: LIMIT OF BASIC MEMORY
0039-003A	57-58	CURRENT BASIC LINE NUMBER
003B-003C	59-60	PREVIOUS BASIC LINE NUMBER
003D-003E	61-62	POINTER: BASIC STATEMENT FOR CONT
003F-0040	63-64	CURRENT DATA LINE NUMBER
0041-0042	65-66	CURRENT DATA ADDRESS
0043-0044	67-68	TEMP. STORAGE DURING INPUT & READ ROUTINES
0045-0046	69-70	CURRENT VARIABLE NAME
0047-0048	71-72	CURRENT VARIABLE ADDRESS
0049-004A	73-74	VARIABLE POINTER FOR FOR/NEXT
004B-004C	75-76	Y-SAVE; OP-SAVE; BASIC POINTER SAVE
004D	77	COMPARISON SYMBOL ACCUMULATOR
004E	78-79	MISC. WORK AREA, POINTERS ETC.
0050	80-81	WORK AREA; POINTER TO STRING DESCRIPTION
0052	82	LENGTH OF ABOVE STRING
0053	83	CONSTANT USED BY GARBAGE COLLECTION - 3 OR 7
0054-0055	84-86	JUMP VECTOR FOR FUNCTIONS
0057-0060	87-96	MISC WORK AREA
005C-0060	92-96	TEMPORARY STORAGE
0061	97	ACCUM #1; EXPONENT
0062-0065	98-101	ACCUM #1: MANTISSA
0066	102	ACCUM #1: SIGN
0067	103	SERIES EVALUATION CONSTANT POINTER
0068	104	ACCUM #1: OVERFLOW

HEX	DECIMAL	DESCRIPTION
0069	105	ACCUM #2: EXPONANT
006A-006D	106-109	ACCUM #2: MANTISSA
006E	110	FLOAT ACCUM #2
006F	111	SIGN OF RESULT OF MATH EVALUATION
0070	112	ACCUM #1 LO-ORDER (ROUNDING)
0071-0072	113-114	CASSETTE BUFFER POINTER
0073-008A	115-138	CHRGET SUBROUTINE; GET BASIC CHARACTER
007A-007B	122-123	BASIC POINTER (WITHIN SUBRTN)
008B-008F	139-143	RND SEED VALUE
0090	144	STATUS WORD ST
0090-00FF	144-255	KERNAL WORK STORAGE AREA
0091	145	STOP AND RVS FLAGS
0092	146	TIMING FOR TAPE
0093	147	LOAD=0, VERIFY=1
0094	148	SERIAL OUTPUT DEFERRED CHARACTER FLAG
0095	149	SERIAL DEFERRED CHARACTER
0096	150	END OF TAPE RECIEVED
0097	151	REGISTER SAVE
0098	152	NUMBER OF OPEN FILES
0099	153	INPUT DEVICE, NORMALLY 0
009A	154	OUTPUT CMD DEVICE, NORMALLY 3
009B	155	TAPE CHARACTER OUTPUT PARITY
009C	156	BYTE-RECIEVED FLAG
009D	157	DIRECT=$80/RUN; 0 OUTPUT CONTROL
009E	158	TAPE PASS 1 ERROR LOG/CHARACTER BUFFER
009F	159	TAPE PASS 2 ERROR LOG CORRECTED
00A0-00A2	160-162	JIFFY CLOCK HOURS,MINUTES,SECONDS
00A3	163	SERIAL BIT COUNT/EOI FLAG
004A	164	CYCLE COUNT
00A5	165	COUNTDOWN, TAPE WRITE/BIT COUNT
00A6	166	TAPE BUFFER POINTER
00A7	167	TAPE WRITE 1 DRIVE COUNT/RD PASS/INBIT
00A8	168	TAPE WRITE NEW BYTE/READ ERROR/INBIT COUNT
00A9	169	START BIT ERROR CHECK
00AA	170	RS-232 INPUT BYTE BUFFER
00AB	171	RS-232 INPUT PARITY
00AC-00AD	172-173	POINTER: TAPE BUFFER, SCREEN SCROLLING
00AE-00AF	174-175	TAPE END ADDS/END OF PROGRAM
00B0-00B1	176-177	TAPE TIMING CONSTANTS
00B2-00B3	178-179	POINTER: START OF TAPE BUFFER
00B4	180	1=TAPE TIMER ENABLED; BIT COUNT
00B5	181	RS-232 NEXT BIT TO SEND/READ OR END
00B6	182	READ CHARACTER ERROR/OUTBYTE BUFFER
00B7	183	NUMBER OF CHARACTERS IN FILE NAME
00B8	184	CURRENT LOGICAL FILE
00B9	185	CURRENT SECONDARY ADDRESS
00BA	186	CURRENT FILE - FIRST ADDRESS
00BB-00BC	187-188	POINTER TO FILE NAME
00BD	189	WRITE SHIFT WORD READ INPUT CHARACTER
00BE	190	NUMBER OF BLOCKS TO READ AND WRITE
00BF	191	SERIAL WORD BUFFER
00C0	192	TAPE MOTOR INTERLOCK
00C1-00C2	193-194	START ADDRESS FOR LOAD & CASSETTE WRITE
00C3-00C4	195-196	KERNAL SETUP POINTER/CASSETTE TEMP
00C5	197	LAST KEY DEPRESSED

HEX	DECIMAL	DESCRIPTION
00C6	198	NUMBER OF CHAR. IN KEYBOARD BUFFER
00C7	199	SCREEN REVERSE FLAG
00C8	200	END OF LINE FOR INPUT POINTER
00C9-00CA	201-202	INPUT CURSOR LOG (ROW, COLUMN)
00CB	203	WHICH KEY: 64 IF NO KEY
00CC	204	0=FLASH CURSOR
00CD	205	CURSOR TIMING COUNTDOWN
00CE	206	CHARACTER UNDER CURSOR
00CF	207	CURSOR STATUS - $00=ENABLED, $01=DISABLED
00D0	208	INPUT FROM SCREEN OR KEYBOARD
00D1-00D2	209-210	POINTER TO SCREEN LINE ADDRESS
00D3	211	POSITION OF CURSOR ON CURRENT LINE
00D4	212	0=DIRECT CURSOR, ELSE PROGRAMMED
00D5	213	CURRENT SCREEN LINE LENGTH
00D6	214	ROW WHERE CURSOR LIVES
00D7	215	LAST INKEY/CHECKSUM/BUFFER
00D8	216	NUMBER OF INSERTS OUTSTANDING
00D9-00F2	217-242	SCREEN LINE LINK TABLE
00F2	242	SCREEN ROW MARKER
00F3-00F4	243-244	SCREEN COLOR POINTER
00F5-00F6	245-246	KEYBOARD POINTER
00F7-00F8	247-248	RS-232 INPUT BUFFER POINTER
00F9-00FA	249-250	RS-232 OUTPUT BUFFER POINTER
00FB-00FE	251-254	FREE ZERO PAGE SPACE FOR USER PROGRAM
00FF	255	BASIC WORK AREA
0100-010A	256-266	FLOATING TO ASCII WORK AREA
0100-103E	256-318	TAPE ERROR LOG
0100-01FF	256-511	PROCESSOR STACK AREA
013F-01FF	319-511	BASIC STACK AREA
0200-0258	512-600	BASIC INPUT BUFFER
0259-0262	601-610	LOGICAL FILE TABLE
0263-0260	611-620	DEVICE TABLE
026D-0276	621-630	SECONDARY ACTIVE FILE ADDRESSES
0277-0280	631-640	KEYBOARD BUFFER
0281-0282	641-642	START OF BASIC MEMORY
0283-0284	643-644	TOP OF BASIC MEMORY
0285	645	SERIAL BUS TIMEOUT FLAG
0286	646	CURRENT COLOR CODE OF CURSOR
0287	647	COLOR UNDER CURSOR
0288	648	HIGH BYTE OF SCREEN MEMORY ADDRESS
0289	649	SIZE OF KEYBOARD BUFFER
028A	650	REPEAT ALL KEYS
028B	651	REPEAT SPEED COUNTER
028C	652	REPEAT DELAY COUNTER
028C	653	KEYBOARD SHIFT/CONTROL FLAG
028E	654	LAST SHIFT PATTERN
028F-0290	655-656	KEYBOARD TABLE SETUP POINTER
0291	657	KEYBOARD SHIFT MODE
0292	658	0=SCROLL ENABLE

** RS-232 PSEUDO 6551 REGISTERS **

0293	659	RS-232 CONTROL REGISTERS
0294	660	RS-232 COMMAND REGISTERS
0295-0296	661-662	RS-232 BIT TIMING

HEX	DECIMAL	DESCRIPTION
0297	663	RS-232 STATUS
0298	664	RS-232 NUMBER OF BITS LEFT TO SEND
0299-029A	665-666	RS-232 BAUD RATE
029B-029E	667-670	BEGINNING & END OF INPUT/OUTPUT BUFFER
029B	667	RS-232 RECIEVE POINTER
029C	668	RS-232 INPUT POINTER
029D	669	RS-232 OUTPUT POINTER START
029E	670	RS-232 OUTPOINT POINTER END
029F-02A0	671-672	IRQ SAVE DURING TAPE INPUT/OUTPUT
02A1	673	RS-232 ENABLES
02A2	674	CIA 1 TIMER A CONTROL LOG
02A3	675	CIA 1 INTERRUPT LOG
02A4	676	CIA 1 TIMER, AN ENABLE FLAG
02A5	677	SCREEN ROW MARKER
02A6	678	60 HERTZ ?
02A7-02FF	679-767	UNUSED
0300-030B	768-779	BASIC INDIRECT VECTOR TABLE
0300-0301	768-769	ERROR MESSAGE LINK
0302-0303	770-771	BASIC WARM START LINK
0304-0305	772-773	CRUNCH BASIC TOKENS LINK
0306-0307	774-775	PRINT TOKENS LINK LIST
0308-0309	776-777	START NEW BASIC CODE LINK
030A-030B	778-779	SET ARITHMETIC ELEMENT LINK
030C-030F	780-783	REGISTER STORAGE AREA
030C	780	SYS A-REGISTER SAVE
030D	781	SYS X-REGISTER SAVE
030E	782	SYS Y-REGISTER SAVE
030F	783	SYS STATUS REGISTER SAVE
0310	784	USR FUNCTION JUMP INSTRUCTIONS
0311-0312	785-786	HIGH & LOW BYTE OF USR START ADDRESS
0313	787	UNUSED
0314-0315	788-789	HARDWARE INTERRUPT VECTOR
0316-0317	790-791	BREAK INTERRUPT VECTOR
0318-0319	792-793	NNI INTERRUPT VECTOR
L31A-032D	794-813	KERNAL INDIRECT VECTORS
031A-031B	794-795	OPEN VECTOR
031C-031D	796-797	CLOSE VECTOR
031E-031F	798-799	SET-INPUT VECTOR
0320-0321	800-801	SET-OUTPUT VECTOR
0322-0323	802-803	RESTORE INPUT/OUTPUT VECTOR
0324-0325	804-805	INPUT VECTOR
0326-0327	806-807	OUTPUT VECTOR
0328-0329	808-809	TEST-STOP VECTOR
032A-032B	810-811	GET VECTOR
032C-032D	812-813	ABORT INPUT/OUTPUT VECTOR
032E-032F	814-815	WARM START VECTOR
0330-0331	816-817	LOAD LINK
0332-0333	818-819	SAVE LINK
0334-03FB	820-1019	CASSETTE BUFFER
0334-033B	820-827	UNUSED
033C-03FB	828-1019	TAPE INPUT/OUTPUT BUFFER
0340-037E	832-894	(SPRITE 13)
0380-03BE	896-958	(SPRITE 14)
03C0-03FE	960-1022	(SPRITE 15)
03FC-03FF	1020-1023	UNUSED

HEX	DECIMAL	DESCRIPTION
0400-07E7	1024-2023	SCREEN MEMORY
07F8-07FF	2040-2047	SPRITE POINTERS
0800-9FFF	2048-40959	BASIC RAM MEMORY
1000-1FFF	4096-8191	CHARACTER ROM IMAGE FOR VIC II CHIP/BANK 0
8000-9FFF	32768-40959	ALTERNATE RAM OR ROM EXROM CARTRIDGE
9000-9FFF	36864-40959	CHARACTER ROM IMAGE FOR VIC II CHIP/BANK 2
A000-A001	40960-49161	COLD START VECTOR
A000-BFFF	40960-59151	ALTERNATE: RAM
A000	40960	KEYWORD ACTION ADDRESS
A002-A003	40962-40963	WARM START VECTOR
A004-A00B	40964-40971	ASCII TEXT CHARACTERS CBM BASIC
A00C-A051	40972-41041	DISPATCH VECTOR TABLE
A046	41030	FUNCTION ACTION ADDRESS
A052-A07F	41042-41087	FUNCTION DISPATCH VECTOR TABLE
A074	41076	OPERATOR ACTION ADDRESS
A080-A09D	41088-41117	OPERATOR DISPATCH VECTOR TABLE
A09E-A19D	41118-41373	KEYWORD TABLE
A19E-A327	41374-41767	ASCII TEXT OF BASIC ERROR MESSAGES
A328-A364	41768-41828	EROR MESSAGE VECTOR TABLE
A365-A389	41829-41865	MISC. MESSAGES
A38A-A3B7	41866-41911	FIND FOR - GOSUB SEARCH STACK
A3B8-A3FA	41912-41978	OPEN MEMORY SPACE
A3FB-A407	41979-41991	TEST STACK DEPTH
A408-A434	41992-42036	CHECK AVAILABLE MEMORY
A435-A468	42037-42088	SEND ERROR MESSAGE
A437-A468	42039-42088	ERROR HANDLER
A469-A473	42089-42099	DISPLAY ERROR OR OTHER MESSAGE
A474-A47F	42100-42111	PRINT READY
A480-A49B	42112-42139	MAIN BASIC PROGRAM LOOP
A49C	42140	ADD OR REPLACE LINE OF TEXT
A533	42291	BASIC LINE CHAINING
A560	42336	RECIEVE LINE FROM THE KEYBOARD
A579	42361	TOKENIZE BASIC LINE
A613	42515	SEARCH FOR LINE NUMBER
A642	42562	PERFORM NEW
A660	42592	PERFORM CLR
A68E	42638	RESET BASIC EXECUTION TO START-OF-PROGRAM
A69C	42652	PERFORM LIST
A7A7	42775	PRINT BASIC TOKENS AS ASCII CHARACTERS
A742	42818	PERFORM FOR
A7AE	42926	SET UP NEXT STATEMENT
A7E4	42980	EXCUTE BASIC STATEMENT
A81D	43037	PERFORM RESTORE
A82C	43052	TEST STOP AND END
A82F	43055	PERFORM STOP
A831	43057	PERFORM END
A857	43095	PERFORM CONT
A871	43121	PERFORM RUN
A883	43139	PERFORM GOSUB
A8A0	43168	PERFORM GOTO
A8D2	43218	PERFORM RETURN
A8EB	43243	PERFORM DATA
A906	43270	SCAN FOR NEXT STATEMENT
A909	43273	SCAN FOR NEXT LINE
A928	43304	PERFORM IF

HEX	DECIMAL	DESCRIPTION
A93B	43323	PERFORM REM
A94B	43339	PERFORN ON
A96B	43371	GET INTERGER FROM TEXT
A9A5	43429	PERFORM LET
AA80	43648	PERFORM PRINT#
AA86	43654	PERFORM CMD
AA9A	43674	PERFORM PRINT
AB1E	43806	PRINT STRING FROM ANY MEMORY
AB3B	43835	PRINT FORMAT CHARACTER
AB4D	43853	PROCESS BAD INPUT
AB7B	43899	PERFORM GET
ABA5	43941	PERFORM INPUT#
ABBF	43967	PERFORM INPUT
ABF9	44025	PROMPT AND INPUT
AC06	44038	PERFORM READ
ACFC	44284	INPUT ERROR MESSAGES
AD1E	44318	PERFORM NEXT
AD78	44408	TYPE MATCH CHECK
AD8A	44426	EVALUATE NUMERIC EXPRESSION
AD9E	44446	EVALUATE EXPRESSION
AEA8	44712	CONSTANT - PI
AEF1	44785	EVALUTE WITHIN BRACKETS
AEF7	44791	CHECK FOR ")"
AEFA	44794	CHECK FOR "("
AEFD	44797	CHECK FOR "-"
AEFF	44799	CHECK FOR ","
AF08	44808	SYNTAX ERROR
AF14	44820	CHECK FOR VARIABLE NAME
AF2B	44843	GET VALUE OF A VARIABLE
AFA7	44967	SET UP FN REFERANCES
AFE6	45030	PERFORM OR
AFE9	45033	PERFORM AND
B016	45078	COMPARISON ROUTINE
B07E	45182	PERFORM DIM
B08B	45195	LOCATE VARIABLE
B11D	45341	CHECK FOR ALPHA ASII
B11F	45343	CREATE NEW VARIABLE
B185	45445	RETURN ADDRESS OF VARIABLE
B194	45460	ARRAY POINTER ROUTINE
B1A5	45477	32768 IN FLOATING POINT
B1AA	45482	CONVERT FLOATING POINT NUMBER
B1BF	45503	FAC 1 TO INTEGER
B1D1	45521	FIND OR CREATE ARRAY
B245	45637	PRINT BAD SUBSCRIPT ERROR MESSAGE
B248	45640	PRINT ILLEGAL QUANITY ERROR MESSAGE
B34C	45900	COMPUTE ARRAY SIZE
B37D	45949	PERFORM FRE
B391	45969	INTEGER TO FAC 1
B39E	45982	PERFORM POS
B3A6	45990	CHECK FOR DIRECT MODE
B3B3	46003	PERFORM DEF
B3E1	46049	CHECK FOR FN SYNTAX
B3F4	46068	EVALUATE FN
B465	46181	PERFORM STR$
B475	46197	CALCULATE STRING VECTOR

HEX	DECIMAL	DESCRIPTION
B487	46215	SET UP STRING
B4F4	46324	MAKE ROOM FOR STRING
B526	46374	COLLECT GARBAGE (MAKE ROOM FOR STRING)
B5BD	46525	CHECK STRING COLLECTION ELIGIBILITY
B606	46598	COLLECT STRING
B63D	46653	CONCATENATE STRING
B67A	46714	BUILD STRING TO MEMORY
B6A3	46755	DISCARD UNWANTED STRING
B6BD	46811	CLEAN THE DESCRIPTOR STACK
B6EC	46828	PERFORM CHR$
B700	46848	PERFORM LEFT$
B72C	46892	PERFORM RIGHT$
B737	46903	PERFORM MID$
B761	46945	PULL STRING PARAMETERS FROM STACK
B77C	46972	PERFORM LEN
B782	46978	EXIT STRING MODE
B78B	46987	PERFORM ASC
B79B	47003	INPUT BYTE PARAMETERS
B7AD	47021	PERFORM VAL
B7EB	47083	GET POKE/WAIT PARAMETERS
B7F7	47095	FAC 1 TO INTERGER
B80D	47117	PERFORM PEEK
B824	47140	PERFORM POKE
B82D	47149	PERFORM WAIT
B849	47177	ADD 0.5 TO FAC 1
B850	47184	SUBTRACT FAC 1 FROM NUMBER IN MEMORY
B853	47187	PERFORM SUBTRACTION
B867	47207	ADD FAC 1 TO NUMBER IN MEMORY
B86A	47210	PERFORM ADDITION
B8A7	47271	MAKE RESULT NEGATIVE IF BORROW
B8FE	47358	NORMALIZE FLOATING POINT ACC. #1
B947	47431	COMPLEMENT FAC 1
B97E	47486	OVERFLOW
B983	47491	SINGLE BYTE MULTIPLY
B9BC	47548	FLOATING POINT CONSTANTS
B9C1	47553	FLOATING POINT TABLE FOR LOG FUNCTION
B9EA	47594	PERFORM LOG
BA28	47656	MULTIPLY FAC 1 * MEMORY
BA30	47664	MULTIPLY FAC 2 * FAC 1
BA59	47705	MULTIPLY A BIT
BA8C	47756	MEMORY TO FAC 2
BAB7	47799	ADJUST FAC 1/FAC 2
BAD4	47828	UNDERFLOW/OVERFLOW
BAE2	47842	MULTIPLY FAC 1 BY 10
BAF9	47865	CONSTANT 10
BAFE	47870	DIVIDE BY 10
BB07	47879	DIVIDE FAC 2 / MEMORY
BB0F	47887	DIVIDE MEMORY / FAC 1
BB12	47890	DIVIDE FAC 2 / FAC 1
BBA2	48034	MEMORY TO FAC 1
BBD7	48087	FAC 1 TO MEMORY
BBFC	48124	FAC 2 TO FAC 1
BC0C	48140	ROUND NUMBER AND MOVE FROM FAC 1 TO FAC 2
BC0F	48143	FAC 1 TO FAC 2
BC1B	48155	ROUND OFF FAC 1
BC2B	48171	GET SIGN

HEX	DECIMAL	DESCRIPTION
BC39	48185	PERFORM SGN
BC58	48216	PERFORM ABS
BC5B	48219	COMPARE FAC 1 TO MEMORY
BC9B	48283	FAC 1 TO INTEGER
BCCC	48332	PERFORM INT
BCF3	48371	ASCII TO FAC 1
BD7E	48510	GET NEW ASCII DIGIT
BDB3	48563	CONSTANTS
BDC0	48576	PRINT IN FOLLOWED BY LINE NUMBER
BDCD	48589	OUTPUT NUMBER IN ASCII DEC. DIGITS
BDDD	48605	FAC 1 TO ASCII
BF11	48913	MORE CONSTANTS
BF1C	48924	POWERS OF MINUS TEN CONSTANTS TABLE
BF3A	48954	TABLE OF CONSTANTS FOR TI$ CONVERSION
BF52	48978	UNUSED AREA
BF71	49009	PERFORM SQR
BF78	49016	PERFORM EXPONENTIATION
BFB4	49076	PERFORM NEGATION
BFBF	49087	MORE CONSTANTS
BFED	49133	PERFORM EXP
C000	49152-53247	RAM AVAILABLE FOR MACHINE LANGUAGE PROGRAMS

** MOS 6566 VIDEO INTERFACE CONTROLLER (VIC) **

D000	53248	SPRITE 0 X POSITION
D001	53249	SPRITE 0 Y POSITION
D002	53250	SPRITE 1 X POSITION
D003	53251	SPRITE 1 Y POSITION
D004	53252	SPRITE 2 X POSITION
D005	53253	SPRITE 2 Y POSITION
D006	53254	SPRITE 3 X POSITION
D007	53255	SPRITE 3 Y POSITION
D008	53256	SPRITE 4 X POSITION
D009	53257	SPRITE 4 Y POSITION
D00A	53258	SPRITE 5 X POSITION
D00B	53259	SPRITE 5 Y POSITION
D00C	53260	SPRITE 6 X POSITION
D00D	53261	SPRITE 6 Y POSITION
D00E	53262	SPRITE 7 X POSITION
D00F	53263	SPRITE 7 Y POSITION
D010	53264	SPRITES 0-7 M.S. BIT OF X-COORD.
D011	53265	VIC CONTROL REGISTER
D012	53266	READ RASTER/WRITE FOR COMP. IRQ
D013	53267	LIGHT-PEN LATCH X-POSITION
D014	53268	LIGHT-PEN LATCH Y-POSITION
D015	53269	SPRITE DISPLAY ENABLE
D016	53270	VIC CONTROL REGISTER
D017	53271	SPRITES 0-7 EXPAND VERTICAL
D018	53272	VIC MEMORY CONTROL REGISTER
D019	53273	VIC INTERRUPT FLAG REGISTER
D01A	53274	IRQ MASK REGISTER
D01B	53275	SPRITE TO BACKGROUND DISPLAY PARITY
D01C	53276	SPRITES 0-7 MULTI-COLOR MODE SEL
D01D	53277	SPRITES 0-7 EXPAND 2 HORIZONTAL
D01E	53278	SPRITE TO SPRITE COLLISION DETECTION

HEX	DECIMAL	DESCRIPTION
D01F	53279	SPRITE TO BACKGROUND COLLISION DETECTION
D020	53280	BORDER COLOR
D021	53281	BACKGROUND COLOR 0
D022	53282	BACKGROUND COLOR 1
D023	53283	BACKGROUND COLOR 2
D024	53284	BACKGROUND COLOR 3
D025	53285	SPRITE MULTICOLOR REGISTER 0
D026	53286	SPRITE MULTICOLOR REGISTER 1
D027	53287	SPRITE 0 COLOR
D028	53288	SPRITE 1 COLOR
D029	53289	SPRITE 2 COLOR
D02A	53290	SPRITE 3 COLOR
D02B	53291	SPRITE 4 COLOR
D02C	53292	SPRITE 5 COLOR
D02D	53293	SPRITE 6 COLOR
D02E	53294	SPRITE 7 COLOR
D02F-D03F	53295-53311	NOT CONNECTED
D040-D3FF	53312-54271	VIC-II REGISTER IMAGES

**** MOS 6581 SOUND INTERFACE DEVICE (SID) ****

D400	54272	VOICE 1: FREQUENCY CONTROL-LOW
D401	54273	VOICE 1: FREQUENCY CONTROL-HIGH
D402	54274	VOICE 1: PULSE WAVEFORM WIDTH-LOW
D403	54275	VOICE 1: PULSE WAVEFORM WIDTH-HIGH
D404	54276	VOICE 1: CONTROL REGISTER
D405	54277	ENVELOPE GENERATOR 1: ATTACK/DECAY
D406	54278	ENVELOPE GENERATOR 1: SUSTAIN/REL
D407	54279	VOICE 2: FREQUENCY - LOW BYTE
D408	54280	VOICE 2: FREQUENCY - HIGH BYTE
D409	54281	VOICE 2: PULSE WAVEFORM - LOW BYTE
D40A	54282	VOICE 2: PULSE WAVEFORM - HIGH
D40B	54283	VOICE 2: CONTROL REGISTER
D40C	54284	ENVELOPE GENERATOR 2: ATTACK/DECAY
D40D	54285	ENVELOPE GENERATOR 2: SUSTAIN/REL
D40E	54286	VOICE 3: FREQUENCY - LOW BYTE
D40F	54287	VOICE 3: FREQUENCY - HIGH BYTE
D410	54288	VOICE 3: PULSE WAVEFORM - LOW BYTE
D411	54289	VOICE 3: PULSE WAVEFORM - HIGH NYBBLE
D412	54290	VOICE 3: CONTROL REGISTER
D413	54291	ENVELOPE GENERATOR 3: ATTACK/DECAY
D414	54292	ENVELOPE GENERATOR 3: SUSTAIN/REL
D415	54293	FILTER CUTOFF FREQUENCY: LOW NIBBLE
D416	54294	FILTER CUTOFF FREQUENCY: HIGH BYTE
D417	54295	FILTER RESONANCE/VOICE INPUT CONTROL
D418	54296	SELECT FILTER MODE AND VOLUME
D419	54297	ANALOG/DIGITAL CONVERTER: PADDLE 1
D41A	54298	ANALOG/DIGITAL CONVERTER: PADDLE 2
D41B	54299	OSCILLATOR 3: RANDOM NUMBER GENERATOR
D41C	54300	ENVELOPE GENERATOR 3 OUTPUT
D41D-D41F	54301-54303	NOT CONNECTED
D420-D7FF	54304-55295	SID IMAGES
D800-DBFF	55296-56319	COLOR RAM

** MOS 6526 COMPLEX INTERFACE ADAPTER (CIA) #1 **

HEX	DECIMAL	DESCRIPTION
DC00	56320	DATA PORT A JOYSTICK 2
DC01	56321	DATA PORT B JOYSTICK 1
DC02	56322	DATA DIRECTION REGISTER: PORT A
DC03	56323	DATA DIRECTION REGISTER: PORT B
DC04	56324	TIMER A: LOW BYTE
DC05	56325	TIMER A: HIGH BYTE
DC06	56326	TIMER B: LOW BYTE
DC07	56327	TIMER B: HIGH BYTE
DC08	56328	TIME-OF-DAY CLOCK: 1/10 SECOND
DC09	56329	TIME-OF-DAY CLOCK: SECONDS
DC0A	56330	TIME-OF-DAY CLOCK: MINUTES
DC0B	56331	TIME-OF-DAY CLOCK: HOURS AND AM/PM
DC0C	56332	SERIAL DATA PORT
DC0D	56333	INTERRUPT CONTROL REGISTER
DC0E	56334	CONTROL REGISTER A
DC0F	56335	CONTROL REGISTER B
DC10-DCFF	56336-56575	CIA #1 REGISTER IMAGES

** MOS 6526 COMPLEX INTERFACE ADAPTER (CIA) #2 **

DD00	56576	DATA PORT A (SERIAL BUS, RS-232)
DD01	56577	DATA PORT B (USER PORT, RS-232)
DD02	56578	DATA DIRECTION REGISTER: PORT A
DD03	56579	DATA DIRECTION REGISTER: PORT B
DD04	56580	TIMER A: LOW BYTE
DD05	56581	TIMER A: HIGH BYTE
DD06	56582	TIMER B: LOW BYTE
DD07	56583	TIMER B: HIGH BYTE
DD08	56584	TIME-OF-DAY CLOCK: 1/10 SECONDS
DD09	56585	TIME-OF-DAY CLOCK: SECONDS
DD0A	56586	TIME-OF-DAY CLOCK: MINUTES
DD0B	56587	TIME-OF-DAY CLOCK: HOURS AND AM/PM
DD0C	56588	SYNCHRONOUS SERIAL I/O BUFFER
DD0D	56589	CIA INTERRUPT CONT. REGISTER
DDOE	56590	CIA CONTROL REGISTER A
DDOF	56591	CIA CONTROL REGISTER B
DD10-DDFF	56592-56831	CIA #2 REGISTER IMAGES
DE00-DEFF	56832-57087	RESERVED FOR FUTURE I/O EXPAN.
DF00-DFFF	57088-57343	RESERVED FOR FUTURE I/O EXPAN.

** KERNAL ROM **

LOCATION 57344 TO 65535 (E000 - FFFF) ARE USED BY THE 8K OPERATING SYSTEM KERNAL ROM. KERNAL ROM IS SUBJECT TO REVISION (BY THE MANUFACTURER) AND THESE ROUTINES MAY NOT STAY IN THE SAME LOCATIONS. IT IS RECOMMENDED THE USER REFER TO THE JUMP TABLE AT THE END OF ROM. BY HAVING YOUR PROGRAM JUMP TO ONE OF THE ADDRESSES IN THE JUMP TABLE, IT SHOULD RUN WITHOUT MODIFICATION ON FUTURE MODELS

E043	57411	SERIES EVALUATION
E059	57433	FUNCTION SERIES EVALUATION SUBROUTINE 2
E08D	57485	MULTIPLICATIVE CONSTANT FOR RND

HEX	DECIMAL	DESCRIPTION
E08D	57485	MULTIPLICATIVE CONSTANT FOR RND
E092	57490	ADDITIVE CONSTANT FOR RND
E097	57495	PERFORM RND
E0F9	57593	CALL KERNAL INPUT/OUPUT ROUTINES
E12A	57642	PERFORM SYS
E156	57686	PERFORM SAVE
E165	57701	PERFORM VERIFY
E168	57704	PERFORM LOAD
E1BE	57790	PERFORM OPEN
E1C7	57799	PERFORM CLOSE
E1D4	57812	SET PARAMETERS FOR LOAD,VERIFY,SAVE
E200	57856	SKIP COMMA & GET INTEGER IN .X
E206	57862	GET CURRENT CHARACTER & CHECK FOR END OF LINE
E20E	57870	CHECK FOR COMMA
E219	57881	SET PARAMETERS FOR OPEN & CLOSE
E264	57956	PERFORM COS
E268	57960	PERFORM SIN
E2B4	58036	PERFORM TAN
E2E0	58080	CONSTANTS FOR TRIG FUNCTIONS
E2E5	58085	5 BYTE FLOATING POINT REP. OF CONSTANT 2*PI
E2EA	58090	5 BYTE FLOATING POINT REP. OF CONSTANT 1/4
E2EF	58095	TABLE OF CONSTANTS FOR EVAL. OF SIN,COS,TAN
E30E	58126	PERFORM ATN
E33E	58174	CONSTANTS FOR ATN
E37B	58235	WARM START BASIC
E38B	58251	ERROR MESSAGE HANDLER
E394	58260	COLD START BASIC
E3A2	58274	CHRGET FOR ZERO PAGE
E3BA	58298	INITIAL RND SEED VALUE
E3BF	58303	INITIALIZE BASIC
E422	58402	PRINT BASIC START-UP MESSAGES
E447	58439	TABLE OF VECTORS/IMPORTANT BASIC ROUTINES
E453	58451	COPY BASIC VECTORS TO RAM
E460	58464	MESSAGES
E4AD	58541	PROGRAM PATCH AREA
E4B7	58551	35 BYTES NOT IN USE
E4DA	58586	CLEAR COLOR RAM
E4E0	58592	PAUSE WHEN FILE FOUND ON CASS.
E505	58629	SET SCREEN LIMITS
E50A	58634	TRACK CURSOR LOCATION
E518	58648	INITIALIZE I/O
E531	58673	NORMALIZE SCREEN
E544	58692	CLEAR SCREEN
E566	58726	HOME CURSOR
E56C	58732	SET SCREEN POINTERS
E5A0	58784	SET I/O DEFAULTS
E5A9	58792	SET VIC CHIP DEFAULTS
E5B4	58804	INPUT FROM KEYBOARD
E5CA	58826	WAIT FOR CARRIAGE RETURN
E632	58930	INPUT FROM SCREEN
E684	59012	QUOTE MARK TEST
E691	59025	SET UP SCREEN PRINT
E6B6	59062	ADVANCE CURSOR
E6F7	59127	RETREAT CURSOR
E701	59137	BACK INTO PREVIOUS LINE
E716	59158	OUTPUT TO SCREEN

HEX	DECIMAL	DESCRIPTION
E87C	59516	GOTO NEXT LINE
E891	59537	DO 'RETURN'
E8A1	59553	CHECK LINE DECREMENT
E8B3	59571	CHECK LINE INCREMENT
E8CB	59595	SET COLOR CODE
E8A1	59601	COLOR CODE TABLE
E8E2	59618	CODE CONVERSION
E8EA	59626	SCROLL SCREEN
E965	59749	OPEN SPACE ON SCREEN
E9C8	59848	MOVE SCREEN LINE
E9E0	59872	SYNCH COLOR TRANSFER
E9F0	59888	SET START OF LINE
E9FF	59903	CLEAR SCREEN LINE
EA13	59923	PRINT TO SCREEN
EA1C	59932	STORE ON SCREEN
EA24	59940	SYNCH COLOR TO CHARACTER
EA31	59953	INTERRUPT (IRQ)
EA87	60039	CHECK KEYBOARD
EAE0	60128	DECODE KEYSTROKE
EB48	60232	SET UP KEYBOARD DECODE TABLE
EB59	60249	SET TEXT MODE
EB79	60281	KEYBOARD VECTORS
EB91	60305	KEYBOARD MAPS
EBC2	60354	SHIFTED KEYBOARD MATRIX
EC44	60484	GRAPHICS/TEXT CONTROL
EC4F	60495	SET GRAPHICS MODE
EC5E	60510	CHECK FOR SPECIAL CHARACTERS
EC78	60536	TABLE OF PETASCII VALUES FOR KEYBOARD
ECB9	60601	VIDEO CHIP DEFAULT TABLE
ECF0	60656	LOW BYTE TABLE SCREEN LINE ADDRESSES
ED09	60681	SEND 'TALK'
ED0C	60684	SEND 'LISTEN'
ED11	60689	SEND CONTROL CHARACTER
ED40	60736	SEND TO SERIAL BUS
EDB0	60848	TIME OUT ON SERIAL
EDB9	60857	SEND LISTEN SA
EDBE	60862	CLEAR ATN
EDC7	60871	SEND TALK SA
EDDD	60893	SEND SERIAL DEFERRED
EDEF	60911	SEND 'UNTALK'
EDFE	60926	SEND 'UNLISTEN'
EE13	60947	RECIEVE FROM SERIAL BUS
EE85	61061	CLOCK LINE ON
EE8E	61070	CLOCK LINE OFF
EEB3	61107	DELAY 1 MS
EEBB	61115	RS-232 SEND (NMI)
EF06	61190	NEW RS-232 BYTE SEND
EF2E	61230	RS-232 ERRORS OR QUIT
EF4A	61258	COMPUTE BIT COUNT
EF59	61273	RS-232 RECIEVE (NMI)
EF7E	61310	SET UP TO RECIEVE
EF90	61328	TEST FOR BIT FROM RS-232
EF97	61335	PLACE BYTE IN RS-232 RECIEVE BUFFER
EFC5	61381	RECIEVE PARITY ERROR
EFCA	61386	RECIEVE OVERRUN ERROR

HEX	DECIMAL	DESCRIPTION
EFCD	61389	RECIEVE BREAK ERROR
EFD0	61392	RECIEVE FRAME ERROR
EFE1	61409	FILE TO RS-232
F017	61463	SEND TO RS-232 BUFFER
F04D	61517	INPUT FROM RS-232 BUFFER
F086	61574	GET FROM RS-232 BUFFER
F0A4	61604	CHECK SERIAL BUS IDLE
F0BD	61629	MESSAGES
F12B	61739	PRINT IF DIRECT
F13E	61758	GET..
F14E	61774	..FROM RS-232
F157	61783	INPUT
F199	61849	GET..TAPE/SERIAL/RS-232
F1CA	61898	OUTPUT A BYTE
F1DD	61917	..TO TAPE
F20E	61966	SET INPUT DEVICE
F250	62032	SET OUTPUT DEVICE
F291	62097	CLOSE
F30F	62223	FIND FILE
F31F	62239	SET FILE VALUES
F32F	62255	ABORT ALL FILES
F333	62259	RESTORE DEFAULT INPUT/OUTPUT
F34A	62282	DO FILE OPENING
F3D5	62421	SEND SA
F409	62473	OPEN RS-232
F49E	62622	LOAD PROGRAM
F5A5	62885	'SEARCHING'
F5B8	62904	PRINT FILE NAME
F5D2	62930	'LOADING/VERIFY'
F5DD	62941	SAVE PROGRAM
F68F	63119	'SAVING'
F69B	63131	BUMP CLOCK
F6DD	63197	GET TIME
F6E4	63204	SET TIME
F6ED	63213	ACTION STOP KEY
F6FB	63227	FILE ERROR MESSAGES
F72C	63276	FIND ANY TAPE HEADER
F76A	63338	WRITE TAPE HEADER
F7D0	63440	GET BUFFER ADDRESS
F7D7	63447	SET BUFFER START-END POINTERS
F7EA	63466	FIND SPECIFIC HEADER
F80D	63501	BUMP TAPE POINTER
F817	63511	'PRESS PLAY'
F82E	63534	CHECK CASSETTE STATUS
F838	63544	'PRESS RECORD'
F841	63553	INITIATE TAPE READ
F864	63588	INITIATE TAPE WRITE
F875	63605	COMMON TAPE READ/WRITE
F8D0	63696	CHECK TAPE STOP
F8E2	63714	SET TIMING
F92C	63788	READ BITS (IRQ)
FA60	64096	STORE CHARACTERS
FB8E	64398	RESET POINTER
FB97	64407	NEW TAPE CHARACTER SETUP
FBA6	64422	TOGGLE TAPE

HEX	DECIMAL	DESCRIPTION
FBC8	64456	DATA WRITE
FBCD	64461	TAPE WRITE (IRQ)
FC57	64599	LEADER WRITE (IRQ)
FBC8	64618	WRITE TO CASSETTE
FC93	64659	RESTORE VECTORS
FCB8	64696	SET VECTOR
FCCA	64714	KILL MOTOR
FCD1	64721	CHECK READ/WRITE POINTER
FCDB	64731	BUMP READ/WRITE POINTER
FCE2	64738	POWERUP ENTRY
FD02	64770	CHECK A-ROM
FD10	64784	CHECK FOR AUTOSTART CARTRIDGE
FD13	64787	SET KERNAL
FD30	64816	TABLE OF RAM VECTORS
FD30	64848	PERFORM RAM TEST
FD52	64850	INITIALIZE SYSTEM CONSTANTS
FD9B	64923	IRQ VECTORS
FDA3	64931	INITIALIZE INPUT/OUTPUT REGISTERS
FDF9	65017	SAVE DATA NAME
FE00	65024	SAVE FILE DETAILS
FE07	65031	GET STATUS
FE18	65048	FLAG ST
FE21	65057	SET TIME OUT
FE25	65061	READ/SET TOP MEMORY
FE34	65076	READ/SET BOTTOM OF MEMORY
FE43	65091	NMI INTERRUPT ENTRY
FE66	65126	RESET/STOP WARM START
FE72	65138	RS-232 HANDLER/NMI
FEBC	65214	RESTORE AND EXIT
FEC2	65218	RS-232 TIMING TABLE
FED6	65238	RS-232 RECIEVE NEXT BIT
FF48	65352	MAIN IRQ ENTRY
FF5B	65371	INITIALIZE SCREEN EDITOR/VIC CHIP
		** JUMBO JUMP TABLE **
FF81	65409	INITIALIZE SCREEN EDITIOR
FF84	65412	INITIALIZE I/O
FF87	65415	INITIALIZE RAM. TAPE BUFFER & SET SCREEN
FF8A	65418	RESTORE DEFAULT I/O VECTORS
FF8D	65421	READ/SET VECTORED I/O
FF90	65424	CONTROL KERNAL MESSAGES
FF93	65427	SEND SECONDARY ADDRESS AFTER LISTEN
FF96	65430	SEND SECONDARY ADDRESS AFTER TALK
FF99	65433	READ AND SET TOP OF MEMORY
FF9C	65436	READ AND SET BOTTOM OF MEMORY
FF9F	65439	SCAN KEYBOARD
FFA2	65442	SET TIME OUT ON THE SERIAL BUS
FFA5	65445	INPUT BYTE FROM SERIAL PORT
FFA8	65448	OUTPUT BYTE TO SERIAL PORT
FFAB	65451	COMMAND SERIAL BUS TO UNTALK
FFAE	65454	COMMAND SERIAL BUS TO UNLISTEN
FFB1	65457	COMMAND SERIAL BUS TO LISTEN
FFB4	65460	COMMAND SERIAL BUS TO TALK
FFB7	65463	READ STATUS OF INPUT/OUTPUT

HEX	DECIMAL	DESCRIPTION
FFBA	65466	SET LOGICAL, FIRST & SECOND ADDRESS
FFBD	65469	SET FILE NAME
FFC0	65472	OPEN LOGICAL FILE
FFC3	65475	CLOSE LOGICAL FILE
FFC6	65478	OPEN CHANNEL FOR INPUT
FFC9	65481	OPEN CHANNEL FOR OUTPUT
FFCC	65484	CLOSE INPUT/OUTPUT CHANNELS
FFCF	65487	INPUT CHARACTER FROM CHANNEL
FFD2	65490	OUTPUT CHARACTER TO CHANNEL
FFD5	65493	LOAD RAM FROM A DEVICE
FFD8	65496	SAVE RAM TO A DEVICE
FFDB	65499	SET TIME CLOCK
FFDE	65502	READ TIME CLOCK
FFE1	65505	SCAN STOP KEY
FFE4	65508	GET CHARACTER FROM DEVICE
FFE7	65511	CLOSE ALL CHANNELS & FILES
FFEA	65514	INCREMENT TIME CLOCK
FFED	65517	RETURN X,Y ORGANIZATION OF SCREEN
FFF0	65520	READ SET X,Y CURSOR POSITION
FFF3	65523	RETURN BASE ADDRESS OF INPUT/OUTPUT DEVICE
FFFA	65530	HARDWARE VECTORS
FFFC	65532	SYSTEM RESET

HELP CHART

DEC	HEX	BASIC	6510	ASCII	SCREEN	
			OPS CODES			
000	0	END LINE	BRK		@	
001	1		ORA($ZZ ,X)		A	a
002	2				B	b
003	3				C	c
004	4				D	d
005	5		ORA $ZZ	WHITE	E	e
006	6		ALS $ZZ		F	f
007	7				G	g
008	8		PHP	DISABLE SHIFT/CMDR	H	h
009	9		ORA$$ii	ENABLE SHIFT/CMDR	I	i
010	A		ASL		J	j
011	B				K	k
012	C				L	l
013	D		ORA$nnnn	RETURN	M	m
014	E		ASL$nnnn	LOWER CASE	N	n
015	F				O	o
016	10		BPL$nnnn		P	p
017	11		ORA($zz),Y	CURSOR DOWN	Q	q
018	12			REVERSE ON	R	r
019	13			CLEAR HOME	S	s
020	14			INSERT DELETE	T	t
021	15		ORA$zz ,X		U	u
022	16		ASL$zz ,X		V	v
023	17				W	w
024	18		LLC		X	x
025	19		ORA$nnnn ,Y		Y	y
026	1A				Z	z
027	1B				[	
028	1C			RED	£	
029	1D		ORA$nnnn ,X	CURSOR RIGHT	]	
030	1E		ASL$nnnn ,X	GREEN	↑	
031	1F			BLUE	←	
032	20	SPACE	JSR$nnnn	SPACE	SPACE	
033	21	!	AND($zz ,X)	!	!	
034	22	"		"	"	
035	23	#		#	#	
036	24	$	BIT$zz	$	$	
037	25	%	AND$zz	%	%	
038	26	&	RDL$zz	&	&	
039	27	'		'	'	
040	28	(	PLP	(	(	
041	29	)	AND$$ii	)	)	
042	2A	*	ROL	*	*	
043	2B	+		+	+	
044	2C	,	BIT$nnnn	,	,	
045	2D	-	AND$nnnn	-	-	
046	2E	.	ROL$nnnn	.	.	
047	2F	/		/	/	
048	30	0	BMI$nn	0	0	
049	31	1	AND($zz),Y	1	1	
050	32	2		2	2	
051	33	3		3	3	

DEC	HEX	BASIC	6510 OPS CODES	ASCII	SCREEN	
052	34	4		4	4	4
053	35	5	AND$zz,X	5	5	5
054	36	6	ROL$zz,X	6	6	6
055	37	7		7	7	7
056	38	8	SEC	8	8	8
057	39	9	AND$nnnn,Y	9	9	9
058	3A	:		:	:	:
059	3B	;		;	;	;
060	3C	<		<	<	<
061	3D	=	AND$nnnn,X	=	=	=
062	3E	>	ROL$nnnn,X	>	>	>
063	3F	?		?	?	?
064	40	@	RTI	@	GRAPHICS	
065	41	A	EOR($zz,X)	A	"	A
066	42	B		B	"	B
067	43	C		C	"	C
068	44	D		D	"	D
069	45	E	EOR$zz	E	"	E
070	46	F	LSR$zz	F	"	F
071	47	G		G	"	G
072	48	H	PHA	H	"	H
073	49	I	EOR#$ii	I	"	I
074	4A	J	LSR	J	"	J
075	4B	K		K	"	K
076	4C	L	JMP$nnnn	L	"	L
077	4D	M	EOR$nnnn	M	"	M
078	4E	N	LSR$nnnn	N	"	N
079	4F	O		O	"	O
080	50	P	BVC$nn	P	"	P
081	51	Q	EOR($zz),Y	Q	"	Q
082	52	R		R	"	R
083	53	S		S	"	S
084	54	T		T	"	T
085	55	U	EOR$zz,X	U	"	U
086	56	V	LSR#zz,X	V	"	V
087	57	W		W	"	W
088	58	X	CLI	X	"	X
089	59	Y	EOR$zz,Y	Y	"	Y
090	5A	Z		Z	"	Z
091	5B	a		LEFT BRACKET		
092	5C	b		ENGLISH POUND		
093	5D	c	EOR$nnnn,X	↑	GRAPHICS	
094	5E	d	LSR$nnnn,X	↑	"	"
095	5F	e		←	"	"
096	60	f	RTS	GRAPHICS	SPACE	
097	61	g	ADC($zz),X	"	GRAPHICS	
098	62	h		"	"	"
099	63	i			"	"
100	64	j				
101	65	k	ADC$zz			
102	66	l	ROR$zz			
103	67	m				
104	68	n	PLA			
105	69	o	ADC#$ii			

DEC	HEX	BASIC	6510 OPS CODES	ASCII	SCREEN
106	6A	p	ROR		
107	6B	q			
108	6C	r	JMP($nnnn)		
109	6D	s	ADC$nnnn		
110	6E	t	ROR$nnnn		
111	6F	u			
112	70	v	BVS$nn		
113	71	w	ADC($zz),y		
114	72	x			
115	73	y			
116	74	z			
117	75		ADC$zz,X		
118	76		RDR$zz,X		
119	77				
120	78		SEI		
121	79		ADC$nnnn,Y		
122	7A				
123	7B				
124	7C				
125	7D		ADC$nnnn,X		
126	7E		ROR$nnnn,X		
127	7F				
128	80	END			
129	81	FOR	STA($zz,X)	ORANGE	
130	82	NEXT			
131	83	DATA			
132	84	INPUT#	STA$zz		
133	85	INPUT	STA$zz	F1	
134	86	DIN	STX$zz	F3	
135	87	READ		F5	
136	88	LET	DEY	F7	
137	89	GOTO		F2	
138	8A	RUN	TXA	F4	
139	8B	IF		F6	
140	8C	RESTORE	STY$nnnn	F8	
141	8D	GOSUB	STA$nnnn	SHIFT/RETURN	
142	8E	RETURN	STX$nnnn	UPPER CASE	
143	8F	REM			
144	90	STOP	BCC$nn	BLACK	
145	91	ON	STA($zz),Y	CURSOR UP	
146	92	WAIT		REVERSE OFF	
147	93	LOAD		CLEAR HOME	
148	94	SAVE	STY$zz,x	INSERT DELETE	
149	95	VERIFY	STA$zz,X	BROWN	
150	96	DEF	STX$zz,Y	LIGHT RED	
151	97	POKE		LIGHT GREY	
152	98	PRINT #	TYA	MED. GREY	
153	99	PRINT	STA$nnnn,Y	LIGHT GREEN	
154	9A	CONT	TXS	LIGHT BLUE	
155	9B	LIST		DARK GREY	
156	9C	CLR		PURPLE	
157	9D	CMD	STA$nnnn,x	CURSOR LEFT	
158	9E	SYS		YELLOW	
159	9F	OPEN		CYAN	

DEC	HEX	BASIC	6510 OPS CODES	ASCII	SCREEN
160	A0	CLOSE	LDY#$ii	SPACE	
161	A1	GET	LDA($zz,X)	GRAPHICS	
162	A2	NEW	LDX#$ii	" "	
163	A3	TAB(		" "	
164	A4	TO	LDY$zz		
165	A5	FN	LDA$zz		
166	A6	SPC(	LDX$zz		
167	A7	THEN			
168	A8	NOT	TAY		
169	A9	STEP	LDA#$ii		
170	AA	+	TAX		
171	AB	-			
172	AC	$	LDY$nnnn		
173	AD	/	LDA$nnnn		
174	AE	↑	LDX$nnnn		
175	AF	AND			
176	B0	OR	BCS$nn		
177	B1	>	LDA($zz),Y		
178	B2	=			
179	B3	<			
180	B4	SGN	LDY$ZZ,X		
181	B5	INT	LDA$ZZ,X		
182	B6	ABS	LDX$ZZ,Y		
183	B7	USR			
184	B8	FRE	CLV		
185	B9	POS	LDA$nnnn,Y		
186	BA	SQR	TSX		
187	BB	RND			
188	BC	LOG	LDY$nnnn,X		
189	BD	EXP	LDA$nnnn,X		
190	BE	COS	LDX$nnnn,Y		
191	BF	SIN			
192	C0	TAN	CPY#$ii		
193	C1	ATN	CMP(zz,X)		
194	C2	PEEK			
195	C3	LEN			
196	C4	STR$	CPY$zz		
137	C5	VAL	CMP$zz		
198	C6	ASC	DEC$zz		
199	C7	CHR$			
200	C8	LEFT$	INY		
201	C9	RIGHT$	CMP$zz		
202	CA	MID$	DEX		
203	CB	GO			
204	CC		CPY$nnnn		
205	CD		CMP$nnnn		
206	CE		DEC$nnnn		
207	CF				
208	D0		BNE$nn		
209	D1		CMP$($zz),Y		
210	D2				
211	D3				
212	D4				
213	D5		CMP$zz,X		

DEC	HEX	BASIC	6510 OPS CODES	ASCII	SCREEN
214	D6		DEC$ZZ,X		
215	D7				
216	D8		CLD		
217	D9		CMP$nnnn,Y		
218	DA				
219	DB				
220	DC				
221	DD		CMP$nnnn,X		
222	DE		DEC$nnnn,X		
223	DF				
224	E0		CPX#$ii		
225	E1		SBC($zz,X)		
226	E2				
227	E3				
228	E4		CPX$zz		
229	E5		SBC($zz,X)		
230	E6		INC$zz		
231	E7				
232	E8		INX		
233	E9		SBC$zz		
234	EA		NOP		
235	EB				
236	EC		COX$nnnn		
237	ED		SBC$nnnn		
238	EE		INC$nnnn		
239	EF				
240	F0		BEQ$nn		
241	F1		SBC($zz),Y		
242	F2				
243	F3				
244	F4				
245	F5		SBC$zz,X		
246	F6		INC$zz,X		
247	F7				
248	F8		SED		
249	F9		SBC$nnnn,Y		
250	FA				
251	FB				
252	FC				
253	FD		SBC$nnnn,X		
254	FE		INC$nnnn,X		
255	FF				

For the beginner or advanced

programmer ______________

THIS IS YOUR BOOK !!

The BRAIN is a clearly written detailed explanation of the Commodore 64 computer's memory. If you are a begginer here is the information you have been hunting for. Advanced programmers will use this booklet again and again as a powerful tool. THE BRAIN is a no-nonsense approach to a guided tour of the most extensive memory map on the market todate.

☆ Written in dictionary format

☆ Easy to understand terms

☆ Extensive sound & graphic sections

☆ Understand existing programs

☆ Identify program storage locations

☆ The Kernal at your finger tips

☆ Addresses in hex and decimal

☆ OPS code help chart

THE BRAIN includes a poster size (21x25) chart of screen and color memory. Every location in print. Stop hunting, do it the easy way like a professional programmer.

Created by users....

for users!!