

# The Complete

www.Commodore.ca  
May Not Reprint Without Permission

# Commodore Inner Space Anthology

Karl J.H. Hildon

## Commodore Disk Specifications

Model	8900	8900	8250	8050	4040	4051	4051
Drives per Head	1	1	2	2	2	2	2
Heads per Drive	6	4	2	2	2	2	2
<b>Formatted Storage</b>							
Capacity per Unit	25.7MB	4.9MB	2.12MB	1.05MB	530KB	170KB	170KB
Max Sequential Files/Drive	741 MB	4.91MB	1.05MB	521 KB	168 KB	168 KB	168 KB
Max Relative Files/Drive	135 MB	4.90MB	1.04 MB	183 KB	167 KB	167 KB	167 KB
Disk Systems Buffer	4KB	4KB	4KB	4KB	4KB	4KB	4KB
<b>Disk Formats</b>							
Cylinders (Tracks)	193	153	77	77	35	35	35
Tracks per Cylinder	120	192	23-29	23-29	17-21	17-21	17-21
Sectors per Track	256	256	256	256	256	256	256
Bytes per Sector	256	256	256	256	256	256	256
Stack Time	3700	3700	3700	3700	3700	3700	3700
<b>Transfer Rates (bytes per second)</b>							
Internal to Unit	200	200	400	400	400	400	400
Unit to Unit	200	200	400	400	400	400	400
<b>Physical Dimensions</b>							
Height (inches)	4.75	5.75	7.0	7.0	7.0	5.5	5.0
Width (inches)	8.25	8.25	15.0	15.0	15.0	19.0	20.0
Depth (inches)	15.25	15.25	13.75	13.75	13.75	14.25	13.0
Weight (pounds)	21	21	28	28	28	20	10
<b>Disk Utility</b>							
Command	Abbreviation	Format	Number of Cylinders				
Format	Format	Format	Format				

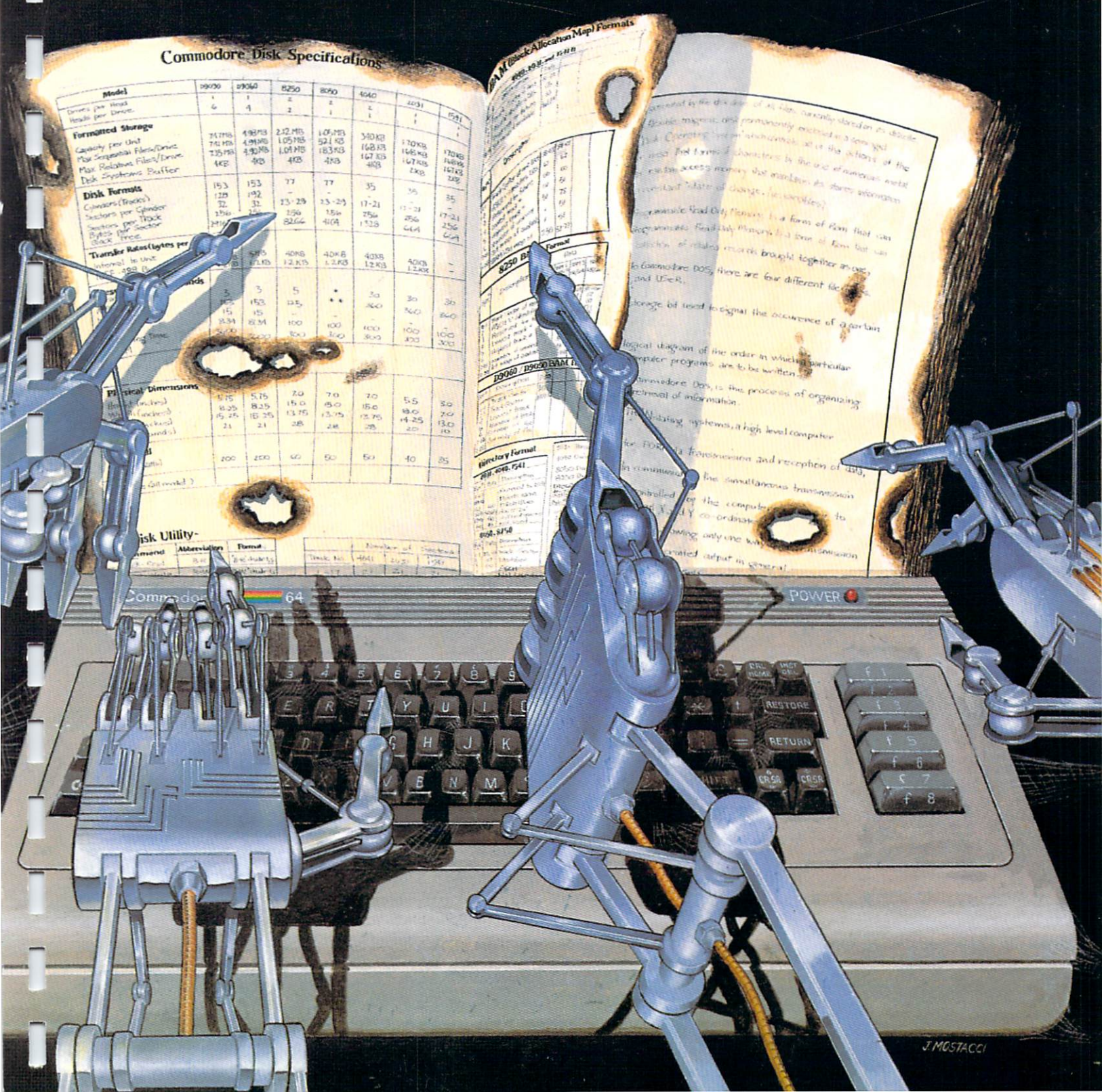
...ment is the disk drive of its own, currently standard in its drive  
 ...drive, magnetic, and permanently mounted in a separate  
 ...disk. Operating system instructions are in the form of the  
 ...disk. The format of instructions is in the form of numerical words  
 ...disk. Access memory that maintain its status information  
 ...control. Make a change in the system.

...programmable Read Only Memory (ROM) form that can  
 ...programmable Read Only Memory (ROM) form that can  
 ...selection of using memory brought together in one  
 ...and USER. There are four different files.

...storage of text to signal the occurrence of a certain  
 ...logical diagram of the order in which particular  
 ...computer programs are to be written.

...Commodore Disk is the process of organizing  
 ...retention of information.

...In many systems, a high level computer  
 ...for ROM is transmission and reception of data,  
 ...in connection with the simultaneous transmission  
 ...controlled by the computer.  
 ...and X-Y co-ordinates to  
 ...allowing only one way transmission  
 ...generated output in general.





# The Complete Commodore Inner Space Anthology

Karl J.H. Hildon



## The Making Of. . .

What you see before you is the collection, culmination, and collation of almost 5 years of information about Commodore Computers. It all began with The Best of The Transactor Volume 2 and a photocopier with a reduction feature. It occurred to me that if all my most referenced facts were together on one page they would be infinitely more useful. Memory maps, conversion charts, machine code tables, and everything else went into the copier over and over until they were small enough to paste together on one sheet. But the photocopier had its drawbacks; each new reduction meant a drop in quality and the distortion factor of the copier had the top lines slanting down and the bottom lines slanting up.

After I departed from Commodore to run The Transactor independently, I was thrust into the world of the phototypesetter, the ultimate printer. At first I was totally consumed by the superb quality of the type, but that didn't last long. I began experimenting with point sizes (character size), leading (line spacing), and the over 300 other commands that are available including an entire text programming language. With vertical spacing down to  $\frac{1}{576}$ th of an inch and horizontal accuracy to  $\frac{1}{1296}$ th of an inch, I found myself accounting

for every fraction. This exact science of typesetting was the perfect answer to the question of how the next generation of compact reference material would be created.

After about eight months of practice I decided it was time. Four months later The Special Reference Issue of The Transactor (Volume 4, Issue 5) was released. The brown cover earned it the nickname, "The Brown Bible" and it wasn't long before many were referring to it as "the most photocopied magazine of all time". Everyone seemed to be happy with it, except me.

It was about six months later when Attic Typesetting took delivery of the first Quadex Preview in Canada, a fabulous device that shows on a screen exactly what the type machine will produce. Typesetting: the Science, became Typesetting: the Art. It was then I decided the next generation was within my reach. Although the Preview simplified the task by easily ten-fold, the amount of target material had more than tripled. After eight months of organizing (in the time between making magazines) and almost two months of double shifts at the type shop, I now find myself writing this paragraph. The Complete Commodore Inner Space Anthology is finally finished.

## Acknowledgements

Special thanks to Richard T. Evers and Chris J. Zamara: two very special talents inside two very special individuals. Invaluable assistance lacked a true definition until you guys.

Extra special thanks to Jim Butterfield: Jim was responsible for the memory maps of all the computers, each one a masterpiece of information dissemination. The original idea of the SuperChart was also Jim's. Your influence and inspiration are exceeded only by your generosity, three quantities I could only hope my appreciation might one day equal.

Attic Typesetting, namely Phyllis Fast and Nate Redmon: your patience and understanding are outweighed only by your typesetting equipment.

Special thanks to Bill Maclean: for backing me up, all the way.

Others I wish to thank include Len Lindsay for providing COMAL memory maps and other valuable data; Jim Gracely of Commodore for providing the Computer Club listing; Nick Sullivan, Editor of TPUG Magazine, for necessary data to create the Chord Derivatives; David Berezowski for finding me a MOS Data Catalog; Domenic DeFrancesco for his help with hardware problems; Jim Yost, Louis Sander, and Colin Arnel for sending in their notes that allowed for improvements; and Raeto Collin West for setting the standard with Programming the PET/CBM.

Cover Design by John Mostacci

Printed in Canada

ISBN 0-9692086-0-X

© March 1985 by Transactor Publishing Incorporated, 500 Steeles Avenue, Milton, Ontario, L9T 3P7 (416-876-4741). Although the information in this book is public domain, the presentation of said information may not be duplicated. Photocopying or visual reproduction of any kind for other than personal use will not be tolerated without written permission from Transactor Publishing Incorporated. Although accuracy is a major objective, Transactor Publishing can assume no liability for errors.

Dedicated to John A. Hildon, my dad.

# The Complete Commodore Inner Space Anthology

## SuperCharts

- 29 BASIC 2.0/4.0 SuperChart
- 37 VIC 20/Commodore 64 SuperChart
- 73 TRUE ASCII Conversion Chart
- 73 Binary Conversion Chart
- 73 Parity Tables
- 73 BCD Conversion Chart

## BASIC Section

- 1 Commands and Statements
- 2 String Functions
- 2 Arithmetic Functions
- 3 Arithmetic Operators
- 3 Special Symbols
- 3 Hierarchy of Operations
- 3 Reserved Variables
- 3 BASIC 4.0 Disk Commands
- 4 BASIC RAM Memory Allocation
- 4 BASIC Text Line Structure
- 4 Variable Formats
- 4 'FOR' Stack Entry
- 4 'GOSUB' Stack Entry
- 4 Reserved Variables: ST, DS, DS\$
- 5 Additional B Series Commands
- 5 Additional +4/C16 Commands
- 6 B/ +4/C16 Escape Key Sequences
- 7 BASIC 2.0/4.0 Error Messages
- 8 B Series/ +4/C16 Error Messages
- 9 BASIC Abbreviations
- 10 C64 Super Expander Commands

## COMAL Section

- 11 Reserved Variables
- 11 COMAL Commands
- 12 Sprite Commands
- 12 Turtle Graphics Commands
- 12 COMAL 2.0 Library Descriptions
- 13 COMAL 2.0 Memory Map
- 15 COMAL 0.14 Memory Map

## Printer Section

- 16 Matrix Printer Control Characters
- 16 Matrix Printer Format Characters
- 16 Letter Quality Printer Commands
- 16 Greek Alphabet Characters

## Business Software Section

- 17 Wordprocessing Reference Guide
- 19 Spreadsheet Commands
- 20 +4: 3+1 Software Commands

## Machine Language Section

- 21 Machine Language Monitor Commands
- 21 Assembler Commands
- 22 CPU Model
- 22 Pocket Op-Codes Chart
- 22 6502 Extra Op-Codes
- 22 Hexadecimal Conversion Table
- 23 Instruction Set Summary
- 25 Instruction Set Descriptions
- 25 Addressing Modes
- 26 User Callable ROM Routines
- 27 BASIC 2.0/4.0 Kernal Routines
- 27 VIC 20/Commodore 64 Kernal Routines
- 28 Keyword Tokens and Entry Points

## Memory Maps

- 31 BASIC 2.0/4.0 RAM, ROM, I/O
- 33 BASIC 2.0/4.0 Zero Page Contents
- 35 VIC 20 RAM, ROM, I/O
- 39 Commodore 64 RAM, ROM, I/O
- 41 VIC 20/C64 Zero Page Contents
- 43 B Series RAM, ROM, I/O
- 45 +4/C16 RAM, ROM, I/O
- 50 4040 Memory Map
- 54 8050 Memory Map
- 57 1541 Memory Map

## **Disk Drives Section**

- 47 Disk Specifications
- 47 Directory Header Formats
- 47 Directory Sector Formats
- 48 Block Availability Map Formats
- 48 Sector Recording Format
- 49 Data File Format
- 49 PET/CBM Disk Access Routines
- 49 Utility Command Set
- 49 User Command Jump Table
- 49 LED Error Diagnostics
- 49 Track/Sector Distribution Table
- 49 GCR Codes
- 50 4040 Memory Map
- 54 8050 Memory Map
- 57 1541 Memory Map

## **Music Section**

- 60 Music Symbols
- 61 Note Frequency Table
- 61 Chord Note Derivatives
- 62 CB2 Note Values
- 62 VIC 20 Note Values
- 62 Commodore 64 SID Note Values
- 62 Commodore 64 ADSR Envelope Values
- 62 +4/C16 SOUND Values

## **Video Section**

- 63 VIC 20 Screen and Border Colours
- 63 6845 Video Chip Registers
- 63 Colour Codes
- 63 8032 Screen Control Characters
- 63 Secondary Address Table
- 64 VIC 20 Screen Memory Addresses
- 64 VIC 20 Character Base Addresses
- 64 Commodore 64 Screen Memory
- 64 Commodore 64 VIC II Chip Addresses
- 64 Commodore 64 Character Base
- 64 Character ROM Contents
- 65 Sprite Design
- 66 Programmable Character Design
- 66 PET/CBM 40 Column Screen Map
- 67 VIC 20 Screen and Colour Table Maps
- 69 C64 Screen and Colour Table Maps
- 70 80 Column Screen Map
- 71 B Series 80 Column Screen Map
- 72 +4/C16 Screen and Colour Table Maps
- 73 Decimal Page Boundary Addresses

## **Telecomputing Section**

- 75 Network Phone Numbers
- 77 CompuServe Commands
- 78 CompuServe Category Index
- 79 Bulletin Boards by Area Code
- 84 Time Zone and Area Code Map
- 85 Bulletin Boards in Alphabetical Order
- 90 Computer Clubs

## **Hardware Section**

- 97 Tape Recording Format
- 97 Cassette Port
- 97 IEEE Standard Definitions
- 98 IEEE 488 Bus Signals
- 98 IEEE Byte Transfer Sequence
- 98 IEEE Cable Connector Pinouts
- 98 IEEE Port Pinouts
- 99 PET/CBM User Port
- 99 6522 Registers
- 99 Commodore 64 User Port
- 99 Commodore 64 Expansion Port
- 99 VIC 20/C64 Keyboard Matrix
- 100 VIC 20 I/O Ports
- 100 Commodore 64 I/O Ports
- 101 6520 PIA Registers
- 102 6522 VIA Control Registers
- 103 6526 CIA Control Registers
- 104 Commodore 64 Board Layout
- 104 Resistor Colour Codes
- 104 Transistor Lead Assignments
- 105 RS 232 and ACIA Control Registers
- 106 B Series I/O Ports
- 107 Chip Pinouts
- 109 Semiconductor Testing Guide

## **Arithmetic and Mathematics**

- 111 Inch Fractions
- 111 International System Of Units
- 112 Names For Large Numbers
- 112 Roman Numerals
- 112 Constant Values
- 112 Boolean Truth Table
- 112 Force Formulae
- 112 Mathematical Functions
- 112 Trigonometry Rules
- 113 Unit to Unit Conversion Tables
- 118 Geometric Areas and Volumes
- 121 Periodic Table Of The Elements

# BASIC – Beginners All-Purpose Symbolic Instruction Code

## Commands and Statements

Command/ Statement	Example	Purpose
CLOSE	10 CLOSE n	Closes logical file 'n'.
CLR	CLR	Sets variables to zero or null.
CMD	CMD D	Keep ieee device 'D' open to monitor bus.
CONT	CONT	Continue program execution after a stop command. No program changes are permitted.
DATA	10 DATA 1,2,3,4 20 DATA TOM, SUE 30 DATA "DOE, TOM"	Specifies data to be read left to right. Alphabets do not need to be enclosed in quotes. if strings contain spaces, commas, colons, or graphic characters, the string must be enclosed in quotes.
DEF	10 DEF FN R(X)	Defines function 'R'
DIM	10 DIM A(n) 20 DIM A(n,m,o,p) 30 DIM A(n),B(m) 40 DIM A(N) 50 DIM A\$(n)	Specifies maximum number of elements in an array or matrix. Specifies maximum number of dimensions in an array. Number of arrays limited by memory. May be dimensioned dynamically. Strings to be dimensioned.
END	999 END	Terminates program execution.
FOR	10 FOR I = 1 TO 10	Begins repetitive loop, specifying loop variable and number of intended iterations (in this example 'I' for 10 iterations).
FRE	PRINT FRE (0)	Returns number of bytes of available memory.
GET	10 GET C 20 GET C\$ 30 GET #d, C 40 GET #d, C\$	Accepts single numeric character from keyboard. Accepts single string character from keyboard. Accepts single character from specified logical file. Accepts specified single string character from logical file.
GOSUB	10 GOSUB n	Begins execution of a subroutine which begins on line 'n'.
GOTO	10 GOTO n	Transfer program execution to line n.
IF...GOTO	10 IF X = 10 GOTO n	Transfers execution to line 'n' if result of condition is true.
IF...THEN	10 IF X = 10 THEN Y = 3	Code following THEN is executed only if result of condition is true. May also be followed by line number to transfer execution.
INPUT	10 INPUT A 20 INPUT A\$ 30 INPUT A,A\$,B,B\$ 40 INPUT #d, A 50 INPUT #d, a\$ 60 INPUT #d, A,A\$,B,B\$	Accepts value of 'A' from keyboard. Accepts value of string variable 'A' from keyboard. The string does not have to be enclosed in quotes. Accepts specified values from keyboard. Accepts value of 'A' from logical file 'd'. accepts specified string from logical file 'd'. Accepts specified values and string from logical file 'd'. Strings do not have to be enclosed in quotes.
LET	LET X = 10	Optional. Assigns variable 'X' the value of 10.
LIST	LIST LIST -n LIST n-m LIST n-	Lists current program. Lists current program through line 'n'. Lists lines 'n' through 'm' of current program. Lists current program from line 'n' to end.
LOAD	10 LOAD 20 LOAD "NAME" 30 LOAD "NAME", d 30 LOAD "NAME", d, c	Loads next encountered program from tape unit into memory. Loads program or file 'NAME' into memory from tape unit. Loads specified file 'NAME' from device 'd'. Loads specified file 'NAME' from device 'd' for command 'c'. (VIC/C64 only - c = 1 for direct memory load)
NEW	NEW	Deletes current program in memory, sets variables to zero.
NEXT	NEXT	Indicates end of code contained in a FOR/NEXT loop.
ON...GOSUB	10 ON A GOSUB I, m, n	Begins execution of subroutine which begins on specified line (in this example, 'I', 'm', or 'n') depending on value of index 'A'.
ON...GOTO	10 ON A GOTO I, m, n	Transfers control to specified line 'I', 'm', or 'n' depending on value of index 'A'.
OPEN	10 OPEN a 20 OPEN a, d 30 OPEN a, d, c 40 OPEN a, d, c, "NAME"	Opens logical file 'a' for read only from tape unit. Opens logical file 'a' for read only from device 'd'. Opens logical file 'a' for command 'c' from device 'd'. Opens logical file 'a' on device 'd'. If device 'd' accepts formatted files, file name is positioned for command.
PEEK	PEEK(a) PEEK(A)	Returns byte value from address 'a'. Address can be dynamic.
POKE	POKE a, b POKE A, B	Puts byte 'b' into address 'a'. Parameters can be dynamic.
POS	10 PRINT POS(0)	Prints next available print position (position of cursor on screen).
PRINT	10 PRINT A 20 PRINT A\$ 30 PRINT A, A\$  40 PRINT #d, A 50 PRINT #d, A\$	Prints value 'A' on display screen. Prints specified string on screen. Prints specified values or strings on screen, beginning in next available print position (pre-tabbed positions are in columns 10,20,30,40 etc.). Prints value of 'A' on device 'd'. Prints specified string on device 'd'.
READ	10 READ A\$, B\$	Reads next two data elements into variables A\$ and B\$.
REM	10 REM Comment	Remark indicator. Execution skips entire line.
RESTORE	10 RESTORE	Resets data pointer so that next READ receives first element of first DATA statement.

## Commands and Statements, cont'd

Command/Statement	Example	Purpose
RETURN	9990 RETURN	Subroutine exit; transfers control to the statement following most recent gosub directing transfer to the subroutine.
RUN	RUN RUN n	Begins execution of program at lowest line number. Begins execution of program a line 'n'.
SAVE	SAVE "NAME" SAVE "NAME", d SAVE "NAME", d, c	Saves current file or program 'NAME' on tape unit. Saves current program or file 'NAME' on device 'd'. Saves file 'NAME' on device 'd'. 'c' specifies eof or eot.
STEP	10 FOR I = 1 TO 10 STEP 2	Alters loop variable increment.
STOP	STOP	Stops program execution.
SYS	SYS (x)	Complete control is transferred to a machine language program at the decimal address contained in the argument. Brackets optional.
USR	USR (x)	Transfers program control to a program whose address is at locations 1 and 2 (VIC/C64 - locations 784,785). 'x' is a parameter passed to and from the machine language program.
VERIFY	VERIFY VERIFY "NAME" VERIFY "NAME", d	Verifies current program against next program on tape unit. Verifies current program against program 'NAME' on tape unit. Verifies current program 'NAME' on device 'd'.
WAIT	WAIT a, b, c	Halts execution of Basic until contents of address 'a', and'ed with value 'b' and exclusive or'ed with value 'c', is not equal to zero. 'c' is optional and defaults to zero.

## String Functions

Function	Example	Purpose
ASC	10 A = ASC("XYZ")	Returns the integer value corresponding to ASCII code of the first character in string.
CHR\$	10 A\$ = CHR\$(n)	Returns character corresponding to ASCII code number.
LEFT\$	10 PRINT LEFT\$(X\$, a)	Returns leftmost 'a' characters from string.
LEN	10 PRINT LEN(X\$)	Returns length of string.
MID\$	10 PRINT MID\$(X\$, a, b)	Returns 'b' characters from string, starting with the 'a'th character.
RIGHT\$	10 PRINT RIGHT\$(X\$, a)	Returns rightmost 'a' characters from string.
STR\$	10 A\$ = STR\$(A)	Returns string representation of variable 'A'
VAL	10 A = VAL(A\$) 20 A = VAL("A")	Returns numeric representation of string. If string not numeric, returns "0".

ASC, LEN and VAL functions return numeric results. They may be used as part of any numerical expression. Assignment statements are used here for examples only; other statement types may be used.

## Arithmetic Functions

Function	Example	Purpose
ABS	10 C = ABS(A)	Returns magnitude of argument without regard to sign.
ATN	10 C = ATN(A)	Returns arctangent of argument. 'c' will be expressed in radians.
COS	10 C = COS(A)	Returns cosine of argument. 'A' must be expressed in radians.
DEF FN	10 DEF FNA(B) = C*D	Allows user to define a function. Function label 'a' must be a single letter; argument 'b' is a dummy.
EXP	10 C = EXP(A)	Returns constant 'e' raised to the power of the argument.
INT	10 C = INT(A)	Returns largest integer less than or equal to argument.
LOG	10 C = LOG(A)	Returns natural logarithm of argument. Argument must be greater than or equal to zero.
RND	10 C = RND(A)	Generates a random number between zero and one. If 'a' is less than 0, the same random number is produced in each call to rnd. If 'a' = 0, the same sequence of random number is generated each time rnd is called. If 'a' is greater than 0, a new sequence is produced for each call to rnd.
SGN	10 C = SGN(A)	Returns -1 if argument is negative, returns 0 if argument is zero, and returns + 1 if argument is positive.
SIN	10 C = SIN(A)	Returns sin of argument. 'A' must be expressed in radians.
SQR	10 C = SQR(A)	Returns the square root of argument.
TAN	10 C = TAN(A)	Returns tangent of argument. 'A' must be expressed in radians.

## Arithmetic Operators

Symbol	Example	Purpose
=	10 A = B 20 LET A = B	Assigns a value to a variable. LET is optional.
↑	30 PRINT A^2	Exponentiation
/	40 C = A/B	Division.
*	50 C = A*B	Multiplication.
+	60 C = A + B	Addition.
-	70 C = A - B	Subtraction.
=	10 IF A = B THEN PRINT C	'A' Equals 'B'.
<>	10 IF A <> B THEN C = 4	'A' Does not equal 'b'.
<	10 IF A < B THEN C\$ = " X "	'A' is less than 'B'.
>	10 IF A > B THEN C\$ = " Y "	'A' is greater than 'B'.
<=	10 IF A <= B THEN C = 20	'A' is less than or equal to 'B'.
>=	10 IF A >= B THEN C = D - 1	'A' is greater than or equal to 'B'.
AND	10 IF A AND B THEN C = 9	'A' and 'B' must both be true for statement 10 to be true.
OR	20 IF A OR B THEN C = 9	'A' must be true or 'B' must be true for statement 20 to be true.
NOT	30 IF NOT A THEN PRINT C	Expression is true if 'A' is false.

Note: the numerical values used in the evaluation of logical comparisons are: 'true' is any non-zero number and 'false' is zero.

Operator	Description
( )	Brackets always dictate priority
↑	Exponentiation
-	Negation (unary minus)
* /	Multiplication & Division
+ -	Addition & Subtraction
< = >	Relational Operations
NOT	Logical NOT (Integer two's complement)
AND	Logical AND
OR	Logical OR

### Reserved Variables

Variable	Purpose
DS	Disk Status number (except 2.0)
DS\$	Disk Status string (except 2.0)
EL	Error Line (B Series/ + 4/C16 only)
ER	Error number (B Series/ + 4/C16 only)
ERR\$(	Error String array. See table for messages. (B Series/ + 4/C16 only)
TI	Time in Jiffies (1/60th's sec.) since power up or TI\$ reset (except B Series)
TI\$	Time in HHMMSS
ST	The Status variable. See table for functions.

### Special Symbols

Symbols	Example	Purpose
:	10 A = 1:B = 2:C = 3	Allows multiple statements on a line.
;	10 PRINT A;B 20 PRINT A\$;B\$	Suppress Carriage Return for same line printing. Optional after \$ or % variables.
.	X = 10.99	Decimal Point
,	10 PRINT A, B LOAD "NAME", d	Allows same line printing. Elements are separated and printed in pre-'tab'ed print positions ( columns 10,20,30, etc.). Separates parameters in load, save, open, mid\$, on..goto, etc.
?	10 ?A	Abbreviation for 'print'. Stores as one character; lists as word PRINT.
\$	10 A\$ = " ABCDEFG "	String identifier.
%	10 A% = INT(X)	Integer identifier.
"	10 A\$ = " ABCDEFG "	String enclosures.
π	10 C = π * D	Value of Pi 3.1415927.

### Basic 4.0 Disk Commands

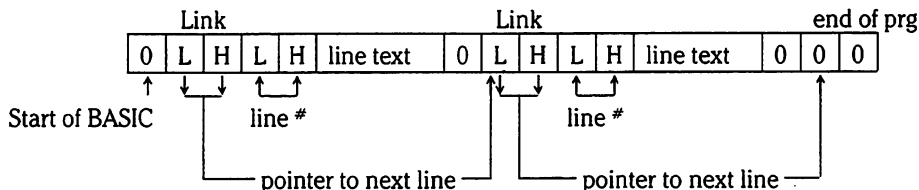
Function	Example	Purpose
APPEND	10 APPEND#d, " NAME "	Open file 'NAME' on device 'd' for appending. New data is added to end of existing data.
BACKUP	BACKUP D0 TO D1	Duplicate disk in drive 0 onto disk in drive 1
CATALOG	CATALOG D0	Displays list of filenames in specified drive.
COLLECT	COLLECT D1	Purges disk in specified drive of any improperly closed files (indicated by * beside file type).
CONCAT	CONCAT " NAME1 " TO " NAME2 ", D1	Concatenates file " NAME1 " to file " NAME2 ". I.e. NAME2 = NAME2 + NAME1
COPY	COPY " NAME ", D0 TO " NAME ", D1 COPY " NAME ", D0 TO " DUP ", D0 COPY D0 TO D1	Copies file " NAME " from drive 0 to drive 1 Makes duplicate of file " NAME " Copies entire contents from D0 to D1
DCLOSE	DCLOSE #n	Closes disk logical file 'n'
DIRECTORY	DIRECTORY D0	Exact same as Catalog. Use preference.
DLOAD	DLOAD " NAME ", Dd,Uu	Loads program " NAME " from drive 'd' on unit 'u'
DOPEN	DOPEN#n, " NAME ", Dd,Uu DOPEN#n, " NAME ", Dd,Uu,W	Opens file " NAME " for reading from drive 'd', unit 'u'. Default values: d=0, u=8. Data is retrieved through file number 'n'. Opens file " NAME " for writing to drive 'd', unit 'u'. Not necessary for RELative files.
DSAVE	DSAVE " NAME ", Dd,Uu	Saves current program to drive 'd' on unit 'u' as file " NAME "
HEADER	HEADER " DISKNAME ", Dd,lid,Uu	Formats disk in drive 'd' unit 'u' assigning it a " DISKNAME " and 'id'.
RECORD	10 RECORD#n, a	Positions relative file open on logical file number 'n' to record number 'a'. 'a' may be dynamic but must be enclosed in brackets.
RENAME	RENAME " NAME " TO " NEWNAME ", D0	Changes a file name.
SCRATCH	SCRATCH " NAME ", D1	Eliminates file " NAME " from disk.



# BASIC RAM Memory Allocation

BASIC Text	Variable Table	Arrays Space	Empty Space	String Space	
0 0 0					
↑	↑	↑	↑	↑	↑
Start of BASIC	Start of Variables	Start of Arrays	End of Arrays	Bottom of Strings	Top of Memory
BASIC 4/2: \$28,29	\$2A,2B	\$2C,2D	\$2E,2F	\$30,31	\$34,35
VIC/C64: \$2B,2C	\$2D,2E	\$2F,30	\$31,32	\$33,34	\$37,38
B Series: \$2D,2E	\$31,32	\$35,36	\$37,38	\$3B,3C	\$0380,0381
+ 4/C16: \$2B,2C	\$2D,2E	\$2F,30	\$31,32	\$33,34	\$37,38

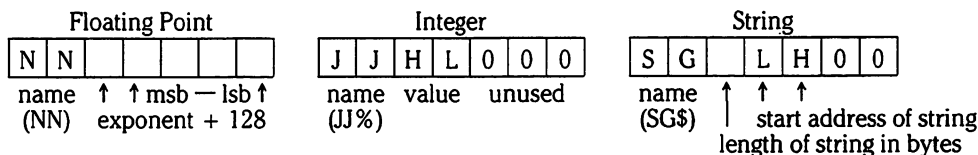
## BASIC Text Line Structure



### 'FOR' Stack Entry

LO	Pointer to first statement in loop
HI	Line number of first statement in loop
M4	
M3	
M2	'TO' value
M1	
EXP	Sign of 'STEP'
M4	
M3	
M2	'STEP' value
M1	
EXP	
HI	Pointer to 'FOR' variable
LO	'FOR' Token (LAST ON)
\$81	

## Variable Formats



### 'GOSUB' Stack Entry

HI	Pointer to 'GOSUB' statement
LO	Line Number of 'GOSUB' statement
HI	'GOSUB' Token (LAST ON)
LO	
\$8D	

## DS & DS\$ - Disk Status Variables

DS	Error Description
0	OK, no error exists
1	files scratched response (not an error)
2-19	Unused: can occur, should be ignored
20	read error; block header not found
21	read error; sync character not found
22	read error; data block not present
23	read error; checksum error in data
24	read error; byte decoding error
25	write error; write verify error
26	write protect on
27	read error; checksum error in header
28	write error; data extends into next block
29	disk id mismatch
30	syntax error; general syntax
31	syntax error; invalid command
32	syntax error; command line > 58 chars
33	syntax error; invalid filename
34	syntax error; no filename given
39	syntax error; command file not given
50	record not present
51	overflow in record
52	file too large
60	file open for write
61	file not open
62	file not found
63	file exists
64	file type mismatch
65	no block; t,s is next available block
66	illegal track or sector
67	illegal system track or sector
70	no channels (available)
71	dir error (directory error)
72	disk full or directory full
73	cbm dos v2 (or v2.x for later dos's); power up message, also indicates write attempt with dos mismatch
74	drive not ready
75	format speed error
76	controller error

## Reserved System Variables

### ST - The Status Variable

Bit	Val	Cassette Read	IEEE/Serial	Tape Load/Ver.	Vic/64 RS-232
0-7	0	OK	OK	OK	OK
0	1		time out on write		parity error
1	2		time out on read		framing error
2	4	short block		short block	rec. buffer overrun
3	8	long block		long block	unused
4	16	unrecoverable read error		any mismatch	CTS signal missing
5	32	checksum error		checksum error	unused
6	64	end of file	EOI		DSR signal missing
7	-128	end of tape	device not present	end of tape	break detected

## Additional B Series Commands

Function	Example	Purpose
BANK	BANK b	Sets bank number to 'b'.
BLOAD	BLOAD "NAME" ,Dd,Uu,ON Bb,Pp	Loads file "NAME" from drive 'd' unit 'u' into bank 'b' at position 'p'
BSAVE	BSAVE "NAME" ON Bb,Pp1 to Pp2	Saves current memory in bank 'b' from address 'p1' to 'p2' as file "NAME" to drive 0 unit 8. Addresses are in decimal.
DCLEAR	DCLEAR D1	Initialize disk in drive 1
DELETE	DELETE 10-30	Deletes lines from current program. Specify line range same as LIST.
DISPOSE	DISPOSE GOSUB	Purges stack of unwanted return addresses (like 'POP')
ELSE	IF ST THEN E = 1 ELSE E = 0	Alternate condition following IF..THEN. May also be used to transfer execution
INSTR	PRINT INSTR (A\$, B\$)	Returns position of string B\$ within A\$. Returns 0 if not found.
KEY	KEY KEYn, "CATALOG D0" + CHR\$(13)	Displays list of function key definitions Defines function key 'n'.
PUDEF	PUDEF "-.,\$"	Re-defines Print Using format characters. Default is "-.,\$". In this example, space is changed to '.', comma to period, period to comma, and dollars to pounds.
RESUME	RESUME RESUME n RESUME NEXT	Continues execution after program error or editing Resumes execution at line 'n' Resumes execution at start of current active FOR/NEXT
TRAP	TRAP 50000	Specifies routine at line 50000 as an ON ERROR routine.
USING	PRINT USING "-###,###";X	Specifies format to be used for numerical output.

## Additional + 4, C16 Commands

Function	Example	Purpose
AUTO	AUTO 100, 10	<b>Editing:</b> Supply line numbers starting with 100 in increments of 10
DELETE	DELETE -10	Delete BASIC lines up to line 10. Parameters work like LIST.
HELP	HELP	Hi-lites BASIC execution error in RVS field
KEY	KEY KEY FK, FK\$	Display Function Key assignments Define Function Key FK (1-8) as FK\$. Allows any string expression.
RENUMBER	RENUMBER 1000, 10, 500	Renumber BASIC text starting with line 1000 in increments of 10, from line 500 on.
TROFF	TROFF	Turns BASIC execution trace feature OFF.
TRON	TRON	Turns BASIC execution trace feature ON.
DO LOOP		<b>Structure:</b> can be followed by WHILE or UNTIL
EL	PRINT EL	Reserved variable: Error Line
ER	PRINT ER	Reserved variable: Error Number
ERR\$	PRINT ERR\$(ER)	Reserved variable: Error Message (example would print last error string)
GETKEY	10 GETKEY A\$	Instead of 10 GET A\$: IF A\$ = " " THEN 10
IF THEN ELSE	,1000 IF J = K THEN 1010 ELSE STOP	Must all be on same line.
INSTR	INSTR A\$, B\$, PO	Insert A\$ into B\$ at position PO.
PRINT USING	PRINT USING F\$, A\$	Print A\$ using format F\$
PUDEF	PUDEF "-.,\$"	Re-Define USING format characters
RESUME	RESUME 1200	Resume loop at 1200
TRAP	5 TRAP 1000	Equivalent to ON ERROR GOTO 1000
EXIT	2090 EXIT	Terminate loops started with DO
FLASH	100 FLASH A\$	<b>Graphics</b> Sets flashing attribute on string A\$
BOX	BOX CS, X1, Y1, X2, Y2, AN, 1	Draws a box from X1,Y1 to X2,Y2, at an angle AN, filled in with same colour as colour source CS
CHAR	210 CHAR CS, X, Y, A\$, 1	Will print A\$ at X,Y position on the Hi-Res screen, using colour source CS, reversed.
CIRCLE	CIRCLE 2, X, Y, XR, YR, S, E, A, I	Draws a circle where: 2 = Use Multicolor 1      S = Starting Arc (default 0 degrees) X,Y = Position of center                      E = Ending Arc (default 360 degrees) XR = X Radius                                      A = Clockwise rotation (default 0) YR = Y Radius                                      I = Increment or Coarseness (default 2)
COLOR	COLOR BK, FG, M1, M2, BD	Set colours for Background, Foreground, Multi-Colour 1, Multi-Colour 2, Border (range 0-15).
DRAW	230 DRAW 4,X1,Y1,X2,Y2,C	Will draw a line from X1,Y1 to X2,Y2 in Border colour

## Additional +4, C16 Commands, cont'd

Function	Example	Purpose
GRAPHIC	GRAPHIC M, C	Specify screen mode M. 0 = Text 1 = Multi-Colour Graphic 2 = Hi-Res Graphic 3 = Split-Screen (Text on bottom 3 lines) C <> 0 clears screen.
GRAPHIC CLR	GRAPHIC CLR	Clear current GRAPHIC screen
GSHAPE	250 GSHAPE S\$, X1, Y1, M	Gets a shape from S\$ and print it on the Hi-Res screen at X1,Y1 using mode M. 0 = Draw Shape as is (default) 1 = Draw Shape inverted 2 = Draw Shape OR'd with Screen 3 = Draw Shape AND'd with Screen 4 = Draw Shape XOR'd with Screen
JOY	PRINT JOY(JS)	Returns direction (0-8) of Joystick 1 or 2 (0-1). Fire Button adds 128 to direction value.
LOCATE	220 LOCATE X1, Y1	Set initial co-ordinates for plotting type commands to X1,Y1
PAINT	PAINT C, X, Y, M	Fills the area surrounding X,Y in colour C using mode M. 0 = Bordered by same colour as C 1 = Bordered by any foreground colour
RCLR	PRINT RCLR (CS)	Returns Colour Source information for: 0 = Background colour number 1 = Foreground colour number 2 = Multi-Colour 1 colour number 3 = Multi-Colour 2 colour number 4 = Border colour number
RDOT	PRINT RDOT (M)	Returns information for the next pixel to be plotted using mode M. 0 = X co-ordinate 1 = Y co-ordinate 2 = Colour Source
RGR	PRINT RGR (0)	Returns current GRAPHIC mode (0-3)
RLUM	PRINT RLUM (CS)	Returns luminance for colour source CS.
SCALE	200 SCALE X	Set scale to: 0 = Standard co-ordinates based on GRAPHIC mode. 1 = 0-1023 co-ordinate system.
SCNCLR	200 SCNCLR	Clears screen in any GRAPHIC mode
SOUND	260 SOUND	Single voice. followed by parameters for note, tone, etc.
SSHAPE	250 SSHAPE SS, X1, Y1, X2, Y2	Saves a shape into S\$ from X2,Y2 to X1,Y1 (the diagonally opposite corner)
VOL	270 VOL V	Sets volume from 0 to 8 maximum
<b>Machine Language:</b>		
DEC	DEC "FFFF"	Converts the string FFFF to decimal. Variable can also be used.
HEX\$	HEX\$(1024)	Converts the number 1024 to a string representing the hexadecimal equivalent. DEC and HEX\$ complement much like ASC and CHR\$
MONITOR	MONITOR	Enters Machine Language Monitor
F	F EA 6000 7000	Fill memory from ADDR1 to ADDR2 with specified hex value
H	H 6000, 7000, A9 FF	Hunt memory from ADDR1 to ADDR2 for the sequence A9 FF
A	A JSR \$FFD2	Assemble. works like Supermon assembler
D	D 6000	Disassemble from \$6000 on.
M	M 6000 6050	Memory dump displays memory contents in hex and screen POKE characters.
G	G 6000	Go to \$6000 and execute machine language there.
X	X	Exit MLM
S	S "program", 08, 6000, 7000	Save ML program between \$6000 and \$7000 on device 8
L	L "program"	Load specified program. Load address is contained in file.
R	R	Display registers

## B Series / +4 / C16 ESCAPE Key Functions

ESCAPE +	Function	ESCAPE +	Function
A	Automatic Insert Mode	N	Set Normal Screen display size
B	Set Bottom of Screen Window	O	Cancel Insert, Quote, and Reverse Modes
C	Cancel Automatic Insert Mode	P	Erase Begin
D	Delete line	Q	Erase End
E	Use Nonflashing Cursor (B Series only)	R	Set Reduced Screen display size
F	Use Flashing Cursor (B Series only)	S	Use Solid Cursor (B Series only)
G	Enable Bell	T	Set Top of Screen Window
H	Disable Bell	U	Use Underscore Cursor (B Series only)
I	Insert a line	V	Scroll Up
J	Move Cursor to Start of Current line	W	Scroll Down
K	Move Cursor to End of Current line	X	Cancel ESCAPE
L	Enable Scrolling	Y	Use Normal Character Set (B Series only)
M	Disable Scrolling	Z	Use Alternate Character Set (B Series only)

# Error Messages

Message	Description
BAD DATA	String data was received from an open file, but the program was expecting numeric data.
BAD SUBSCRIPT	The program was trying to reference an element of an array whose number is outside of the range specified in the DIM statement.
CAN'T CONTINUE	The CONT command will not work, either because the program was never 'RUN', there has been an error, or a line has been edited.
DEVICE NOT PRESENT	The required I/O device was not available for an 'OPEN', 'CLOSE', 'CMD', 'PRINT#', 'INPUT#', or 'GET#'.
DIVISION BY ZERO	Division by zero is a mathematical oddity and not allowed.
EXTRA IGNORED	Too many items of data were typed in response to an input statement. Only the first few items were accepted.
FILE NOT FOUND	If you were looking for a file on tape, an 'end-of-tape' marker was found. If you were looking on a disk, no file with that name exists.
FILE NOT OPEN	The file specified in a 'CLOSE', 'CMD', 'PRINT#', 'INPUT#', or 'GET#', must first be 'OPEN'ed.
FILE OPEN	An attempt was made to OPEN a file using the number of an already open file.
FORMULA TOO COMPLEX	The string expression being evaluated should be split into at least two parts for the system to work with, or a formula has too many parentheses.
ILLEGAL DIRECT	The 'INPUT' statement can only be used within a program, and not in direct mode.
ILLEGAL QUANTITY	A number used as the argument of a function or statement is out of the allowable range.
LOAD	A problem has occurred during program LOAD, disk or tape
NEXT WITHOUT FOR	This is caused by either incorrectly nesting loops or having a variable name in a 'NEXT' statement that doesn't correspond with one in a 'FOR' statement.
NOT INPUT FILE	An attempt was made to 'INPUT' or 'GET' data from a file which was specified to be for output only.
NOT OUTPUT FILE	An attempt was made to 'PRINT' data to a file which was specified as input only.
OUT OF DATA	A 'READ' statement was executed but there is no data left unread in a 'DATA' statement.
OUT OF MEMORY	There is no more 'ram' available for program or variables. This may also occur when too many 'FOR' loops have been nested, or when there are too many 'GOSUB's in effect.
OVERFLOW	The result of a computation is larger than the largest number allowed, which is 1.70141884e +38.
REDIM'D ARRAY	An array may only be 'DIM' ensioned once. If an array variable is used before that array is 'DIM' d, an automatic 'DIM' operation is performed on that array setting the number of elements to ten, and any subsequent 'DIM's will cause this error.
REDO FROM START	Character data was typed in during an 'INPUT' statement when numeric data was expected. Just re-type the entry so that it is correct, and the program will continue by itself.
RETURN WITHOUT GOSUB	A 'RETURN' statement was encountered, and no 'GOSUB' command has been issued.
STRING TOO LONG	(except 2.0) Maximum string length is 255 characters. This error will also occur if INPUT# receives more than 80 characters without a carriage return (ie. BASIC input buffer is 80 bytes long), or if a disk filename is longer than 16 characters.
SYNTAX	A statement or command is unrecognizable. A missing or extra parenthesis, misspelled keywords, etc.
TYPE MISMATCH	This error occurs when a number is used in place of a string, or vice-versa.
UNDEF'D FUNCTION	A user defined function was referenced, but it has never been defined using the 'DEF FN' statement.
UNDEF'D STATEMENT	An attempt was made to 'GOTO' or 'GOSUB' or 'RUN' a line number that doesn't exist.
VERIFY	The program on tape or disk does not match the program currently in memory.

## Notes

# B Series, + 4, and C16 Error Messages

This list is a summary of error messages that are displayed by PRINTing ERR\$(X) where X equals the value down the left column.

X	Message	Explanation
0	?STOP KEY DETECTED	Occurs when doing a KERNAL I/O function and the STOP key is pressed. May occur during LOAD or SAVE (or OPEN, CLOSE, GET#, INPUT#, PRINT# when the cassette tape is moving). CLOSE any open write files to save data.
1	?TOO MANY FILES	Maximum OPEN files is ten.
2	?FILE OPEN	An attempt was made to OPEN or DOPEN a file with a file number already in use.
3	?FILE NOT OPEN	An attempt was made to access a file not previously OPEN or DOPENed
4	?FILE NOT FOUND	The file specified in OPEN or LOAD was not found on the device specified. For tape I/O, an end of tape marker was encountered.
5	?DEVICE NOT PRESENT	An attempt was made to access a device not currently connected or powered-up on the IEEE-488 bus. May happen on OPEN, CLOSE, CMD, INPUT#, GET#, PRINT#. If filename is not specified with OPEN, this error will occur.
6	?NOT INPUT FILE	An attempts was made to read a file originally OPENed for writing.
7	?NOT OUTPUT FILE	An attempts was made to write data to a file originally OPENed for reading. The keyboard cannot be written to.
8	?MISSING FILENAME	All LOADs and SAVEs from the IEEE port (eg. disk) require a filename.
9	?ILLEGAL DEVICE NUMBER	Occurs if you try to access a device in an illegal manner. For example, LOADing or SAVEing from/to the keyboard, screen, or RS-232.
10	?ARE YOU SURE	Confirmation prompt for BACKUP, SCRATCH, and HEADER. It is not an error message and occurs only in direct mode, not during BASIC program execution.
11	?BAD DISK	Media failure on HEADER command.
12	<return> READY. <return>	This Is Not An Error Message. This message lets you know that your system is ready to use.
13	<space> IN <space>	Not An Error Message. Used to indicate which line an error has occurred "in".
14	?BREAK	This occurs when the STOP key is pressed during BASIC execution. CONT can be used to restart the program.
15	?EXTRA IGNORED	Too many items of data or separators were entered in response to an INPUT statement.
16	?REDO FROM START	This diagnostic message occurs when a numeric variable is used with INPUT and non-numeric data is received. INPUT continues to function until acceptable data has been received.
17	Last Evaluated Number	This Is Not An Error Message. This is the last value that has been processed through the numerical output buffer. (eg. print 100/10 : print ERR\$(17) ...will print 10 both times.
18	"MORE" <return>	This Is Not An Error Message. Prints "MORE" and carriage return.
19	Power On Message	This Is Not An Error Message. Prints the same screen message that is displayed immediately after power-up
20	?NEXT WITHOUT FOR	Either a NEXT is improperly nested or the variable in a NEXT statement corresponds to no previously executed FOR statement.
21	?SYNTAX	BASIC cannot recognize the statement you have typed. Caused by such things as missing parenthesis, illegal characters, incorrect punctuation, misspelled keyword.
22	?RETURN WITHOUT GOSUB	A RETURN statement was encountered with noprevious GOSUB.
23	?OUT OF DATA	An attempt was made to READ data from a DATA statement but no data exists or the program has already read them all.
24	?ILLEGAL QUANTITY	Occurs when a function is accessed with a parameter out of range caused by: 1. A matrix subscript out of range (0 < X < 32767) 2. LOG (negative or zero argument) 3. SQR (negative argument) 4. A#B where A<0 and B not integer. 5. Call of USR before a machine language subroutine has been patched in. 6. Use of string functions MID\$, LEFT\$, RIGHT\$, with length parameters out of range. 7. Index on...GOTO out of range. 8. Addressof PEEK, POKE, WAIT or SYS out of range. 9. Byte parameters of WAIT, POKE, TAB and SPC out of range.
25	?OVERFLOW	Numbers resulting from computations or input that are greater than 1.70141184E + 38 or less than 2.93873587E-39.
26	?OUT OF MEMORY	BASIC text space, or Variables space, or Arrays memory space has been completely filled
27	?UNDEFINED STATEMENT	A GOTO, GOSUB, or THEN has been executed with a line number that does not exist.
28	?BAD SUBSCRIPT	An attempt was made to reference an array element which is outside the dimensions specified in the DIM statement.
29	?REDIM'D ARRAY	An attempt was made to define an array using a variable already used in an array.
30	?DIVISION BY ZERO	Illegal divide. Message is followed by the line number - list and check variables.
31	?ILLEGAL DIRECT	INPUT, INPUT#, GET, GET#, and DEF cannot be used in direct mode.
32	?TYPE MISMATCH	An arithmetic operation has been given non-numeric data, or a string operation has been numeric data.
33	?STRING TOO LONG	Maximum string length is 255 characters. This error will also occur if INPUT# receives more than 80 characters without a carriage return (ie. BASIC input buffer is 80 bytes long), or if a disk filename is longer than 16 characters.
34	?FILE DATA	Occurs when a numeric variable is used with INPUT# and non-numeric data is received.
35	?FORMULA TOO COMPLEX	BASIC has run out of temporary pointers to keep track of substrings in evaluating a string expression. Break the expression into two smaller parts to cure the problem.
37	?UNDEFINED FUNCTION	Reference was made to a user defined function which had never been defined with DEF.
38	?LOAD ERROR	Cassette tape only. To improve tape reliability, programs are recorded twice with SAVE. This error will occur if LOAD finds recording errors in corresponding positions of both recordings. If more than 31 errors are detected in the first pass, LOAD will not attempt to read the second.
39	?VERIFY ERROR	A VERIFY operation did not match the contents of file with the contents of memory. Re-SAVE your program on another disk or tape.
40	?OUT OF STACK	Too many open FOR...NEXT loops or too many GOSUB calls.
41	?UNABLE TO RESUME	Resume will not operate after a fatal error.
42	?UNABLE TO DISPOSE	All of the DISPOSE type items have been disposed of or none exist.
43	?OUT OF TEXT	A LOAD or DLOAD has attempted to bring in a file larger than 64K. This error will not occur when using the BLOAD command.

# BASIC Abbreviations

Command	Abbreviation	2.0	3.5	4.0	B	Command	Abbreviation	2.0	3.5	4.0	B	Command	Abbreviation	2.0	3.5	4.0	B
ABS	a SHIFT B	•	•	•	•	FRE	f SHIFT R	•	•	•	•	RDOT	r SHIFT D	•	•	•	•
APPEND	a SHIFT P	•	•	•	•	GET	g SHIFT E	•	•	•	•	READ	r SHIFT E	•	•	•	•
ASC	a SHIFT S	•	•	•	•	GETKEY	getk SHIFT E	•	•	•	•	RECORD	re SHIFT C	•	•	•	•
ATN	a SHIFT T	•	•	•	•	GET#	none	•	•	•	•	REM	none	•	•	•	•
AUTO	a SHIFT U	•	•	•	•	GOTO	g SHIFT O	•	•	•	•	RENAME	re SHIFT N	•	•	•	•
BACKUP	b SHIFT A	•	•	•	•	GOSUB	go SHIFT S	•	•	•	•	RENUMBER	ren SHIFT U	•	•	•	•
BANK	ba SHIFT N	•	•	•	•	GRAPHIC	g SHIFT R	•	•	•	•	RESTORE	re SHIFT S	•	•	•	•
BLOAD	b SHIFT L	•	•	•	•	GSHAPE	g SHIFT S	•	•	•	•	RESUME	res SHIFT U	•	•	•	•
BOX	b SHIFT O	•	•	•	•	HEX\$	h SHIFT E	•	•	•	•	RETURN	re SHIFT T	•	•	•	•
BSAVE	b SHIFT S	•	•	•	•	HEADER	h SHIFT E	•	•	•	•	RGR	r SHIFT G	•	•	•	•
CHR\$	c SHIFT H	•	•	•	•	IF	none	•	•	•	•	RIGHT\$	r SHIFT I	•	•	•	•
CHAR	ch SHIFT A	•	•	•	•	INPUT	none	•	•	•	•	RLUM	r SHIFT L	•	•	•	•
CIRCLE	c SHIFT I	•	•	•	•	INPUT#	i SHIFT N	•	•	•	•	RND	r SHIFT N	•	•	•	•
CLOSE	cl SHIFT O	•	•	•	•	INSTR	in SHIFT S	•	•	•	•	RUN	r SHIFT U	•	•	•	•
CLR	c SHIFT L	•	•	•	•	INSTR	in SHIFT S	•	•	•	•	SAVE	s SHIFT A	•	•	•	•
CMD	c SHIFT M	•	•	•	•	INT	none	•	•	•	•	SCNCLR	s SHIFT C	•	•	•	•
CONT	c SHIFT O	•	•	•	•	JOY	j SHIFT O	•	•	•	•	SCALE	sc SHIFT A	•	•	•	•
COLOR	co SHIFT L	•	•	•	•	KEY	k SHIFT E	•	•	•	•	SCRATCH	s SHIFT C	•	•	•	•
COLLECT	co SHIFT L	•	•	•	•	LET	l SHIFT E	•	•	•	•	sc SHIFT R	s SHIFT R	•	•	•	•
CONCAT	col SHIFT L	•	•	•	•	LEFT\$	le SHIFT F	•	•	•	•	SGN	s SHIFT G	•	•	•	•
COPY	co SHIFT N	•	•	•	•	LEN	none	•	•	•	•	SIN	s SHIFT I	•	•	•	•
COS	co SHIFT P	•	•	•	•	LIST	l SHIFT I	•	•	•	•	SOUND	s SHIFT O	•	•	•	•
COS	none	•	•	•	•	LOAD	l SHIFT O	•	•	•	•	SPC(	s SHIFT P	•	•	•	•
DATA	d SHIFT A	•	•	•	•	LOCATE	lo SHIFT C	•	•	•	•	SQR	s SHIFT Q	•	•	•	•
DCLOSE	d SHIFT C	•	•	•	•	LOG	none	•	•	•	•	SSHape	s SHIFT S	•	•	•	•
DCLEAR	dc SHIFT L	•	•	•	•	LOOP	lo SHIFT O	•	•	•	•	STOP	s SHIFT T	•	•	•	•
DEC	none	•	•	•	•	MID\$	m SHIFT I	•	•	•	•	STR\$	st SHIFT R	•	•	•	•
DEFFN	d SHIFT E	•	•	•	•	MONITOR	m SHIFT O	•	•	•	•	SYS	s SHIFT Y	•	•	•	•
DELETE	de SHIFT L	•	•	•	•	NEW	none	•	•	•	•	TAB(	t SHIFT A	•	•	•	•
DIM	d SHIFT I	•	•	•	•	NEXT	n SHIFT E	•	•	•	•	TAN	none	•	•	•	•
DIRECTORY	di SHIFT R	•	•	•	•	ON	none	•	•	•	•	TRAP	t SHIFT R	•	•	•	•
DISPOSE	di SHIFT S	•	•	•	•	OPEN	o SHIFT P	•	•	•	•	TRON	tr SHIFT O	•	•	•	•
DLOAD	d SHIFT L	•	•	•	•	PAINT	p SHIFT A	•	•	•	•	TROFF	tro SHIFT F	•	•	•	•
DO	none	•	•	•	•	PEEK	p SHIFT E	•	•	•	•	UNTIL	u SHIFT N	•	•	•	•
DOPEN	d SHIFT O	•	•	•	•	POKE	p SHIFT O	•	•	•	•	USR	u SHIFT S	•	•	•	•
DRAW	d SHIFT R	•	•	•	•	POS	none	•	•	•	•	VAL	none	•	•	•	•
DSAVE	d SHIFT S	•	•	•	•	PRINT	?	•	•	•	•	VERIFY	v SHIFT E	•	•	•	•
END	e SHIFT N	•	•	•	•	PRINT#	p SHIFT R	•	•	•	•	VOL	v SHIFT O	•	•	•	•
ERR\$	e SHIFT R	•	•	•	•	PRINT USING	?us SHIFT I	•	•	•	•	WAIT	w SHIFT A	•	•	•	•
EXP	e SHIFT X	•	•	•	•	PUDEF	p SHIFT U	•	•	•	•	WHILE	w SHIFT H	•	•	•	•
FOR	f SHIFT O	•	•	•	•	RCLR	r SHIFT C	•	•	•	•						

## C64 Super Expander Commands

Function	Example	Purpose
BOX	BOX 1, X1, Y1, X2, Y2, 45, 1	Draws a box in the foreground colour, from X1,Y1 to X2,Y2, at a 45 degree angle, filled in with same colour.
CHAR	210 CHAR CS, X, Y, A\$, 1	Will print A\$ at X,Y position on the Hi-Res screen, using colour source CS, reversed.
CIRCLE	CIRCLE 2, X, Y, XR, YR, S, E, A, I	Draws a circle where: 2 = Use Multicolor 1      S = Starting Arc (default 0 degrees) X,Y = Position of center      E = Ending Arc (default 360 degrees) XR = X Radius      A = Clockwise rotation (default 0) YR = Y Radius      I = Increment or Coarseness (default 2)
COLINT	COLINT 0, 1050	Process events at BASIC line 1050: 0 = Sprite to Sprite collisions 1 = Sprite to Bit Map display collisions 2 = Light Pen activity
COLOR	COLOR BK, FG, M1, M2, BD	Set colours for Background, Foreground, Multi-Colour 1, Multi-Colour 2, Border (range 0-15).
DRAW	230 DRAW 4,X1,Y1,X2,Y2,C	Will draw a line from X1,Y1 to X2,Y2 in Border colour
FILTER	230 FILTER CO, LP, BP, HP, R	Set filter parameters. CO = Cutoff frequency (0-2048) LP = Low Pass (1 = ON, 0 = OFF) BP = Band Pass (1 = ON, 0 = OFF) HP = High Pass (1 = ON, 0 = OFF) R = Resonance (0-15)
GRAPHIC	GRAPHIC M, C	Specify screen mode M. 0 = Text 1 = Multi-Colour Graphic 2 = Hi-Res Graphic 3 = Split-Screen (Text on bottom 3 lines) C <> 0 clears screen.
GSHAPE	250 GSHAPE S\$, X1, Y1, M	Gets a shape from S\$ and print it on the hi-res screen at X1,Y1 using mode M. 0 = Draw Shape as is (default) 1 = Draw Shape inverted 2 = Draw Shape OR'd with Screen 3 = Draw Shape AND'd with Screen 4 = Draw Shape XOR'd with Screen
KEY	KEY KEY FK, FK\$	Display Function Key assignments Define Function Key FK (1-8) as FK\$. Allows any string expression.

# C64 Super Expander Commands, cont'd

Function	Example	Purpose
LOCATE	220 LOCATE X1, Y1	Set initial co-ordinates for plotting type commands to X1,Y1
MOVSPR	240 MOVSPR N, X, Y	Move Sprite N to X, Y
PAINT	PAINT C, X, Y, M	Fills the area surrounding X,Y in colour C using mode M. 0 = Bordered by same colour as C 1 = Bordered by any foreground colour
RBUMP	PRINT RBUMP (E)	Returns collision information for: 0 = Sprite to Sprite 1 = Sprite to Background
RCLR	PRINT RCLR (CS)	Returns Colour Source information for: 0 = Background colour number 1 = Foreground colour number 2 = Multi-Colour 1 colour number 3 = Multi-Colour 2 colour number 4 = Border colour number
RDOT	PRINT RDOT (M)	Returns information for the next pixel to be plotted using mode M. 0 = X co-ordinate 1 = Y co-ordinate 2 = Colour Source
RGR	PRINT RGR(0)	Returns GRAPHIC mode (0-3).
RJOY	PRINT RJOY(JS)	Returns direction (0-8) of Joystick 1 or 2. Fire Button adds 128 to direction value.
RPEN	PRINT RPEN(L)	Returns Location of Lightpen. 0 = X co-ordinate 1 = Y co-ordinate
RPOT	PRINT RPOT(P)	Returns Position (0-255) of Paddle P. 0 = Paddle 1 1 = Paddle 2 2 = Paddle 3 3 = Paddle 4 Fire Button adds 256 to position value
RSPCOL	PRINT RSPCOL(C)	Returns Spritecolour information. 0 = Multi-Colour 1 number 1 = Multi-Colour 2 number
RSPPOS	PRINT RSPPOS(SP,C)	Returns information for Sprite SP (0-7). C = 0 X co-ordinate C = 1 Y co-ordinate
RSPR	PRINT RSPR(SP,F)	Returns information for Sprite SP (0-7). F = 0 Sprite ON or OFF (1 or 0) F = 1 Foreground colour (0-15) F = 2 Display Priority (0 = above, 1 = below) F = 3 X Expand (1 = ON) F = 4 Y Expand (1 = ON) F = 5 Display mode (0 = Hi-Res, 1 = Multicolour)
SCALE	200 SCALE X	Set scale to: 0 = Standard co-ordinates based on GRAPHIC mode. 1 = Super Expander co-ordinate system.
SCNCLR	200 SCNCLR	Clears screen in any GRAPHIC mode
SPRCOL	200 SPRCOL M1, M2	Set sprite Multicolours 1 and 2 (0-15)
SPRDEF	SPRDEF	Enter Sprite Designer Function. Key detected are: 0-7 Destination Sprite (prompted) A Automatic Cursor movement toggle CRSR keys Moves Cursor RETURN Move to start of next line RETURN Exit Sprite Designer (prompted) HOME Move to Home position CLR Erase grid 1-4 Selects Colour Source CTRL 1-8 Sprite Foreground Colour (0-7) Commodore 1-8 Sprite Foreground Colour (8-15) STOP Cancel changes Shift RETURN Save Sprite X X Expand Y Y Expand M Multi-Colour/Hi-Res toggle
SPRITE	200 SPRITE SP, EN, FG, PR, XE, YE, M	Set Sprite parameters. SP = Sprite number (0-7) EN = Enable (1 = ON) FG = Sprite Foreground colour (0-15) PR = Priority (0 = above, 1 = below) XE = X Expand (1 = ON) YE = Y Expand (1 = ON) M = Mode (0 = Hi-Res, 1 = Multi-Colour)
SPRSAV	200 SPRSAV SP, SP\$	Save Sprite SP into SP\$
SSHAPE	250 SSHAPE S\$, X1, Y1, X2, Y2	Saves a shape into S\$ from X2,Y2 to X1,Y1 (the diagonally opposite corner)
TEMPO	200 TEMPO T	Sets Tempo T = 0-255 (default 8)
TUNE	200 TUNE EV, AT, DC, SU, RL, WV, WT	Sounds note using: EV = Envelope number (0-9) AT = Attack rate (0-15) DC = Decay rate (0-15) SU = Sustain volume (0-15) RL = Release rate (0-15) WV = Waveform 0 = Triangle 1 = Sawtooth 2 = Pulse 3 = Noise 4 = Ring Modulation WT = Pulse Width (with WV = 2 only)

# COMAL Commands

## COMAL Flags & Reserved Variables

<b>EOD</b>	EOD	End Of Data flag
<b>EOF</b>	EOF(<filename>)	End Of File flag
<b>ESC</b>	ESC	stop key pressed flag
	TRAP ESC<type>	
<b>FALSE</b>	FALSE	predefined value = 0
<b>STATUS\$</b>	STATUS\$	status of disk channel
<b>TRUE</b>	TRUE	predefined value of 1

Note 1: Commodore BASIC, with the exception of a few commands, is a subset of COMAL. COMAL has all but ASC, CLR, DEF FN, GOSUB & RETURN, POS, REM, USR, VERIFY, WAIT, and BASIC 4.0 Disk Commands are sent via the COMAL PASS Command; other I/O commands (DLOAD, DCLOSE, RECORD\*, etc) are much like BASIC 2.0 format.

Note 2: GOSUB (and ON...GOSUB) & RETURN are replaced by PROC Commands

Format: ( ) Numeric Brackets - numeric input required  
< > Angle Brackets - denotes user supplied input  
[ ] Square Brackets - indicates optional input

Thus: [[< >]] would specify the user supplied input must be of numeric nature, if the option is exercised.

## Commands Common to COMAL and CBM BASIC With NO Differences

<b>ABS</b>	gives the absolute value
<b>AND</b>	logical AND
<b>ATN</b>	arctangent in radians
<b>CHR\$</b>	gives that numbers character
<b>COS</b>	cosine in radians
<b>DATA</b>	provides data for a READ
<b>END</b>	halt program execution
<b>EXP</b>	natural log e to n
<b>INT</b>	gives nearest integer less than or equal
<b>LEN</b>	gives the length of string
<b>LET</b>	assign value to variable
<b>LOG</b>	natural logarithm of n
<b>NEW</b>	clears program from memory
<b>NOT</b>	logical NOT
<b>OR</b>	logical OR
<b>PEEK</b>	look at memory
<b>POKE</b>	change memory location
<b>RESTORE</b>	reuse DATA with READ
<b>RUN</b>	run program now in memory
<b>SGN</b>	-1 if neg, 0 if 0, 1 if pos
<b>SIN</b>	gives sine in radians
<b>SQR</b>	gives square root
<b>STOP</b>	halt program execution
<b>SYS</b>	transfer control to assembly language
<b>TAB</b>	print spaces up to specified column
<b>TAN</b>	gives tangent in radians
<b>THEN</b>	part of IF structure
<b>TO</b>	increment FOR variable start TO end

## SPECIAL INFO

Line numbers allowed: 1-9999.  
Identifiers up to 16 chars (unshifted alpha, digits, [ , ] , ' , <- , ^)  
Null input is accepted.  
First time into graphics: SETGRAPHIC 0  
After that simply: SETGRAPHIC  
RUN/STOP RESTORE keys restore default colors.  
To clean up the identifier name table:  
LIST "PROGRAM.L"  
(frees up memory. NEW  
removes unused identifiers) ENTER "PROGRAM.L"  
Save a program to disk: SAVE "PROGRAM"  
Load a program from disk: LOAD "PROGRAM"  
List a program to printer: SELECT "LP:"  
LIST

## COMAL 64 Colours List (COMAL 0.14/2.0)

Number	Colour	CHR\$	Number	Colour	CHR\$
0	BLACK	144	8	ORANGE	129
1	WHITE	5	9	BROWN	149
2	RED	28	10	LIGHT RED	150
3	CYAN	159	11	DARK GREY	151
4	PURPLE	156	12	MEDIUM GREY	152
5	GREEN	30	13	LIGHT GREEN	153
6	BLUE	31	14	LIGHT BLUE	154
7	YELLOW	158	15	LIGHT GREY	155

## COMAL Commands NOT Found in CBM BASIC (\* except BASIC 3.5)

<b>*AUTO</b>	AUTO [<start line>][,<increment>]	automatic line numbering
<b>BASIC</b>	BASIC	back into BASIC mode
<b>CASE</b>	CASE <control expression> [OF]	multiple choice decisions
<b>CHAIN</b>	CHAIN <filename>	load & run program on disk
<b>CLOSED</b>	PROC <procname>{(<params>)} [CLOSED] FUNC <funcname>{(<params>)} [CLOSED]	all proc or func variables local
<b>*DEL</b>	DEL <range>	deletes lines
<b>DIV</b>	<dividend> DIV <divisor>	division with integer answer
<b>*DO</b>	DO <statements>	do the following statements
<b>EDIT</b>	EDIT [<range>]	lists lines without indentations
<b>ELIF</b>	ELIF <expression> [THEN]	short for ELSE IF condition
<b>*ELSE</b>	ELSE	alternative statements in IF structure
<b>ENDCASE</b>	ENDCASE	end of CASE structure
<b>ENDFOR</b>	ENDFOR [<control variable>]	end of FOR structure
<b>ENDFUNC</b>	ENDFUNC [<function name>]	end of function
<b>ENDIF</b>	ENDIF	end of IF structure
<b>ENDPROC</b>	ENDPROC [<procedure name>]	end of procedure
<b>ENDWHILE</b>	ENDWHILE	end of WHILE structure
<b>ENTER</b>	ENTER <filename>	merge a program segment from disk
<b>EXEC</b>	[EXEC] <procname>{(<actual parameter list>)}	execute a procedure
<b>FUNC</b>	FUNC <name>{(<params>)} [EXTERNAL <filename>] FUNC <name>{(<params>)} [CLOSED]	start of a multiline function
<b>IN</b>	<string1> IN <string2>	locate position of string1 within string2
<b>KEYS</b>	KEY\$	scans keyboard (not in PET COMAL 0.14)
<b>LABEL</b>	<label name>:	assigns a label name to the line
<b>MOD</b>	<dividend> MOD <divisor>	gives remainder of division (modulo)
<b>NULL</b>	NULL	does nothing (no op)
<b>OF</b>	CASE <expression> [OF] DIM <stringvar> OF <max char> DIM <stringarray>(<array index>) OF <max char>	part of DIM or CASE structure
<b>OTHERWISE</b>	OTHERWISE	default for CASE
<b>PROC</b>	PROC <name>{(<params>)} [EXTERNAL <filename>] PROC <name>{(<params>)} [CLOSED]	start of multiline procedure
<b>RANDOM</b>	OPEN FILE <filename>,<filename>.RANDOM <recln>	random access disk file
<b>RANDOMIZE</b>	RANDOMIZE	generate new random numbers
<b>REF</b>	REF <var>	param var used in reference in proc
<b>*RENUM</b>	RENUM [<targetstart>][,<increment>]	renumber program
<b>REPEAT</b>	REPEAT	start of REPEAT structure
<b>*TRAP</b>	TRAP ESC<type>	disable stop key
<b>*UNTIL</b>	UNTIL <expression>	end of REPEAT loop
<b>*USING</b>	PRINT USING <format>: <var list> PRINT [FILE <filename>:] USING <format>:<vars>	allows formatted output (not PET 0.14) including FILE output.
<b>WHEN</b>	WHEN <list of values>	choice in CASE structure
<b>*WHILE</b>	WHILE <expression> [DO] [<statement>]	start of WHILE structure
<b>WRITE</b>	WRITE FILE <filename>{,<recnum>}: <var list> OPEN [FILE] <filename>,<filename>.WRITE	write to a file
<b>ZONE</b>	ZONE <tab interval> ZONE	tab increment

## Commands Common to COMAL and CBM BASIC With SLIGHT Differences

<b>//</b>	//[<anything>]	allows comments in a program
<b>APPEND</b>	OPEN [FILE] <filename>,<filename>.APPEND	start at end of seq file
<b>CAT</b>	CAT [<drive number>]	gives disk directory
<b>CLOSE</b>	CLOSE [FILE] [<filename>]	closes files
<b>CON</b>	CON	continue program execution
<b>DELETE</b>	DELETE <filename>	deletes a file from disk
<b>DIM</b>	DIM <string var> OF <max char> DIM <str array>(<array index>) OF <max char> DIM <array name>(<array index>)	reserves/allocates string & array space
<b>FILE</b>	INPUT FILE <filename>{,<recnum>}: <var list> PRINT FILE <filename>{,<recnum>}: <val list> READ FILE <filename>{,<recnum>}: <var list> WRITE FILE <filename>{,<recnum>}: <var list> OPEN [FILE] <filename>,<filename>{,<type>} CLOSE [FILE] [<filename>]	specifies that a file is to be used
<b>FOR</b>	FOR <var> = <start> TO <end> [STEP <step>] [DO]	start of FOR loop structure
<b>GOTO</b>	GOTO <label name>	go to line with this name
<b>IF</b>	IF <condition> [THEN] IF <condition> THEN <statement>	start of conditional IF structure
<b>INPUT</b>	INPUT [<prompt>:] <var list> INPUT FILE <filename>{,<recnum>}:<var list>	input from keyboard or file
<b>LIST</b>	LIST [<range>:] [<filename>]	list program
<b>LOAD</b>	LOAD <filename>	load a program from disk
<b>OPEN</b>	OPEN [FILE] <filename>,<filename>{,<type>}	open a file
<b>ORD</b>	ORD(<string expression>) (same as ASC in BASIC)	returns integer representing the char
<b>OUTPUT</b>	SELECT [OUTPUT] <type>	select output location Like CMD
<b>PASS</b>	PASS <disk command>	passes a string to disk command channel
<b>PRINT</b>	PRINT [FILE <filename>:] [<items>] PRINT [FILE <filename>:] USING <format>:<vars> (RANDOM file use: [FILE <filename>,<recnum>:]	prints items to screen/printer/file
<b>READ</b>	READ <var list> READ FILE <filename>{,<rec num>}: <var list> OPEN [FILE] <filename>,<filename>.READ	read data from DATA line or file
<b>RND</b>	RND(<num>) RND(<start num>:<end num>)	random number
<b>SAVE</b>	SAVE <filename>	record program on disk
<b>SELECT</b>	SELECT [OUTPUT] <type>	choose output location
<b>SIZE</b>	SIZE	reports on memory usage (free memory)
<b>STEP</b>	STEP <numeric expression>	increment FOR loop var by this amount
<b>UNIT</b>	OPEN FILE <*>,<nm>.UNIT <dev>{,<sec>}{,<typ>}	specify unit (device)







Table with multiple columns containing hex addresses (e.g., 028C, 028D, 028E), decimal values (e.g., 652, 653, 654), names (e.g., DELAY, SHLFLG, LSTSHF), descriptions (e.g., Repeat Delay Counter, Keyboard Shift Key/Ctrl Key/Commodore Key), and various code references (e.g., C7E7, C7E8, C7E9).

COMAL

The Complete Commodore Inner Space Anthology



# Printer Control Characters

CHR\$ values are sent to printer with Secondary Addr 0 or 1. Not all codes are implemented on all printers

CHR\$	Operation	CHR\$	Operation	CHR\$	Operation
1	Begin double-width (enhanced) character mode	14	Begin double-width character mode	19	Set top of page
129	End double-width character mode	15	End double-width character mode	147	Feed to top of next page
8	Begin dot-programmable graphic mode	16	Tab to position in next 2 characters	26	Repeat graphics data
10	Line Feed	17	Switch to upper/lower case character set	27	Move to specified dot position
13	'Carriage Return' (automatic Line Feed on CBM printers)	145	Switch to upper case/graphics character set	29	Skip to next format field
141	Carriage Return without Line Feed	18	Begin reverse character mode	160	Shifted Space is necessary for leading spaces
		146	End reverse character mode	254	Output Programmable Character

## Commodore Dot-Matrix Printer Format Characters

Format Char	Format Supplied	Data Supplied	Output Result	Description
9	99999.99 .99 99.99	3.14159 3.14159 23	3.14 .14 23.00	Specifies numeric field, leading zeros suppressed
z	zzzz.zz	3.14159	00003.14	Specifies numeric field, leading zeros printed
.				Decimal point. Used to align data
\$	\$\$\$\$.99	129.95	\$129.95	Specifies numeric field with a \$ sign printed preceding data
s	s999.99 \$\$\$\$.99	-273.6 129.95	-273.60 +\$129.95	Prints sign of value as first character in field
-	\$999.99- s999.99- s\$\$\$\$.99-	-129.95 -273.6 129.95	\$129.95- -273.60- +\$129.95	Prints trailing sign if negative
a	aaaaa aaa	String String	String Str	Specifies a left-justified alpha field
b				Space or blank. Use spaces to separate fields
r	r?aaaa 999	over 100	?over 100	Allows format-string characters to be printed

## Letter Quality Printer Command Summary

Commands are for the StarWriter F10 printer. Most letter-quality printers are similar. Note: ESC is escape, or chr\$(27).

Command Format	Description	Command Format	Description
chr\$(12)	Form Feed	ESC Pnn	Feed paper to line nn
chr\$(8)	Backspace	ESC A	Alternate Ribbon Colour
ESC Lnn	Line feed spacing	ESC B	Normal Ribbon Colour
ESC chr\$(10)	Backwards Line Feed	ESC U	Half Line Feed
ESC 9	Set Left Margin	ESC D	Half Backwards Line Feed
ESC Enn	Set horizontal spacing to nn/120	ESC I	Set Horizontal Tab at Current position
ESC 2	Clear all horizontal tabs	ESC Hnnn	Move Carriage nnn spaces horizontally
ESC 8	Clear one Horizontal tab at current position	ESC Vnnn	Line feed of nnn/48 inches
ESC (t1,t2,...ff	Sets horizontal tabs at t1, t2, etc.	ESC Fnn	Set number of lines per page
ESC )t1,t2,...ff	Clears horizontal tabs at t1, t2, etc.	ESC N	Ignore auto-spacing on next character
ESC Cnn	Move to Column nn		

## Greek Alphabet

Dot Matrix CHR\$ Values	Letter	Upper Case	Lower Case	Roman Equiv.	Common Unit
14 17 10 4 26 1	Alpha	A	a	A	Area, Angles, Coefficients
0 1 62 80 42 4	Beta	B	β	B	
0 64 54 9 54 64	Gamma	Γ	γ	G	Angles, Coefficients, Flux Density, Transistor Amplification Factor
0 22 41 41 6 0	Delta	Δ	δ	D	Specific Gravity, Conductivity, Micrograms
0 10 21 21 17 2	Epsilon	E	ε	E	Density, Variation
0 64 44 50 35 64	Zeta	Z	ζ	Z	Natural Logarithm Base (e <sup>2.1242657</sup> )
0 64 48 65 62 0	Eta	H	η	H	Coefficients, Coordinates, Impedance
0 62 73 73 62 0	Theta	Θ	θ	V	Efficiency, Hysteresis Coefficient
0 0 30 1 2 0	Iota	I	ι	I	Phase Angle, Temperature
17 14 4 8 30 17	Kappa	K	κ	K	Dielectric Constant, Susceptibility
65 66 52 12 2 1	Lambda	Λ	λ	L	
1 126 32 32 120 4	Mu	M	μ	M	Wavelength
0 16 12 3 4 24	Nu	N	ν	N	Amplification Factor, micro (10 <sup>-6</sup> ), Permeability
0 66 53 41 65 0	Xi	Ξ	ξ	Y	Reluctivity
0 6 9 17 18 12	Omicron	O	ο	O	3.1415926
0 9 30 16 30 33	Pi	Π	π	P	
0 62 73 72 48 0	Rho	P	ρ	R	Resistivity
6 9 9 14 8 8	Sigma	Σ	ς	S	Summation
99 85 73 65 65 99	Capital Sigma				Time Constant
0 8 16 30 17 16	Tau	T	τ	T	
8 6 1 1 18 12	Upsilon	Υ	υ	U	Angles, Magnetic Flux
48 73 14 24 40 48	Phi	Φ	φ	F	
34 36 24 22 33 65	Chi	Χ	χ	X	Dielectric Flux, Phase Difference
112 9 126 8 48 64	Psi	Ψ	ψ	W	
0 6 9 2 9 6	Omega	Ω	ω	Q	Ohms, Angular Velocity
25 38 64 64 38 25	Capital Omega				

# Wordprocessing Reference Guide

Function	Superscript Control = RVS Key	EasyScript 64 Control = F1 Key	PaperClip Control = PET/CBM:RVS, 64:CTRL	Speedscript 64 Control = CTRL Key	WordPro Control = RVS Key	WordPro 64 Control = CBM Key
Restart Exit to BASIC	Control CLR Control STOP	Control CLR Control STOP	Control X		Control Shift Q	Control Q
<b>TOGGLE MODES</b>	<b>Superscript</b>	<b>EasyScript 64</b>	<b>PaperClip</b>	<b>Speedscript 64</b>	<b>WordPro</b>	<b>WordPro 64</b>
Capitals Decimal Insert Sound LJNE Mode Forced Space Mode	ESC or Control Shift/C Control . Control I Control *	Control Shift/C F6 Control I Control *	↑  Shift Ctrl (64:CBM Key)	Control I	\ Control N Shift Control Control \	⌂  Control I  F1 Control -
<b>CURSOR POSITIONING</b>	<b>Superscript</b>	<b>EasyScript 64</b>	<b>PaperClip</b>	<b>Speedscript 64</b>	<b>WordPro</b>	<b>WordPro 64</b>
Scroll Right Scroll Left Scroll Down Rapid Scroll Down Scroll Up Rapid Scroll Up Up a Line Next Screen Previous Screen Next Word Previous Word Next Sentence Previous Sentence Next Paragraph Previous Paragraph Beginning of File Home Position End of Text Goto Line x Goto Maximum Line Number Pan Up Pan Down Pan Left Pan Right Stop Panning Speed Panning Highlight Panning Cursor Pause Panning	CRSR Right CRSR Left CRSR Down  CRSR Up  Control Space Control Shift/Space  CLR HOME Control G E or O Control G Control G 999 Control CRSR Up Control CRSR Down Control CRSR Left Control CRSR Right STOP Shift hold Space tap Space	CRSR Right CRSR Left CRSR Down  CRSR Up  Control Space Control Shift/Space  CLR HOME Control G E Control G Control G 999 Control CRSR Up Control CRSR Down Control CRSR Left Control CRSR Right STOP Shift tap Space	CRSR Right CRSR Left CRSR Down Control CRSR Down CRSR Up Control CRSR Up     HOME twice HOME Shift RUN/STOP	CRSR Right CRSR Left CRSR Down Control CRSR Down CRSR Up CRSR Up   F1 F2 F3 F4 F5 F6  Control Z	CRSR Right CRSR Left CRSR Down Control CRSR Down CRSR Up CRSR Up   HOME twice HOME  Control G	CRSR Right CRSR Left CRSR Down Control CRSR Down CRSR Up CRSR Up   HOME  Control G
<b>TEXT</b>	<b>Superscript</b>	<b>EasyScript 64</b>	<b>PaperClip</b>	<b>Speedscript 64</b>	<b>WordPro</b>	<b>WordPro 64</b>
Change Line Length Reformat Paragraph Delete Line Insert Line Insert Multiple Lines Delete Text Erase All Erase Remainder Erase Paragraph Erase Sentence Erase Word Erase Delete Buffer Retrieve Buffer Contents Set Range Transfer Range Copy Range Erase Range Append Characters Append Lines Switch Text Space Set Column Move Column Delete Column Erase Column Shift Column Insert Space Before Column Repeat Column Add Numbers in Column Sort Column Set Sort Delimiters Set Delimiter Column Add Row Using Delimiters Modify Hunt/Search & Replace Text Hunt or Find Local Hunt or Find Global Hunt C Display Old Search & Replace Search & Replace Local Search & Replace Global Set Phrase Move Phrase Kill Phrase Toggle Case Toggle Case in Phrase Transpose Characters Change Border Colour Change Background Colour Change Character Colour Copy Text to Status Line Copy NX Filename to Status Line Read Stored Filename Display Available Memory Automatic Optional Hyphen Forced Space Breakpoint (soft Space)	Control CLR  Control DEL Control INST  Control D Control E A Control E R Control E P Control E S  Control R Control T Control A         Control M Control H L Control H G Control H C  Control @ L Control @ G  Control U    Shift Control  Control - Shift Space	Control CLR  Control DEL Control INST  Control D Control E A Control E R Control E P Control E S  Control R Control X Control A    Control S Control H L Control H M  Control @ L Control @ M  Control U    Control - Shift Space	Control Shift L  Control - Control + Control I Control D Control E  Control R Control T Control C Control E  Control Shift C Control Shift M Control Shift D Control Shift E Control Shift S Control Shift I Control Shift R Control = Control Shift A Control Shift Q Control Shift W Control Shift H Control F Control F or H  Shift RUN/STOP Control @ Control @ Control P Control M Control K  Control Shift K  F2 F4 F6 RUN/STOP Shift RUN/STOP  Control : Shift Space	Control D Shift CLR  Control D P or E P Control D S or E S Control D W or E W Control K Control R       Control Shift H Control H  Control A  Control X Control B Control B Control L  Control =  Control - Shift Space	Control DEL Control INST  Control E A Control E R  Control D S Control D W  Control R Control T Control L Control E L Control V Control A Control X     Control M S or R Control H or F L Control F G  HOME  Control - Shift Space	Control R    Control E A Control E R Control D P Control D S Control D W  Control J Control T  Control V Control A Control X    Control H or F L  Control @ L Control @ G  Control V Control A Control X  HOME F5 Shift - Shift Space

TABS	Superscript	EasyScript 64	PaperClip	Speedscript 64	WordPro	WordPro 64
Set Decimal Point Set Decimal Tabs Set Horizontal Tab Clear Horizontal Tab Tab 5 Spaces Set Vertical Tab Clear Vertical Tab Set Graphic Tab Goto Next Horizontal Tab Goto Next Vertical Tab Display Horizontal Tab positions Clear All Tabs Clear All Horizontal Tabs Clear All Vertical Tabs	Control . Control S H Control C H  Control S V Control C V  TAB (or Shift >) Shift TAB (or Shift <) Control P  Control K H Control K V	Control . Control T H Control C H  Control T V Control C V  F7 F8 Control P  Control Z H Control Z V	Control . Control N Shift CLR Shift CLR   TAB or RUN/STOP  Control CLR	RUN/STOP         Control † Control 4 \$0, \$1 RETURN i0 or i1	Control N Control S Control C  TAB or ←  Control K	Control S Control C  Control ₤ ←  Control K
FILES	Superscript	EasyScript 64	PaperClip	Speedscript 64	WordPro	WordPro 64
Enter FILE Mode Insert or Merge Files Load PRG Text File Load SEQ Text File Load Printer Interface File Save PRG Text File Save SEQ Text File Verify Data File Save Range Read Screen from Cursor Copy Global/Linked Files Scan loaded Directory names Disk Command Mode Display Directory Load Directory to Text Display Disk Status Initialize Drive(s)	Set Insert Mode, Load Control L Control L Control  Control F Control F Control  Control Shift F  Control Q Shift Control Control > \$0 or \$1 +\$0 or +\$1 RETURN i0 or i1	Set Insert Mode, Load Control L Control L Control  Control F Control F Control  Control Shift F  Control Q  F4 \$0 or \$1 +\$0 or +\$1 RETURN i0 or i1	Control A Control L Control J Control W Control S Control Z Control U Control Q  Control G  Control > Control 0, 1, 2  Control < i0 or i1	F7  F8  Control V  Control † Control 4 \$0, \$1 RETURN i0 or i1	Shift CLR Shift CLR I Shift CLR R  Shift CLR M  Shift CLR M R  Control *  Control . or >  Control 0, 1, 2 Control , RUN/STOP 0 or 1	F7 or CLR (F1 cancels) F7 or Shift CLR I F7 or Shift CLR R F8 Control P F7 or Shift CLR M  F7 or Shift CLR M R F3 Control *  Control . F3  Control , i0 or i1
All other disk commands are entered in CBM DOS Command Channel format (ie c = Copy, d = Duplicate, n = New, r = Rename, s = Scratch, v = Validate).						
FILL FILES	Superscript	EasyScript 64	PaperClip	Speedscript 64	WordPro	WordPro 64
Set Fill File Name Variable Block Variable Block Separator Measured Variable Block Fill Next Variable Block Fill Blocks from Cursor on Fill All Variable Blocks Clear Variable Blocks Find Next Variable Block Reset Data Pointer Close Fill File	Control B   Control V  Control Shift V Control TAB or Shift >  Control HOME	Control B  Ctrl B, CRSR Left, Ctrl M  Control V  Control Shift V Control F7  Control HOME	Control Shift Z Control B  Control Shift B  Control Shift V Control Shift N Control Shift F		Control B Control Z  Control TAB  Control I Control † Control TAB Control HOME	Control B Control Z  Control M  F4 (1st set) or F6 F2
OUTPUT FORMAT	Superscript	EasyScript 64	PaperClip	Speedscript 64	WordPro	WordPro 64
Format Command Indicator Format Command Separator Text following Format Commands Justification On, Off Centering On, Off Right Alignment On, Off Linefeeds On, Off Left Margin Add to Left Margin Subtract from Left Margin Right Margin Edge Right Add to Right Margin Subtract from Right Margin Release Left Margin Left Release Left Margin Right Auto Indent Paragraphs Right Auto Indent Paragraphs Left Offset from Column 1 on Printer Double Column Width Total Lines per Page (Paper Length) Text Lines per Page (Text Length) Line Spacing Vertical Positioning Bottom Margin Advance Lines Pause Output Force Paging Force Paging within N Lines * of List Data Fields Next Linked File Non-Specific Global File Link External File Link Open Table of Contents File Add to Table of Contents File Lines per Inch (form advance) Characters per Inch (pitch) Comment Heading Alternate Heading Footing Alternate Footing Set Page Number Output Page Number Heading/Footing Left Margin Heading/Footing Right Margin Unlock Header Margins Lock Header Margins Printer Command Send True ASCII Define Character as ASCII Value	Control / (✓) : ; ✓ju1, ju0 ✓cn1, cn0 ✓ra1, ra0 ✓lf1, lf0 ✓lm ✓rm ✓ma ✓of ✓pp ✓pg ✓sp ✓vp ✓ln ✓ps ✓fp0 ✓fpN ✓nx:filename ✓fa ✓pt ✓cm ✓hdxx:text.. ✓ftxx:text.. ✓p" Control " (in hd/ft) ✓hl ✓hr  Control 0-9 ✓1-9 = N	F3 ( ) : ; ✓ju1, ju0 ✓cn1, cn0 ✓ra1, ra0 ✓lm ✓rm ✓ma ✓of ✓pl ✓il ✓sp ✓vp ✓ln ✓ps ✓fp0 ✓fpN ✓nx:filename ✓lp ✓pt ✓nb ✓hdxx:text.. ✓ftxx:text.. ✓p" Control " (in hd/ft) ✓hl ✓hr  Control 0-9 ✓1-9 = N	Control \ or ₤ (✓) : ; ✓ju1, ju0 ✓cn1, cn0 ✓ra1, ra0 ✓lm ✓lm+ ✓lm- ✓rm ✓rm+ ✓rm- ✓ma- ✓ma+ ✓ai+ ✓ai- ✓pp ✓pg ✓sp ✓vp ✓ln ✓ps ✓fp ✓fpN ✓nx:filename ✓lk ✓ex: ✓tf:filename ✓tb: ✓ls ✓pt ✓cm ✓hdxx:text.. ✓ftxx:text.. ✓p" Control " (in hd/ft) ✓hl ✓hr ✓ml0 ✓ml1  Control : 1-9 ✓1-9 = N	Control ₤  c  l  r e  t b  w  h  f  Control ₤ "	Control / (✓) : ; ✓ju1, ju0 ✓cn1, cn0 ✓ra1, ra0 ✓lf1, lf0 ✓lm ✓rm ✓ma  t b  w  h  f  Control ₤ "	Control / (✓) : ; ✓ju1, ju0 ✓cn1, cn0 ✓ra1, ra0 ✓lf1, lf0 ✓lm ✓rm ✓ma- ✓ma+  mo dc (1-160) pp pg sp vp ln ps fp fpN ld nx:filename fa pt cm: hdxx:text.. ftxx:text.. p" Shift ₤ hl hr  pc Control 0-9 0-9 = N

OUTPUT	Superscript	EasyScript 64	PaperClip	Speedscript 64	WordPro	WordPro 64		
Select Default Output			Control Shift O					
Set Disk Device Number			Control \$					
Set Printer Device Number			Control #					
Select Output Options	Control O +	Control O +	Control O +	Control P	Control O +	F5 or Control O +		
Continuous Print	C	C			C	C N		
Non-Continuous (sheets)								
Device Number	D				S	L		
Fill File to be used	F	F			L	G		
Fill Using List Data					G			
Linked or Global File	G	L				L		
Global Restart						G		
Map Mode						R		
Odd Mode (odd # pages)						M		
Even Mode (even # pages)						O		
Number of Copies	X	X			X	E		
Output to Printer	P	P			Default	X		
Output to Video	V	V	Control V		V	V		
Output to SEQ file	S	S			D			
Speed up Video Output	Hold down Shift	Hold down Shift						
Pause Video Output	Tap space	Tap space						
Stop Output	STOP	STOP						
Continue Output	C	C						
Toggle Video/Printer Output	V/P	V/P						
Toggle Continuous/Non-Continuous	Shift P	Shift P						
Toggle Map/Video Mode						A		
<b>BACKGROUND PRINTING</b>								
Start Background Printing	Control X				Control P (file "dp")			
Resume after Page Break	Z (X for non-8032)							
Stop Background Printing	Control Shift X				Control P			
<b>PRINTER CONTROL CHARACTERS</b>								
Letter Quality	MX80	CBM	Superscript	EasyScript 64	PaperClip	Speedscript 64	WordPro	WordPro 64
Underline ON	Enhance ON	Enhance ON	Control [	Control [	Control [	Control ⌘ U	Control [	Control [
Underline OFF	Enhance OFF	Enhance OFF	Control ]	Control ]	Control ]	Control ⌘ U	Control ]	Control ]
Bold ON	Emphasise ON	Reverse ON	Control (	Control (	Control (		Control 8	Control (
Bold OFF	Emphasise OFF	Reverse OFF	Control )	Control )	Control )		Control 9	Control )
Shadow ON	Double print ON	n/a	Control &					
Shadow OFF	Double print OFF	n/a	Control *					
Print Red	Condense ON	n/a	Control !	Control Shift (				
Print Black	Condense OFF	n/a	Control "	Control Shift )				
Single Superscript	n/a	n/a	Control '		Control 4		Control 4	Control 6
Superscript Begin	n/a	n/a			Control 7			
Superscript End	n/a	n/a			Control 8			
Single Subscript	n/a	n/a	Control ,		Control 6		Control 6	Control 4
Subscript Begin	n/a	n/a			Control 9			
Subscript End	n/a	n/a			Control /			
Bold ON	n/a	n/a	Control ;					
Bold OFF	n/a	n/a	Control :					
Special Character	Special Character	Special Character			Control ;			

## Spreadsheet Commands

Commands shown are for the CalResult spreadsheet, but most spreadsheet programs use similar syntax.

System Commands:	Description
<b>B</b>	Blank: Cancel Contents of Cell Under Cursor
<b>L</b>	Leave: Title, Split-Screen, Window
<b>O</b>	Order of Recalculation (Row or Column)
<b>Q</b>	Quit Program
<b>R</b>	Recalculate: Automatic or Manual
<b>-</b>	Automatic Repetition of Characters at Cell Under Cursor

E: Edit Command	Description
<b>E C</b>	Copy Data Area to another Data Area
<b>E D</b>	Delete Row or Column
<b>E G</b>	Graphics: Histogram instead of Values
<b>E I</b>	Insert Row or Column
<b>E M</b>	Move Data Area to another Data Area
<b>E P</b>	Print Worksheet or User-Defined Format
<b>E R</b>	Replicate Data Area to other Data Areas
<b>E S</b>	Split Screen (Horizontally or Vertically)
<b>E T</b>	Title: Protects a Title in the Left Column
<b>E W</b>	Insert Window on Screen

F: Format Command	Description
<b>F C</b>	Select Colour
<b>F G</b>	Global Cell: Sets global format Global: Clears all Formats to CalcResult's normal power-up mode (labels left, values right and maximum precision)
<b>F M</b>	Maximum Precision display mode
<b>F I</b>	Integer display mode
<b>F \$</b>	Two Decimal display mode
<b>F L</b>	Sets Contents at Left
<b>F R</b>	Sets Contents at Right
<b>F *</b>	Replaces Integer Number digits with stars (always left justified)

P: Page Command	Description
<b>P A</b>	Add Pages, checking that label and formula match
<b>P C</b>	Copy one Page to another
<b>P D</b>	Delete Page from Work Area
<b>P E</b>	Erase Work Area
<b>P G</b>	Get Page from Work Area
<b>P N</b>	Negate: Change Signs (+ and -) in one Page
<b>P P</b>	Put 2nd Page from Work Area (to get extra memory)
<b>P R</b>	Renumber Page
<b>P +</b>	Add Pages, reading Values and Formulae only

G: Global Command	Description
<b>G C</b>	Sets Global Column Width, except in Protected Title-Column
<b>G F</b>	Set Format in all Cells
<b>G R</b>	Recalculate Pages by moving the highest column in one Page to the Alpha Column in the Next Page

D: Disk Command	Description
<b>D B</b>	Backup Drive 0 to Drive 1
<b>D C</b>	Catalog of Drive 1
<b>D D</b>	Save and Load DIF-files
<b>D E</b>	Erase File on Drive 1
<b>D I</b>	Initialize Drives 0 and 1
<b>D L</b>	Load File from Disk to Work Area
<b>D N</b>	New Disk (formatted in Drive 1)
<b>D S</b>	Save Work Area to Drive 1
<b>D U</b>	User Register: Contains language for Help screens, type of printer, paper format, etc. Type of Printer: 1 = 8023P 2 = 4022 4 = ASCII 3 = 8024, 8026, 8027, 8028, 8026b
<b>D V</b>	Load a VisiCalc-File



## Word Processor

### Special Keys:

<b>INST/DEL</b> Insert/delete character	<b>CTRL 9</b> Set reverse video for formatting instructions
<b>HOME</b> Move cursor to top line of text	<b>CTRL 0</b> Turn off reverse video
<b>CLR</b> Move cursor to bottom line of text	<b>C= C</b> Enter command mode
<b>RETURN</b> Terminate a paragraph	<b>F1 or C= L</b> Move cursor to left margin
<b>SHIFT RETURN</b> Move cursor to left margin of next line	<b>F2 or C= R</b> Move cursor to column 41
<b>SHIFT =</b> Tab key	<b>C= Q</b> Repeat previous keystroke
<b>CTRL =</b> Set a tab	<b>C= @</b> Replace line deleted by a RETURN

### Commands: All commands are initiated with C= C

<b>CA</b> Display disk directory (Catalog)	<b>DL</b> Delete a Line of text	<b>PR</b> Saves current document to disk with name "...tw" then prints it
<b>CB</b> Create a Block	<b>EP</b> Erase a Pointer	<b>RE</b> Search and Replace words or phrases
<b>CM</b> Clear Memory	<b>IB</b> Insert a Block created with CB	<b>RF</b> Save File to disk
<b>CP</b> Clear Pointers	<b>ID</b> Initialize Disk	<b>SP</b> Set a Pointer
<b>CT</b> Clear Tabs	<b>IL</b> Insert a Line of text	<b>SR</b> Search for a word or phrase
<b>DB</b> Delete Block	<b>LF</b> Load a File from disk	<b>*P</b> Print document
<b>DF</b> Delete a disk File	<b>MF</b> Merge a File from disk into text	

### Formatting Instructions: (enter in lowercase)

<b>ASC</b> Send an ASCII character to the printer	<b>OTHER</b> Used for non-Commodore printers (standard ASCII)
<b>CENTER</b> Center the text on the current line	<b>PAGELENn;</b> Set the number of lines on a page to 'n' lines (default 60)
<b>JUSTIFY</b> Right-justify text	<b>PAGEPAUSE</b> Stops printing after each page
<b>LINKFILE</b> Links documents at print time	<b>PAPERSIZEn;</b> Sets up paper size to 'n' lines long (default 66)
<b>LMARn;</b> Set left margin to 'n' (default 0)	<b>PAUSE</b> Stops printing until RETURN is pressed
<b>NEXTPAGE</b> Forces a new page	<b>RMARn;</b> Sets the right margin to 'n' (default 77)
<b>NOJUSTIFY</b> Turns off right justification (default)	<b>SET*PGn;</b> Sets page number to 'n'
<b>NOWRAP</b> Turns off word-wrap; used for spreadsheet tables	<b>*PAGE</b> Prints page number at bottom of each page
<b>NO*PAGE</b> Turns off page numbering	<b>WRAPON</b> Turns word-wrap on (default)

## Spreadsheet

### Special Keys:

<b>Cursor Down</b> moves the cursor down a cell	<b>F1 or C= L</b> moves the cursor left a cell	<b>C= T</b> Enter text in current cell
<b>Cursor Up</b> moves the cursor up a cell	<b>C= C</b> enters command mode	<b>C= F</b> Enter a formula in current cell
<b>F2 or C= R</b> moves the cursor right a cell	<b>C= Q</b> repeats last command	<b>C= N</b> Enter a number in current cell

### Commands: (Command mode is entered with C= C)

<b>AUTO</b> Turns on automatic calculation mode	<b>HOME</b> Moves the cursor to cell 1;1
<b>BLKMAPr;c</b> Moves block of cells from cursor to 'r;c' into the Word Processor	<b>ID</b> Initialize Disk
<b>CA</b> Display disk directory	<b>IN</b> Displays number in current cell in integer format
<b>CCO c;</b> Copies column 'c' to the cursor's column	<b>LEFTJ</b> Left justifies number in current cell
<b>CDEL</b> Deletes the current column	<b>LF</b> Load spreadsheet File from disk
<b>CINS</b> Inserts a new column	<b>MAN</b> Manual calculation mode (default)
<b>CM</b> Clear memory; deletes current spreadsheet	<b>MAP</b> Maps cell contents into the Word Processor
<b>COLOR n;</b> Changes the screen colour to colour 'n' (default 0)	<b>OFF</b> Turns off MAP mode (default)
<b>COPY r;c</b> Copies cell 'r;c' to the current cell	<b>RCO r;</b> Copies row 'r' to the current row
<b>DF</b> Delete a disk file	<b>RDEL</b> Deletes the current row
<b>FIT r;c</b> Copies the formula in 'r;c' to current cell and adjusts it to reflect the new cell position	<b>RESET</b> System reset (same as pressing RESET button)
<b>FL</b> Puts number in current cell in floating point format	<b>RIGHTJ</b> Right justifies number in current cell (default)
<b>FORMAT</b> Format a disk	<b>RINS</b> Inserts a new row
<b>FRE</b> Freeze - locks a cell - cannot be modified until THAWed	<b>SF</b> Saves current spreadsheet to disk
<b>FU</b> Full screen display mode (default)	<b>THAW</b> Unfreezes a frozen cell
<b>GOTO r;c</b> Moves the cursor to cell 'r;c'	<b>TW</b> To the Word Processor
<b>HA</b> Half screen display mode - allows simultaneous display of Word processor and spreadsheet	<b>\$\$</b> Displays number in current cell in dollar format (two decimal places)

### Arithmetic Operators:

<b>*</b> Indicates a numeric constant in formula	<b>DIV r1;c1 TO r2;c2</b> Divides a series of numbers in a row or column
<b>+, -, *, /</b> Add, Subtract, Mult, Divide	<b>MAX r1;c1 TO r2;c2</b> Gives the largest value of the specified row or column
<b>↑</b> Exponentiation	<b>MIN r1;c1 TO r2;c2</b> Gives the smallest value of the specified row or column
<b>EXP</b> Raises e (2.71828183) to a given power	<b>MLT r1;c1 TO r2;c2</b> Multiplies all values in the given row or column
<b>LOG</b> Calculates logarithm	<b>SUB r1;c1 TO r2;c2</b> Subtracts all values in the given row or column
<b>ABS</b> Absolute value	<b>SUM r1;c1 TO r2;c2</b> Adds all values in the given row or column
<b>ATN</b> Arctangent (in radians)	<b>r1;c1 ← r2;c2</b> Moves the contents of cell 'r2;c2' to cell 'r1;c1'
<b>COS</b> Cosine	<b>IFTRUE</b> Used with ← to move the contents of a cell to another if the condition is true
<b>SIN</b> Sine in radians	<b>IFTRUE operators: =, &gt;, &lt;, nte (not =), not</b>

## File Manager

### Commands: (C= C enters command mode)

<b>CA</b> Display disk directory	<b>RV n;</b> Reviews records in a file starting with record 'n' (pause with S, stop with Q)
<b>DS f1;f2;f3</b> DiskSort - Sorts a disk file by specified fields (up to 3)	<b>PI</b> Pick a range of records meeting certain criteria to create a subfile
<b>HIGHRC n;</b> Specifies max record for sorts, searches, reviews, selects, reports	<b>SR</b> Search for a record
<b>NR</b> Next Record - updates current record and displays next record	<b>TC</b> Move to the Spreadsheet
<b>RC n;</b> Displays record number 'n'	<b>TF</b> Display filename, number of records left, and the last record " entered
<b>RESETLIST</b> Sets upper record limit set by HIGHRC to maximum number of records in the file	<b>TW</b> To the Word Processor
	<b>UD</b> Update Record - files displayed record; use UDn; to file under record " 'n'

### Word Processor commands used with the File Manager

<b>TF::RC;</b> Indicates that the document is using File Manager data	<b>FLD n;</b> Prints the contents of field number 'n'
<b>RC n;</b> Start printed output with record number 'n'	<b>*RC</b> Prints the record number
<b>TTL n;</b> Prints the name of field number 'n'	<b>EOF?</b> If placed at the end of a document, causes output to continue for all records in the file

# Machine Language Monitor Commands

The following is a summary of typical MLM commands. Command syntax shown may vary slightly between different monitors.

<b>ASSEMBLE</b> .A 2000 BEQ \$2010	Assemble at address \$2000. Branch offsets are calculated.	<b>QUICK TRACE</b> .Q 1000	Trace code from \$1000 (or PC if no address specified), disassembly suppressed.
<b>BANK</b> .BBIN .BBOU .BKIN .BKOUT	Bank BASIC IN (Commodore 64) Bank BASIC OUT Bank Kernal IN Bank Kernal OUT	<b>POWER ON RESET</b> .P	Executes BASIC cold start
<b>BREAK SET</b> .B 1000 00FF	Sets a break at 1000 HEX on the FF HEX occurrence of the instruction at 1000.	<b>REGISTER DISPLAY</b> .R	Displays the PC, IRQ, Status or .P, .A, .X, .Y, and Stack Pointer.
<b>COMPARE MEMORY</b> .C 1000 2000 C000	Print the locations of bytes from \$1000 to \$2000 that are unequal to corresponding memory at \$C000.	<b>SAVE</b> .S "1:FILENAME",08,7000,8000	Save to drive 1 from \$7000 to \$7FFF (end address -1)
<b>DISASSEMBLE</b> .D 2000 3000	Disassemble from \$2000 to \$3000 (second parameter optional).	<b>TRANSFER MEMORY</b> .T 1000 1FFF 7000	Memory from \$1000 to \$1FFF is transferred to \$7000
<b>FILL</b> .F 1000 2000 FF	Fills memory from \$1000 to \$2000 with \$FF.	<b>WALK CODE</b> .W 1000	Single step code from \$1000 (or PC if no address specified) and disassemble each code executed.
<b>GO</b> .G 1000	Execute code at \$1000. Uses PC register as start address if none specified.	<b>EXIT TO BASIC</b> .X .E .K	Returns to BASIC READY mode. In Micromon, combines .X with .K. In Micromon, restores BRK & IRQ vectors
<b>HUNT</b> .H C000 D000 'READ' .H C000 D000 20 D2 FF	Hunt for the ASCII string "READ" from \$C000 to \$D000. Hunt for the byte sequence of 20 D2 FF	<b>CHANGE CHARACTER SETS</b> .Z	Upper Case/Graphics to Lower/Upper Case mode or vice versa.
<b>INTERROGATE</b> .I 7000 8000	Displays memory from \$7000 to \$8000 with screen printable characters.	<b>HEX CONVERSION</b> .4142	Displays Dec (16706), the ASCII characters (a b), and Binary (0100 0001 0100 0010)
<b>LOAD</b> .L "FILENAME",08	Load file from device 8, BASIC text pointers unaltered.	<b>DECIMAL CONVERSION</b> .#16706	Displays Hex (\$4142) followed by ASCII and Binary as above.
<b>MEMORY DISPLAY</b> .M 0000 0100	Display memory from \$0000 to \$0100.	<b>BINARY CONVERSION</b> . % 0100000101000010	Displays Hex, Decimal, followed by ASCII
<b>NEW LOCATE</b> .N 1000 17FF 6000 1000 1FFF [W]	Relocate code from 1000 to \$17FF at \$6000, adjusting any address within \$1000 to \$1FFF. Use W to adjust WORD tables.	<b>ASCII CONVERSION</b> . "A	Displays Hex (41), Decimal (65), and Binary (0100 0001)
<b>CALCULATE BRANCH OFFSET</b> .O 6000 5FFF FD	Calculate Branch Offset from \$6000 to \$5FFF (Result is \$FD)	<b>ADD</b> .+ 8000 7FFF	Displays the sum of the two Hex values (FFFF)
		<b>SUBTRACT</b> .- FFFF 7FFF	Displays the difference of the two Hex values (8000)
		<b>CHECKSUM</b> . & 7000 7FFF	Displays a Checksum of memory from \$7000 to \$7FFF

## Assembler Commands

### Assembler Pseudo-Ops

<b>.BYTE</b>	Place bytes in memory according to the operands specified
<b>.DBYTE</b>	Place 16-bit values in memory, stored hi order, low order (not in PAL)
<b>.END</b>	Ends assembly of a source file
<b>.FIL</b>	(.FILE in PAL) Links another source file to the current one
<b>.LIB</b>	Allows Library files to be inserted during assembly
<b>.OPT</b>	Sets options for assembly
<b>.PAGE</b>	Advances the listing to a new page (noy in PAL)
<b>.SKIP</b>	Generates blank lines in listing
<b>.TEXT</b>	(.ASC in PAL) Puts a string of ASCII characters in memory
<b>.WORD</b>	Puts 16-bit values in memory, stored low order, high order
<b>* =</b>	Set program counter to a given address
<b>=</b>	Equate: assigns a value to a symbol
<b>* = * + N</b>	Reserve N bytes for data storage

### Additional PAL Pseudo-Ops

<b>.IF</b>	Conditional assembly pseudo-op. Follow with EXPR: and the source code to assemble if EXPR is true.
<b>.GOTO</b>	Transfers assembly to the line number specified.
<b>.GTB</b>	Go To BASIC. Exits assembly and enables the BASIC interpreter.
<b>.STM</b>	Symbol Table Minimum. Prevents the Symbol Table from inhabiting memory below the specified address.
<b>.SST</b>	Save Symbol Table
<b>.LST</b>	Load Symbol Table
<b>.SYS</b>	JSR to the specified address during assembly (either pass).

### Prefix Characters

.	Indicates an assembler directive
#	Immediate Addressing mode
( )	Indirect Addressing mode
!	Forces Zero-Page Addressing mode
\$	Specifies a hexadecimal value
%	Specifies a binary value
@	Specifies an octal value
'	Specifies an ASCII literal
;	Indicates that comments follow
<	Specifies the low byte of a 16-bit value.
>	Specifies the high byte of a 16-bit value

### Expression Operators

+	Add values or expressions.
-	Subtract
*	Multiply
!	Boolean OR
&	Boolean AND
↑	Boolean Exclusive OR
<	Placed to the right of an expression specifies the expression shifted left n bits. EXPR<4 would shift EXPR left 4 bits. EXPR can be 16 bits.
>	Placed to the right of an expression specifies the expression shifted right n bits. EXPR<4 would shift EXPR right 4 bits.
!	Forces Absolute Addressing

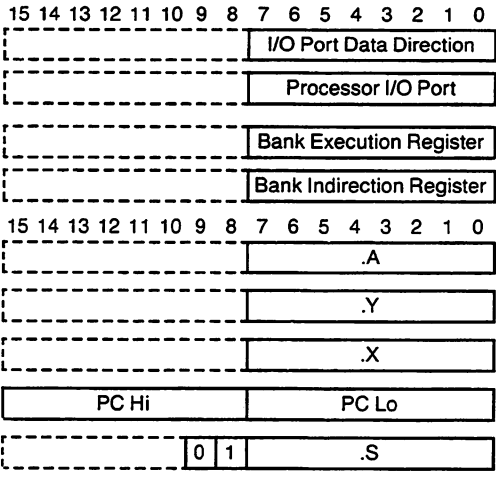
### CBM .OPT Directives

<b>ERR</b>	Generate Error File (default)
<b>NOE</b>	Suppress Error File generation
<b>LIST</b>	Generate Listing File containing the Assembler output, including errors, comments, symbol table, etc. (default)
<b>NOL</b>	Suppress Listing File
<b>MEM</b>	Generate Memory File (default)
<b>NOM</b>	Suppress Memory File
<b>GEN</b>	Display beyond the first two bytes of a .BYTE (ie. for ASCII strings)
<b>NOG</b>	Show only the first two bytes of a .BYTE directive. (default)

### PAL .OPT Directives

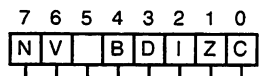
<b>P</b>	Print Assembly Listing
<b>Pn</b>	Print Assembly Listing to the previously OPENed logical file n.
<b>P=</b>	Print through a user routine at the address specified after the = sign (character in .A)
<b>O</b>	Output Object code to BASIC Arrays memory
<b>OO</b>	Output Object code to Origin
<b>On</b>	Output Object code to the previously OPENed logical file n (start address included).
<b>O=</b>	Output Object code through a user routine at the address specified after the = sign.
<b>N</b>	Null or reset .OPT directives

CPU Model



6510 (C64), 7501 (+ 4/C16)

6509 (B Series)



- .P - Processor Status
1 = Carry or No Borrow
1 = Result Zero
1 = IRQ Disabled
1 = Dec, 0 = Binary Mode
BRK Command = 1
Not Used
1 = Overflow
1 = Negative

Table with columns: Mde, IMM, ZPg, Z.X, (I),Y, ABS, A.X, A.Y. Rows include ORA, AND, EOR, ADC, STA, LDA, CMP, SBC.

Table with columns: Mde, IMM, ZPg, Z.X, ABS, A.X. Rows include BIT, STY, LDY, CPY, CPX.

Branches -0 table with columns: Mde, ABS, (IND). Rows include BPL, BVC, BCC, BNE.

Jumps table with columns: Mde, ABS, (IND). Rows include JSR, JMP.

Table with columns: Mde, IMM, ZPg, Z.X, Z.Y, ABS, A.X, A.Y. Rows include ASL, ROL, LSR, ROR, STX, LDX, DEC, INC.

Single Byte Op Codes (\* Accumulator Mode) table with columns: 0-, 1-, 2-, 3-, 4-, 5-, 6-, 7-, 8-, 9-, A-, B-, C-, D-, E-, F-. Rows include BRK, PHP, ASL, etc.

6502 Extra Op-Codes

The table shows Op-Codes that are not generally recognized as part of the 650X Instruction Set. Mnemonics and descriptions are from B. Grainger's article in IPUG (Jan 1981) and "Programming the PET/CBM" by Raeto Collin West

Table with columns: Instruction, Description, Abs, Abs,X, Abs,Y, Zer, Zer,X, Zer,Y, (Ind,X), (Ind,Y), Imm. Rows include ASO, RLA, LSE, RRA, AXS, LAX, DCM, INS, ALR, ARR, XAA, OAL, SAX, MKA, MKX, NOP, SKB, SKW.

Hexadecimal Conversion Chart

Hexadecimal Conversion Chart table with columns: Hex, -0, -1, -2, -3, -4, -5, -6, -7, -8, -9, -A, -B, -C, -D, -E, -F, -00, -000.

Bit Values

Bit Values table with columns: Bit, Dec, Hex. Rows include 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15.



Instr	Addressing Mode	Assembler Format	Operation	Op Code Hex	Dec	Bytes	Clock Cycles	Status Register - P	Instr
<b>LDA</b>	Immediate	LDA #oper	# → A	A9	169	2	2	N V D I Z C	<b>LDA</b>
	Zero Page	LDA addr	[addr] → A	A5	165	2	3	✓ - - - ✓ -	
	Zero Page, X	LDA addr, X	[addr+.X] → A	B5	181	2	4		
	Absolute	LDA ADDR	[ADDR] → A	AD	173	3	4		
	Absolute, X	LDA ADDR, X	[ADDR+.X] → A	BD	189	3	4*		
	Absolute, Y	LDA ADDR, Y	[ADDR+.Y] → A	B9	185	3	4*		
	(Indirect, X)	LDA (addr, X)	[[addr+.X+1, addr+.X]] → A	A1	161	2	6		
	(Indirect, Y)	LDA (addr, Y)	[[addr+.1, addr+.Y] → A	B1	177	2	5*		
<b>LDX</b>	Immediate	LDX #oper	# → X	A2	162	2	2	N V D I Z C	<b>LDX</b>
	Zero Page	LDX addr	[addr] → X	A6	166	2	3	✓ - - - ✓ -	
	Zero Page, Y	LDX addr, Y	[addr+.Y] → X	B6	182	2	4		
	Absolute	LDX ADDR	[ADDR] → X	AE	174	3	4		
	Absolute, Y	LDX ADDR, Y	[ADDR+.Y] → X	BE	190	3	4*		
<b>LDY</b>	Immediate	LDY #oper	# → Y	A0	160	2	2	N V D I Z C	<b>LDY</b>
	Zero Page	LDY addr	[addr] → Y	A4	164	2	3	✓ - - - ✓ -	
	Zero Page, X	LDY addr, X	[addr+.X] → Y	B4	180	2	4		
	Absolute	LDY ADDR	[ADDR] → Y	AC	172	3	4		
	Absolute, X	LDY ADDR, X	[ADDR+.X] → Y	BC	188	3	4*		
<b>LSR</b>	Accumulator	LSR A	.A (←) → .A ; 0→bit7, bit0→C	4A	74	1	2	N V D I Z C	<b>LSR</b>
	Zero Page	LSR addr	[addr] (←) → [addr]	46	70	2	5	0 - - - ✓ ✓	
	Zero Page, X	LSR addr, X	[addr+.X] (←) → [addr+.X]	56	86	2	6		
	Absolute	LSR ADDR	[ADDR] (←) → [ADDR]	4E	78	3	6		
	Absolute, X	LSR ADDR, X	[ADDR+.X] (←) → [ADDR+.X]	5E	94	3	7		
<b>NOP</b>	Implied	NOP	No Operation	EA	234	1	2	- - - - -	<b>NOP</b>
<b>ORA</b>	Immediate	ORA #oper	.A U # → A	09	9	2	2	N V D I Z C	<b>ORA</b>
	Zero Page	ORA addr	.A U [addr] → A	05	5	2	3	✓ - - - ✓ -	
	Zero Page, X	ORA addr, X	.A U [addr+.X] → A	15	21	2	4		
	Absolute	ORA ADDR	.A U [ADDR] → A	0D	13	3	4		
	Absolute, X	ORA ADDR, X	.A U [ADDR+.X] → A	1D	29	3	4*		
	Absolute, Y	ORA ADDR, Y	.A U [ADDR+.Y] → A	19	25	3	4*		
	(Indirect, X)	ORA (addr, X)	.A U [[addr+.X+1, addr+.X]] → A	0i	1	2	6		
	(Indirect, Y)	ORA (addr, Y)	.A U [[addr+.1, addr+.Y] → A	11	17	2	5*		
<b>PHA PLA PHP PLP</b>	Implied	PHA	.A †, SP - 1 → SP	48	72	1	3	N V D I Z C	<b>PHA PLA PHP PLP</b>
	Implied	PLA	.A †, SP + 1 → SP	68	104	1	4	- - - - -	
	Implied	PHP	.P †, SP - 1 → SP	08	8	1	3	All Push/Pulls xcpt PLP from stack	
	Implied	PLP	.P †, SP + 1 → SP	28	40	1	4		
<b>ROL</b>	Accumulator	ROL A	.A (←) → A ; C→bit0, bit7→C	2A	42	1	2	N V D I Z C	<b>ROL</b>
	Zero Page	ROL addr	[addr] (←) → [addr]	26	38	2	5	✓ - - - ✓ ✓	
	Zero Page, X	ROL addr, X	[addr+.X] (←) → [addr+.X]	36	54	2	6		
	Absolute	ROL ADDR	[ADDR] (←) → [ADDR]	2E	46	3	6		
	Absolute, X	ROL ADDR, X	[ADDR+.X] (←) → [ADDR+.X]	3E	62	3	7		
<b>ROR</b>	Accumulator	ROR A	.A (→) → A ; C→bit7, bit0→C	6A	106	1	2	N V D I Z C	<b>ROR</b>
	Zero Page	ROR addr	[addr] (→) → [addr]	66	102	2	5	✓ - - - ✓ ✓	
	Zero Page, X	ROR addr, X	[addr+.X] (→) → [addr+.X]	76	118	2	6		
	Absolute	ROR ADDR	[ADDR] (→) → [ADDR]	6E	110	3	6		
	Absolute, X	ROR ADDR, X	[ADDR+.X] (→) → [ADDR+.X]	7E	126	3	7		
<b>RTI</b>	Implied	RTI	P †, PC †, SP + 3 → SP, PC + 1 → PC	40	64	1	6	from stack	<b>RTI</b>
<b>RTS</b>	Implied	RTS	PC †, SP + 2 → SP, PC + 1 → PC	60	96	1	6	- - - - -	<b>RTS</b>
<b>SBC</b>	Immediate	SBC #oper	.A - # - C̄ → A, C C̄ = Borrow	E9	233	2	2	N V D I Z C	<b>SBC</b>
	Zero Page	SBC addr	.A - [addr] - C̄ → A, C	E5	229	2	3	✓ ✓ - - ✓ ✓	
	Zero Page, X	SBC addr, X	.A - [addr+.X] - C̄ → A, C	F5	245	2	4		
	Absolute	SBC ADDR	.A - [ADDR] - C̄ → A, C	ED	237	3	4		
	Absolute, X	SBC ADDR, X	.A - [ADDR+.X] - C̄ → A, C	FD	253	3	4*		
	Absolute, Y	SBC ADDR, Y	.A - [ADDR+.Y] - C̄ → A, C	F9	249	3	4*		
	(Indirect, X)	SBC (addr, X)	.A - [[addr+.X+1, addr+.X]] - C̄ → A, C	E1	225	2	6		
	(Indirect, Y)	SBC (addr, Y)	.A - [[addr+.1, addr+.Y] - C̄ → A, C	F1	241	2	5*		
<b>SEC SED SEI</b>	Implied	SEC	1→C	38	56	1	2	N V D I Z C	<b>SEC SED SEI</b>
	Implied	SED	1→D	F8	248	1	2	- - - 1 - - -	
	Implied	SEI	1→I	78	120	1	2	- - - 1 - - -	
<b>STA</b>	Zero Page	STA addr	.A → [addr]	85	133	2	3	N V D I Z C	<b>STA</b>
	Zero Page, X	STA addr, X	.A → [addr+.X]	95	149	2	4	- - - - -	
	Absolute	STA ADDR	.A → [ADDR]	8D	141	3	4		
	Absolute, X	STA ADDR, X	.A → [ADDR+.X]	9D	157	3	5		
	Absolute, Y	STA ADDR, Y	.A → [ADDR+.Y]	99	153	3	5		
	(Indirect, X)	STA (addr, X)	.A → [[addr+.X+1, addr+.X]]	81	129	2	6		
	(Indirect, Y)	STA (addr, Y)	.A → [[addr+.1, addr+.Y]	91	145	2	6		
<b>STX</b>	Zero Page	STX addr	.X → [addr]	86	134	2	3	N V D I Z C	<b>STX</b>
	Zero Page, Y	STX addr, Y	.X → [addr+.Y]	96	150	2	4	- - - - -	
	Absolute	STX ADDR	.X → [ADDR]	8E	142	3	4		
<b>STY</b>	Zero Page	STY addr	.Y → [addr]	84	132	2	3	N V D I Z C	<b>STY</b>
	Zero Page, X	STY addr, X	.Y → [addr+.X]	94	148	2	4	- - - - -	
	Absolute	STY ADDR	.Y → [ADDR]	8C	140	3	4		
<b>TAX TXA TAY TYA TSX TXS</b>	Implied	TAX	.A → X	AA	170	1	2	N V D I Z C	<b>TAX TXA TAY TYA TSX TXS</b>
	Implied	TXA	.X → A	8A	138	1	2	✓ - - - ✓ -	
	Implied	TAY	.A → Y	A8	168	1	2		
	Implied	TYA	.Y → A	98	152	1	2		
	Implied	TSX	SP → X	BA	186	1	2	All Transfers xcpt TXS	
	Implied	TXS	X → SP	9A	154	1	2	- - - - -	

## MCS65XX Microprocessor Instruction Set

Mnemonic	Definition
ADC	Add memory to accumulator with carry.
AND	AND memory with accumulator.
ASL	Shift left one bit (memory or accumulator).
BCC	Branch on carry clear.
BCS	Branch on carry set.
BEQ	Branch on result zero.
BIT	Test bits in memory with accumulator.
BMI	Branch on result minus.
BNE	Branch on result not zero.
BPL	Branch on result plus.
BRK	Force break.
BVC	Branch on overflow clear.
BVS	Branch on overflow set.
CLC	Clear carry flag.
CLD	Clear decimal mode.
CLI	Clear interrupt disable bit.
CLV	Clear overflow flag.
CMP	Compare memory and accumulator.
CPX	Compare memory and index 'X'.
CPY	Compare memory and index 'Y'.
DEC	Decrement memory by one.
DEX	Decrement index 'X' by one.
DEY	Decrement index 'Y' by one.
EOR	Exclusive-OR memory with accumulator.
INC	Increment memory by one.
INX	Increment index 'X' by one.
INY	Increment index 'Y' by one.
JMP	Jump to new location.
JSR	Jump to new location saving return address.
LDA	Load accumulator with memory.
LDX	Load index 'X' with memory.
LDY	Load index 'Y' with memory.
LSR	Shift right one bit (memory or accumulator).
NOP	No operation.
ORA	OR memory with accumulator.
PHA	Push accumulator on stack.
PHP	Push processor status on stack.
PLA	Pull accumulator from stack.
PLP	Pull processor status from stack.
ROL	Rotate one bit left (memory or accumulator).
ROR	Rotate one bit right (memory or accumulator).
RTI	Return from interrupt.
RTS	Return from subroutine.
SBC	Subtract memory from accumulator with borrow.
SEC	Set carry flag.
SED	Set decimal mode.
SEI	Set interrupt disable status.
STA	Store accumulator in memory.
STX	Store index 'X' in memory.
STY	Store index 'Y' in memory.
TAX	Transfer accumulator to index 'X'.
TAY	Transfer accumulator to index 'Y'.
TSX	Transfer stack pointer to index 'X'.
TXA	Transfer index 'X' to accumulator.
TXS	Transfer index 'X' to stack pointer.
TYA	Transfer index 'Y' to accumulator.

## Addressing Modes

**Accumulator Addressing** - This form of addressing is represented with a one byte instruction, implying an operation on the accumulator.

**Immediate Addressing** - In immediate addressing, the operand is contained in the second byte of the instruction, with no further memory addressing required.

**Absolute Addressing** - In absolute addressing, the second byte of the instruction specifies the eight low order bits of the effective address while the third byte specifies the eight high order bits. Thus, the absolute addressing mode allows access to the entire 65k bytes of addressable memory.

**Zero Page Addressing** - The zero page instructions allow for shorter code and execution times by only fetching the second byte of the instructions and assuming a zero high address byte. Careful use of the zero page can result in significant increase in code efficiency.

**Indexed Zero Page Addressing** - (X, Y Indexing) - This form of addressing is used in conjunction with the index register and is referred to as "Zero Page, X" or "Zero Page, Y". The effective address is calculated by adding the second byte to the contents of the index register. Since this is a form of "Zero Page" addressing, the content of the second byte references a location in page zero. Additionally due to the "Zero Page" addressing nature of this mode, no carry is added to the high order 8 bits of memory and crossing of page boundaries does not occur.

**Indexed Absolute Addressing** - (X, Y Indexing) - This form of addressing is used in conjunction with X and Y index register and is referred to as Absolute, X", and "Absolute, Y". The effective address is formed by adding the contents of X or Y to the address contained in the second and third bytes on the instruction. This mode allows the index register to contain the index or count value and the instruction to contain the base address. This type of indexing allows any location referencing and the index to modify multiple fields resulting in reduced coding and execution time.

**Implied Addressing** - In the implied addressing mode, the address containing the operand is implicitly stated in the operation code of the instruction.

**Relative Addressing** - Relative addressing is used only with branch instructions and establishes a destination for the conditional branch. The second byte of the instruction becomes the operand which is an "offset" added to the contents of the lower eight bits of the program counter when the counter is set at the next instruction. The range of the offset is -128 to +127 bytes from the next instruction.

**Indexed Indirect Addressing** - In indexed indirect addressing (referred to as (Indirect, X)), the second byte of the instruction is added to the contents of the X index register, discarding the carry. The result of the addition points to a memory location on page zero whose contents is the low order eight bits of the effective address. The next memory location in page zero contains the high order eight bits of the effective address. Both memory locations specifying the high and low order bytes of the effective address must be in page zero.

**Indirect Indexed Addressing** - In indirect indexed addressing (referred to as (Indirect, Y)), the second byte of the instruction points to a memory location in page zero. The contents of this memory location is added to the contents of the Y register, the result being the low order eight bits of the effective address. The carry from this addition is added to the contents of the next page zero memory location, the result being the high order eight bits of the effective address.

**Absolute Indirect** - The second byte of the instruction contains the low order eight bits of a memory location. The high order eight bits of that memory location is contained in the third byte of the instruction. The contents of the fully specified memory location is the low order byte of the effective address which is loaded into the sixteen bits of the program counter.

# User Callable ROM Subroutines

Some I/O routines require extra memory set up. See the appropriate Memory Map. Address pairs within parenthesis are for Basic 2.0/4.0 users. (Direct call) indicates no required set up.

#	Entry Point For:								Operation	Registers In			Registers Out		
	2.0	4.0	VIC 20		C64		A	X		Y	A	X	Y		
1	C2D8	49880	B350	45904	C3BB	50107	A3BB	41915	Open Up Space In BASIC Text	New:	AryTop Lo		AryTop Hi	Unaltered	
2	C328	49960	B3A0	45984	C408	50184	A408	41992	Check Available Memory (called by 1)	(same as above) Start address of move in \$5F, 60 (\$5C, 5D)					
3	C355	50005	B3CD	46029	C435	50229	A435	42037	?OUT OF MEMORY	(direct call)					
4	C357	50007	BC3F	48191	C437	50231	A437	42039	Send BASIC Error Message	Error #					
5	C389	50057	B3FF	46079	C474	50292	A474	42100	Warm start, BASIC	(direct call)					
6	C399	49960	B40D	46093	C48A	50314	A48A	42122	Main CHRGET entry	(direct call) \$7A = #\$FF, \$7B = #\$01 (\$77, 78) ;01FF = Basic Inbuf-1					
7	C3AB	50091	B41F	46111	C49C	50220	A49C	42028	Crunch tokens, insert line	Inbuf len.					
8	C439	50233	B4AD	46253	C52A	50474	A52A	42282	Fix chaining, CLR, & READY.	(direct call)					
9	C442	50242	B4B6	46262	C533	50483	A533	42291	Fix chaining	(direct call)					
10	C46F	50287	B4E2	46306	C560	50528	A560	42336	Receive line from keyboard	(direct call) \$7A = #\$FF, \$7B = #\$01 (\$77, 78) ;01FF = Basic Inbuf-1					
11	C495	50213	B4FB	46331	C579	50553	A579	42361	Crunch tokens (called by 7)	.X = Inbuf Len. (\$0200,X) = #\$00					
12	C52C	50476	B5A3	46499	C613	50707	A613	42515	Find line in BASIC	StrtBAS Lo	StrtBAS Hi				
13	C55D	50525	B5D4	46548	C642	50754	A642	42562	Do NEW	(direct call)					
14	C572	50546	B5E9	46569	C659	50777	A659	42585	Reset BASIC and do CLR	(direct call)					
15	C575	50549	B5EC	46572	C65E	50782	A65E	42590	Do CLR	(direct call)					
16	C597	50583	B612	46610	n/a	n/a	n/a	n/a	Purge stack of all Returns & Nexts (POP)	(direct call)					
17	C5A7	50599	B622	46626	C68E	50830	A68E	42638	Reset Chrget to Start of BASIC	(direct call)					
18	C6C4	50884	B74A	46922	C857	51287	A857	43095	Continue BASIC execution [CONT]	CurLin Lo		CurLin Hi			
19	C873	51315	B8F6	47350	C96B	49771	A96B	41579	Get fixed-pt number from BASIC text	Address of text in Chrget ptr; \$7A, 7B (\$77, 78)					
20	C9DE	49886	BADB	47835	CAD3	51923	AAD3	43731	Send RETURN, LF if in screen mode	(direct call)					
21	C9E2	49890	BADF	47839	CAD7	51927	AAD7	43735	Send RETURN, LINEFEED	(direct call)					
22	CA1C	51740	BB1D	47901	CB1E	51998	AB1E	43806	Print: string from A, Y	Addr Lo		Addr Hi			
23	CA22	51746	BB23	47907	CB24	52004	AB24	43812	Print pre-computed string	Length	Addr in \$22,23 (\$1F,20)				
24	CA43	51779	BB44	47940	CB45	52037	AB45	43845	Print '?'	(direct call)					
25	CA45	51781	BB46	47942	CB47	52039	AB47	43847	Print char (output .A to device)	Char			Char		
26	CC9F	52383	BD98	48536	CD9E	52638	AD9E	44446	Evaluate Result: string \$0D = #\$FF (\$07) Expression numeric \$0D = #\$00 (\$07)	Address of Expression		Addr Lo		Addr Hi	
27	CD8F	52728	BEF5	48885	CEFF	52991	AFFD	44797	Check for comma	In Chrget Pointer			result in Acc#1		
28	CD7F	52727	BEF2	48882	CEFA	52986	AFFA	44794	Check for '('	(direct call)					
29	CD74	52724	BEEF	48879	CEF7	52983	AFF7	44791	Check for ')'	(direct call)					
30	CE03	52739	BF00	48896	CF08	53000	AF08	44808	Send 'SYNTAX ERROR'	(direct call)					
31	CFC9	53193	C187	49543	D0E7	53479	B0E7	45287	Find fl-pt variable, given name				VarAddr Lo		VarAddr Hi
32	D069	53353	C2B9	49849	D185	53637	B185	45445	Bump Variable Addr by 2 (called by 31)	Name in \$45, 46 (\$42, 43)		VarAddr Lo		VarAddr Hi	
33	D09A	53290	C2EA	49898	D1BF	53695	B1BF	45503	Float to Fixed conversion in Acc#1	(direct call)					
34	D26D	53869	C4BC	50364	D391	54049	B391	45857	Fixed to Float conversion in Acc#1	(direct call)					
35	D67B	54907	C8D7	51415	D79E	55086	B79E	46894	Get Acc#1 least significant byte to X register					Data	
36	D68F	54927	C8E8	51435	D7B5	55221	B7B5	47029	Evaluate string [VAL]	Address = (Chrget Ptr.)			Fl. Pt. result in Acc#1		
37	D69D	54931	C8EF	51439	D7B9	55225	B7B9	47033	Evaluate string from X, Y (above + 4)	Addr Lo	Addr Hi		Fl. Pt. result in Acc#1		
38	D6C6	54982	C921	49697	D7EB	55275	B7EB	47083	Get two params for POKE, WAIT	Address = (Chrget Ptr.)			.X = Pram2, Pram1 in Acc#1 (fxd pt)		
39	D773	55155	C99D	49709	D867	55399	B867	47207	Add (from memory)	Addr Lo		Addr Hi		Fl. Pt. result in Acc#1	
40	D934	53812	CB5E	52062	DA28	55848	BA28	47656	Multiply from memory location	Addr Lo		Addr Hi		Fl. Pt. result in Acc#1	
41	D9EE	53998	CC18	52248	DAE2	56034	BAE2	47842	Multiply Acc#1 by ten					(result in Acc#1)	
42	DAAE	55982	CCD8	52440	DBA2	56226	BBA2	48034	Unpack memory variable to Acc#1	Addr Lo		Addr Hi			
43	DAE3	56035	CD0D	52493	DBD7	56279	BBD7	48087	Copy Acc#1 to (X,Y) Location	Addr Lo	Addr Hi				
44	DB08	56072	CD32	52530	DBFC	56316	BBFC	48124	Move Acc#2 to Acc#1	(direct call)					
45	DB18	56088	CD42	52546	DC0C	56332	BC0C	48140	Move Rounded Acc#1 to Acc#2	(direct call)					
46	DB1D	56093	CD45	52549	DC0F	56335	BC0F	48143	Move Un-Rounded Acc#1 to Acc#2	(direct call)					
47	DB27	56103	CD51	52561	DC1B	56347	BC1B	48155	Round Acc.#1	(direct call)					
48	DCD9	56537	CF83	53123	DDCD	56781	BDCD	48589	Print fixed-point value	Value Hi	Value Lo				
49	DCE3	56547	CF8D	53133	DDD7	56791	BDD7	48599	Print floating-point value in Acc#1	(direct call)					
50	DCE9	56553	CF93	53027	DDDD	56797	BDDD	48605	Convert num to strng at \$0100 (called by 48)	#\$00			#\$01		
51	FD11	64785	D472	54386	n/a	n/a	n/a	n/a	Entry to M.L.M.	(direct call)					
52	E3D8	58328	E202	57858	E742	59202	E716	59158	Print a character	Char					
53	F156	61782	F185	61829	F1E6	61926	F12F	61743	Print system message				Offset		
54	FOB6	61622	F0D2	61650	EE14	60948	ED09	60681	Send 'talk' to IEEE/Serial	Dev #					
55	F0BA	61626	F0D5	61653	EE17	60951	ED0C	60684	Send 'listen' to IEEE/Serial	Dev #					
56	F128	61736	F143	61763	FF93	65427	FF93	65427	Send secondary address	SA OR \$60					
57	F16F	61807	F19E	61742	EEE4	61156	ED40	60736	Send char to IEEE/Serial	Char					
58	F17F	61823	F1AE	61870	EEF6	61174	EDEF	60911	Send 'untalk'	(direct call)					
59	F183	61827	F1B9	61881	EF04	61188	EDFE	60926	Send 'unlisten'	(direct call)					

## BASIC 4.0 / 2.0 Kernal Routines

CBM Label	Address		Operation	Registers In			Registers Out		
	Hex	Dec		.A	.X	.Y	.A	.X	.Y
CHKIN	FFC6	65478	Open channel for input		LF#		alt.		
CHKOUT	FFC9	65481	Open channel for output		LF#		alt.		
CHRIN	FFCF	65487	Input character from channel				data	alt.	
CHROUT	FFD2	65490	Output character to channel	data					
CLALL	FFE7	65511	Close all channels and files				alt.	alt.	
CLOSE	FFC3	65475	Close a specified logical file	LF#			alt.	alt.	alt.
CLRCHN	FFCC	65484	Restore default I/O devices				alt.	alt.	
CSYS	FFDE	65502	SYS vector		addr lo	addr hi	alt.	alt.	alt.
CVERF	FFDB	65499	Verify ram from a device		start lo	start hi		end lo + 1	end hi
GETIN	FFE4	65508	Get character from current input device				data	alt.	alt.
LOAD	FFD5	65493	Load ram from a device		start lo	start hi		end lo + 1	end hi
OPEN	FFC0	65472	Open a logical file				alt.	alt.	alt.
SAVE	FFD8	65496	Save 'ram' to device, from \$28,29 to .X,.Y	#<txttab (= # \$28)	end lo	end hi		end lo + 1	end hi
STOP	FFE1	65505	Scan stop key depressed	yes: .Z = 1, no .A = last row kybd scan					
UDTIM	FFEA	65514	Increment real time clock				alt.	alt.	

alt. = altered

## VIC 20 And Commodore 64 Kernal Routines

CBM Label	Address		Operation	Registers In			Registers Out		
	Hex	Dec		.A	.X	.Y	.A	.X	.Y
ACPTR	FFA5	65445	Input byte from Serial Port				data	alt.	
CHKIN	FFC6	65478	Open channel for input		LF#		alt.		
CHKOUT	FFC9	65481	Open channel for output		LF#		alt.		
CHRIN	FFCF	65487	Input character from channel				data	alt.	
CHROUT	FFD2	65490	Output character to channel	data					
CIOUT	FFA8	65448	Output byte to serial port	data					
CINT	FFB1	65409	Initialize screen editor				alt.	alt.	alt.
CLALL	FFE7	65511	Close all channels and files				alt.	alt.	
CLOSE	FFC3	65475	Close a specified logical file	LF#			alt.	alt.	alt.
CLRCHN	FFCC	65484	Restore default I/O devices				alt.	alt.	
GETIN	FFE4	65508	Get character from current input device				data	alt.	alt.
IOBASE	FFF3	65523	Returns base address of I/O devices					addr lo	addr hi
IOINIT	FFB4	65412	Initialize Input/Output				alt.	alt.	alt.
LISTEN	FFB1	65457	Command devices on the serial bus to listen	DEV#					
LOAD	FFD5	65493	Load (.A = 0) or Verify (.A = 1) 'ram' from a device		start lo	start hi		end lo + 1	end hi
MEMBOT	FF9C	65436	Read (.C = 1) or Set (.C = 0) the bottom of memory	.C = 0:	bot lo	bot hi	.C = 1:	bot lo	bot hi
MEMTOP	FF99	65433	Read (.C = 1) or Set (.C = 0) the top of memory	.C = 0:	top lo	top hi	.C = 1:	top lo	top hi
OPEN	FFC0	65472	Open a logical file				alt.	alt.	alt.
PLOT	FFF0	65520	Read (.C = 1) or Set (.C = 0) x, y cursor position		row	col		row	col
RAMTAS	FFB7	65415	Init. ram, allocate tape buff, set screen \$0400				alt.	alt.	alt.
RDTIM	FFDE	65502	Read real time clock				msb	msb2	lsb
READST	FFB7	65463	Read I/O status word				ST		
RESTOR	FFB8	65418	Restore default I/O vectors				alt.	alt.	alt.
SAVE	FFD8	65496	Save 'ram' to device, from \$2B,2C to .X,.Y	#<txttab (= # \$2B)	end lo	end hi		end lo + 1	end hi
SCNKEY	FF9F	65439	Scan keyboard				alt.	alt.	alt.
SCREEN	FFED	65517	Return screen size in rows & columns					#rows	#cols
SECOND	FF93	65427	Send secondary address after 'listen'	SA OR \$60					
SETLFS	FFBA	65466	Set logical, first, and second addresses	LF#	DEV#	SA			
SETMSG	FF90	65424	Enable/Disable 'Kernal' messages	.A val: \$40 control msgs on, \$80 error msgs on, \$00 off					
SETNAM	FFBD	65469	Set file name	len	addr lo	addr hi			
SETTIM	FFDB	65499	Set real time clock	msb	msb2	lsb			
SETTMO	FFA2	65442	Set (.A < #128) Reset (.A > #127) Serial/IEEE timeout						
STOP	FFE1	65505	Scan stop key depressed	yes: .Z = 1, no .A = last row kybd scan					
TALK	FFB4	65460	Command serial bus device to 'talk'	DEV#					
TKSA	FF96	65430	Send secondary address after 'talk'	SA					
UDTIM	FFEA	65514	Increment real time clock				alt.	alt.	
UNLSN	FFAE	65454	Command serial bus to 'unlisten'				alt.		
UNTLK	FFAB	65451	Command serial bus to 'untalk'				alt.		
VECTOR	FFB8	65421	Store (.C = 1) or Restore (.C = 0) ram vectors	.C = 1:	tabl lo	tabl hi	.C = 0:	tabl lo	tabl hi

alt. = altered





# SuperChart: BASIC 2.0 / 4.0

DECIMAL	HEX	ASCII	SCREEN	BASIC	6502	DECIMAL
0	00		@	end-line	BRK	0
1	01		A		ORA(I,X)	1
2	02		B			2
3	03	stop	C			3
4	04		D			4
5	05		E		ORA Z	5
6	06		F		ASL Z	6
7	07	bell	G			7
8	08		H		PHP	8
9	09	tab	I		ORA #	9
10	0A		J		ASL A	10
11	0B		K			11
12	0C		L			12
13	0D	car ret	M		ORA	13
14	0E	text	N		ASL	14
15	0F	top left	O			15
16	10		P		BPL	16
17	11	cur down	Q		ORA(I),Y	17
18	12	reverse	R			18
19	13	cur home	S			19
20	14	delete	T			20
21	15	del line	U		ORA Z,X	21
22	16	ers start	V		ASL Z,X	22
23	17		W			23
24	18		X		CLC	24
25	19	scroll dn	Y		ORA Y	25
26	1A		Z			26
27	1B	escape	[			27
28	1C		\			28
29	1D	cur right	]		ORA X	29
30	1E		↑		ASL X	30
31	1F		←			31
32	20	space	space	space	JSR	32
33	21	!	!	!	AND(I,X)	33
34	22	"	"	"		34
35	23	#	#	#		35
36	24	\$	\$	\$	BIT Z	36
37	25	%	%	%	AND Z	37
38	26	&	&	&	ROL Z	38
39	27	'	'	'		39
40	28	(	(	(	PLP	40
41	29	)	)	)	AND #	41
42	2A	*	*	*	ROL A	42
43	2B	+	+	+		43
44	2C	,	,	,	BIT	44
45	2D	-	-	-	AND	45
46	2E	.	.	.	ROL	46
47	2F	/	/	/		47
48	30	0	0	0	BMI	48
49	31	1	1	1	AND(I),Y	49
50	32	2	2	2		50
51	33	3	3	3		51
52	34	4	4	4		52
53	35	5	5	5	AND Z,X	53
54	36	6	6	6	ROL Z,X	54
55	37	7	7	7		55
56	38	8	8	8	SEC	56
57	39	9	9	9	AND Y	57
58	3A	:	:	:		58
59	3B	;	;	;		59
60	3C	<	<	<		60
61	3D	=	=	=	AND X	61
62	3E	>	>	>	ROL X	62
63	3F	?	?	?		63

DECIMAL	HEX	ASCII	SCREEN	BASIC	6502	DECIMAL
64	40	@	␣	@	RTI	64
65	41	A	␣,a	A	EOR(I,X)	65
66	42	B	␣,b	B		66
67	43	C	␣,c	C		67
68	44	D	␣,d	D		68
69	45	E	␣,e	E	EOR Z	69
70	46	F	␣,f	F	LSR Z	70
71	47	G	␣,g	G		71
72	48	H	␣,h	H	PHA	72
73	49	I	␣,i	I	EOR #	73
74	4A	J	␣,j	J	LSR A	74
75	4B	K	␣,k	K		75
76	4C	L	␣,l	L	JMP	76
77	4D	M	␣,m	M	EOR	77
78	4E	N	␣,n	N	LSR	78
79	4F	O	␣,o	O		79
80	50	P	␣,p	P	BVC	80
81	51	Q	␣,q	Q	EOR(I),Y	81
82	52	R	␣,r	R		82
83	53	S	␣,s	S		83
84	54	T	␣,t	T		84
85	55	U	␣,u	U	EOR Z,X	85
86	56	V	␣,v	V	LSR Z,X	86
87	57	W	␣,w	W		87
88	58	X	␣,x	X	CLI	88
89	59	Y	␣,y	Y	EOR Y	89
90	5A	Z	␣,z	Z		90
91	5B	[	␣,[	[		91
92	5C	\	␣,\	\		92
93	5D	]	␣,]	]	EOR X	93
94	5E	↑	␣,↑	↑	LSR X	94
95	5F	←	␣,←	←		95
96	60		␣		RTS	96
97	61		␣		ADC(I,X)	97
98	62		␣			98
99	63		␣			99
100	64		␣			100
101	65		␣		ADC Z	101
102	66		␣		ROR Z	102
103	67		␣			103
104	68		␣		PLA	104
105	69		␣,␣		ADC #	105
106	6A		␣		ROR A	106
107	6B		␣			107
108	6C		␣		JMP(I)	108
109	6D		␣		ADC	109
110	6E		␣		ROR	110
111	6F		␣			111
112	70		␣		BVS	112
113	71		␣		ADC(I),Y	113
114	72		␣			114
115	73		␣			115
116	74		␣			116
117	75		␣		ADC Z,X	117
118	76		␣		ROR Z,X	118
119	77		␣			119
120	78		␣		SEI	120
121	79		␣		ADC Y	121
122	7A		␣,␣			122
123	7B		␣			123
124	7C		␣			124
125	7D		␣		ADC X	125
126	7E		␣		ROR X	126
127	7F		␣			127

DECIMAL	HEX	ASCII	SCREEN	BASIC	6502	DECIMAL
128	80		@	END		128
129	81		A	FOR	STA(I,X)	129
130	82		B	NEXT		130
131	83	load & run	C	DATA		131
132	84		D	INPUT#	STY Z	132
133	85		E	INPUT	STA Z	133
134	86		F	DIM	STX Z	134
135	87	bell	G	READ		135
136	88		H	LET	DEY	136
137	89	set/clr tab	I	GOTO		137
138	8A		J	RUN	TXA	138
139	8B		K	IF		139
140	8C		L	RESTORE	STY	140
141	8D	car ret	M	GOSUB	STA	141
142	8E	graphics	N	RETURN	STX	142
143	8F	bot right	O	REM		143
144	90		P	STOP	BCC	144
145	91	cur up	Q	ON	STA(I),Y	145
146	92	rvs off	R	WAIT		146
147	93	clear	S	LOAD		147
148	94	insert	T	SAVE	STY Z,X	148
149	95	ins line	U	VERIFY	STA Z,X	149
150	96	ers end	V	DEF	STX Z,Y	150
151	97		W	POKE		151
152	98		X	PRINT#	TYA	152
153	99	scroll up	Y	PRINT	STA Y	153
154	9A		Z	CONT	TXS	154
155	9B	escape	[	LIST		155
156	9C		\	CLR		156
157	9D	cur left		CMD	STA X	157
158	9E		↑	SYS		158
159	9F		↓	OPEN		159
160	A0	□	█	CLOSE	LDY #	160
161	A1	□	█	GET	LDA(I,X)	161
162	A2	□	█	NEW	LDX #	162
163	A3	□	#	TAB(		163
164	A4	□	%	TO	LDY Z	164
165	A5	□	&	FN	LDA Z	165
166	A6	█	&	SPC(	LDX Z	166
167	A7	□		THEN		167
168	A8	□	(	NOT	TAY	168
169	A9	█, ▨	)	STEP	LDA #	169
170	AA	□	+	+	TAX	170
171	AB	□	-	-		171
172	AC	□	*	*	LDY	172
173	AD	□	/	/	LDA	173
174	AE	□	↑	↑	LDX	174
175	AF	□	/	AND		175
176	B0	□	0	OR	BCS	176
177	B1	□	1	>	LDA(I),Y	177
178	B2	□	2	=		178
179	B3	□	3	<		179
180	B4	□	4	SGN	LDY Z,X	180
181	B5	□	5	INT	LDA Z,X	181
182	B6	□	6	ABS	LDX Z,Y	182
183	B7	□	7	USR		183
184	B8	□	8	FRE	CLV	184
185	B9	□	9	POS	LDA Y	185
186	BA	□, ▨	.	SQR	TSX	186
187	BB	□	.	RND		187
188	BC	□	<	LOG	LDY X	188
189	BD	□		EXP	LDA X	189
190	BE	□	v	COS	LDX Y	190
191	BF	□	∇	SIN		191

DECIMAL	HEX	ASCII	SCREEN	BASIC	6502	DECIMAL
192	C0	▣		TAN	CPY #	192
193	C1	▣, a		ATN	CMP(I),X	193
194	C2	▣, b		PEEK		194
195	C3	▣, c		LEN		195
196	C4	▣, d		STR\$	CPY Z	196
197	C5	▣, e		VAL	CMP Z	197
198	C6	▣, f		ASC	DEC Z	198
199	C7	▣, g		CHR\$		199
200	C8	▣, h		LEFT\$	INY	200
201	C9	▣, i		RIGHT\$	CMP #	201
202	CA	▣, j		MID\$	DEX	202
203	CB	▣, k		GO		203
204	CC	▣, l		CONCAT	CPY	204
205	CD	▣, m		DOPEN	CMP	205
206	CE	▣, n		DCLOSE	DEC	206
207	CF	▣, o		RECORD		207
208	D0	▣, p		HEADER	BNE	208
209	D1	▣, q		COLLECT	CMP(I),Y	209
210	D2	▣, r		BACKUP		210
211	D3	▣, s		COPY		211
212	D4	▣, t		APPEND		212
213	D5	▣, u		DSAVE	CMP Z,X	213
214	D6	▣, v		DLOAD	DEC Z,X	214
215	D7	▣, w		CATALOG		215
216	D8	▣, x		RENAME	CLD	216
217	D9	▣, y		SCRATCH	CMP Y	217
218	DA	▣, z		DIRECTORY		218
219	DB	▣				219
220	DC	▣				220
221	DD	▣			CMP X	221
222	DE	▣, ▨			DEC X	222
223	DF	▣, ▨				223
224	E0	▣			CPX #	224
225	E1	▣			SBC(I),X	225
226	E2	▣				226
227	E3	▣				227
228	E4	▣			CPX Z	228
229	E5	▣			SBC Z	229
230	E6	▣			INC Z	230
231	E7	▣				231
232	E8	▣			INX	232
233	E9	▣, ▨			SBC #	233
234	EA	▣			NOP	234
235	EB	▣				235
236	EC	▣			CPX	236
237	ED	▣			SBC	237
238	EE	▣			INC	238
239	EF	▣				239
240	F0	▣			BEQ	240
241	F1	▣			SBC(I),Y	241
242	F2	▣				242
243	F3	▣				243
244	F4	▣				244
245	F5	▣			SBC Z,X	245
246	F6	▣			INC Z,X	246
247	F7	▣				247
248	F8	▣			SED	248
249	F9	▣			SBC Y	249
250	FA	▣, ▨				250
251	FB	▣				251
252	FC	▣				252
253	FD	▣			SBC X	253
254	FE	▣			INC X	254
255	FF	π				255

Reverse of ASCII

# BASIC 2.0 / BASIC 4.0 Memory Map

Supplied by Jim Butterfield. Reference to DOS, MLM, 80-Column, or those marked with an \* are for BASIC 4.0 only.

Hex	Dec	Description	Hex	Dec	Description
0000	0-002	USR jump	0097	151	Which key down; 255 = no key
0003	3	Search character	0098	152	Shift key: 1 if depressed
0004	4	Scan-between-quotes flag	0099	153-154	Correction clock
0005	5	Input buffer pointer; * of subscripts	009B	155	Keyswitch PIA: STOP and RVS flags
0006	6	Default DIM flag	009C	156	Timing constant for tape
0007	7	Type: FF = string, 00 = numeric	009D	157	Load=0, Verify=1
0008	8	Type: 80 = integer, 00 = floating point	009E	158	Number of characters in keybd buffer
0009	9	Flag: DATA scan: LIST quote; memory	009F	159	Screen reverse flag
000A	10	Subscript flag: FNx flag	00A0	160	IEEE output; 255 = character pending
000B	11	0 = INPUT; \$40 = GET; \$98 = READ	00A1	161	End-of-line-for-input pointer
000C	12	ATN sign/Comparison Evaluation flag	00A3	163-164	Cursor log (row, column)
000D	13-15	Disk status DSS descriptor	00A5	165	IEEE output buffer
0010	16	Current I/O device for prompt-suppress	00A6	166	Key image
0011	17-18	Integer value (for SYS, GOTO etc)	00A7	167	0 = flash cursor
0013	19-21	Pointers for descriptor stack	00A8	168	Cursor timing countdown
0016	22-30	Descriptor stack(temp strings)	00A9	169	Character under cursor
001F	31-34	Utility pointer area	00AA	170	Cursor in blink phase
0023	35-39	Product area for multiplication	00AB	171	EOT received from tape
0028	40-41	Pointer: Start of BASIC	00AC	172	Input from screen/from keyboard
002A	42-43	Pointer: Start of Variables	00AD	173	X save
002C	44-45	Pointer: Start of Arrays	00AE	174	How many open files
002E	46-47	Pointer: End of Arrays	00AF	175	Input device, normally 0
0030	48-49	Pointer: String Storage (moving down)	00B0	176	Output CMD device, normally 3
0032	50-51	Pointer: Utility String	00B1	177	Tape character parity
0034	52-53	Pointer: Limit of Memory	00B2	178	Byte received flag
0036	54-55	Current BASIC line number	00B3	179	Logical Address temporary save
0038	56-57	Previous BASIC line number	00B4	180	Tape buffer character/MLM command
003A	58-59	Pointer: BASIC statement for CONT	00B5	181	File name pointer/MLM flag, counter
003C	60-61	Current DATA line number	00B7	183	Serial bit count
003E	62-63	Current DATA address	00B9	185	Cycle counter
0040	64-65	Input vector	00BA	186	Tape writer countdown
0042	66-67	Current variable name	00BB	187-188	Tape buffer pointers, #1 and #2
0044	68-69	Current variable address	00BD	189	Write leader count; read pass1/2
0046	70-71	Variable pointer for FOR/NEXT	00BE	190	Write new byte; read error flag
0048	72-73	Y-save; op-save; BASIC pointer save	00BF	191	Write start bit; read bit seq error
004A	74	Comparison symbol accumulator	00C0	192-193	Error log pointers, pass1/2
004B	75-80	Misc work area, pointers, etc	00C2	194	0 = Scan/1-15 = Count/\$40 = Load/\$80 = End
0051	81-83	Jump vector for functions	00C3	195	Write leader length; read checksum
0054	84-93	Misc numeric work area	00C4	196-197	Pointer to screen line
005E	94	Accum#1: Exponent	00C6	198	Position of cursor on above line
005F	95-98	Accum#1: Mantissa	00C7	199-200	Utility pointer: tape, scroll
0063	99	Accum#1: Sign	00C9	201-202	Tape end addr/End of current program
0064	100	Series evaluation constant pointer	00CB	203-204	Tape timing constants
0065	101	Accum#1 hi-order (overflow)	00CD	205	0 = direct cursor, else programmed
0066	102-107	Accum#2: Exponent, etc.	00CE	206	Tape read timer 1 enabled
006C	108	Sign comparison, Acc#1 vs #2	00CF	207	EOT received from tape
006D	106	Accum#1 lo-order (rounding)	00D0	208	Read character error
006E	110-111	Cassette buff len/Series pointer	00D1	209	* characters in file name
0070	112-135	CHRGET subroutine; get BASIC char	00D2	210	Current file logical address
0077	119-120	BASIC pointer (within subrtn)	00D3	211	Current file secondary address
0088	136-140	Random number seed.	00D4	212	Current file device number
008D	141-143	Jiffy clock for TI and TIS	00D5	213	Right-hand window or line margin
0090	144-145	Hardware interrupt vector	00D6	214-215	Pointer: Start of Tape Buffer
0092	146-147	BRK interrupt vector	00D8	216	Line where cursor lives
0094	148-149	NMI interrupt vector	00D9	217	Last key/checksum/misc.
0096	150	Status word ST	00DA	218-219	File name pointer

00DC	220	Number of INSERTs outstanding	E810	Diagnostic Sense EOI In	IEEE Sense EOI In	Cassette Sense #1	Keyboard Row Select PA	59408
00DD	221	Write shift word/read character in	E811	Tape #1 Input Flag	EOI Out	CA2	DDRA Access	59409
00DE	222	Tape blocks remaining to write/read	E812	Keyboard Row Input				59410
00DF	223	Serial word buffer	E813	Retrace I Flag	Cassette #1 Motor Output I Flag	Motor Output CB2	DDRB Access	59411
00E0	224-248	(40-column) Screen line wrap table	E820	IEEE Input				59424
00E0	224-225	(80-column) Top, bottom of window	E821	ATN I Flag	IEEE NDAC Out	CA2	DDRA Access	59425
00E2	226	(80-column) Left window margin	E822	IEEE Output				59426
00E3	227	(80-column) Limit of keybd buffer	E823	SRQ I Flag	IEEE DAV Out	CB2	DDRB Access	59427
00E4	228	(80-column) Key repeat flag	E840	DAV In	NRFD In	Retrace In	Cass. #2 Motor	59456
00E5	229	(80-column) Repeat countdown	E841	Parallel User Port (PUP.) I/O with Handshake				59457
00E6	230	(80-column) New key marker	E842	Data Direction Register B (for E840)				59458
00E7	231	(80-column) Chime time	E843	Data Direction Register A (for E84F, PUP.)				59459
00E8	232	(80-column) HOME count	E844					L 59460
00E9	233-234	(80-column) Input vector	E845	Timer 1				H 59461
00EB	235-236	(80-column) Output vector	E846					L 59462
00FB	249-250	Cassette status, #1 and #2	E847	Timer 1 Latch				H 59463
00FD	251-252	Tape start address/MLM Pointer	E848					L 59464
00FF	253-254	MLM/DOS pointer/misc.	E849	Timer 2				H 59465
0100	255-266	STRs work area/MLM work	E84A	Shift Register				59466
0100	267-318	Tape read error log	E84B	T1 Control PB7 Out	T2 Ctrl PB6 Sense	Shift Register Control		59467
0100	319-331	Processor stack	E84C	CB2 (PUP. Pin M) In/Out	T1 In	CA2 (Graphics, Lower Case) In/Out	PB, PA Latch Control CA1 In	59468
0200	512-592	MLM work area: Input buffer	E84D	IRQ Status	T1 INT	T2 INT	CB1 Cass #2 INT	59469
0251	593-602	File logical address table	E84E	Enable Clear/Set	T1 INT	T2 INT	CB1 INT	59470
025B	603-612	File device number table	E84F	Parallel User Port I/O (PA)				59471
0265	613-622	File secondary addr table						
026F	623-632	Keyboard input buffer						
027A	634-825	Tape#1 input buffer						
033A	826-1017	Tape#2 input buffer						
033A	826	DOS character pointer						
033B	827	DOS drive 1 flag						
033C	828	DOS drive 2 flag						
033D	829	DOS length/write flag						
033E	830	DOS syntax flags						
033F	831-832	DOS disk ID						
0341	833	DOS command string count						
0342	834-850	DOS file name buffer						
0353	851-896	DOS command string buffer						
03EE	1006-1015	(80-column) Tab stop table						
03FA	1018-1019	Monitor extension vector						
03FC	1020	IEEE timeout defeat * SFF-disable						
0400	1024-32767	Available RAM including expansion						
8000	3276833791	(40-column) Video RAM						
8000	3276834815	(80-column) Video RAM						
9000	3686445055	Available ROM expansion area* (2.0: -BFFF, -49151)						
B000	DFFF	BASIC, DOS, Machine Lang Monitor (2.0 BASIC: C000-EDF8, 49152-57592)						
E000	E7FF	Screen, Keyboard, Interrupt programs (2.0: E0F9-)						
E810	E813	PIA 1 - Keyboard I/O						
E820	E823	PIA 2 - IEEE-488 I/O						
E840	E84F	VIA - I/O and timers						
E880	E881	(80-column) CRT Controller						
F000	FFFF	Reset, I/O handlers, Tape routines						

The Complete Commodore Inner Space Anthology

Memory Maps: PET/CBM







VIC 20 Memory Map

Main VIC 20 Memory Map table with columns for address, description, and data. Includes entries like 0000 -0002 USR jump, 0003 -0004 Fixed-Float vector, etc.

VIC 20 ROM Routines table with columns for address, description, and data. Includes entries like C000 ROM control vectors, C00C Keyboard action vectors, etc.



### VIC 20 Standard Configuration

FFFF	8K Kernal ROM	65535
E000	8K BASIC ROM	57344
C000		49152
A000		40960
95FF	Colour Nybble Area	38399
9600	VIC Chip & I/O	38400
9000	Character Set	36864
8000		32768
2000	1/2K Screen RAM from basic VIC 20	4096
1E00	3 1/2 K RAM for BASIC	7680
1000		4096
0400	1K RAM Work Space	1024
0000		0

### VIC 20 Expansion RAM Memory Changes

Exp RAM at:	BASIC Text	Screen	Colour Table
none	4096 / \$1000	7680 / \$1E00	38400 / \$9600
1024 / 4095*	1024 / \$0400	7680 / \$1E00	38400 / \$9600
8192 and up	4608 / \$1200	4096 / \$1000	37888 / \$9400

\* VIC 1210 3K RAM Expander

### VIC 20 With 40K RAM

- VIC 1020 Expansion Module Required with:
- 1 - VIC 1210 3K RAM
  - 2 - VIC 1110 8K RAM (Switches 2,3,4 down - Switch 1 up)
  - 3 - VIC 1110 8K RAM (Switches 1,3,4 down - Switch 2 up)
  - 4 - VIC 1111 16K RAM

FFFF	8K Kernal ROM	65535
E000	8K BASIC ROM	57344
C000		49152
A000	VIC 1110 8K RAM (2) (usable only with PEEK, POKE & M/L)	40960
95FF	Colour Nybble Area	38399
9400	VIC Chip & I/O	37888
9000	Character Set	36864
8000		32768
	VIC 1110 8K RAM (3)	
	VIC 1111 16K RAM (4)	27 1/2 K for BASIC
	3 1/2 K of RAM from basic VIC 20	
1200	1/2K Screen RAM from basic VIC 20	4608
1000	VIC 1210 3K RAM (1) (usable only with PEEK, POKE & M/L)	4096
0400	1K RAM Work Space	1024
0000		0

### 6560 VIC Chip

9000	Interlace	Left Margin (= 5)	36864	
9001		Top Margin (= 25)	36865	
9002	Screen Ad Bit 9	Number of Columns (= 22)	36866	
9003	Bit 0	Number of Rows (= 23) Double Char	36867	
9004		Input Raster Value: Bits 1-8	36868	
9005	Screen Address Bits 13-10	Character Address Bits 13-10	36869	
9006	Light Pen Input	Horizontal	36870	
9007		Vertical	36871	
9008		X	36872	
9009	Paddle Input	Y	36873	
900A	ON	Voice 1 Frequency	36874	
900B	ON	Voice 2 Frequency	36875	
900C	ON	Voice 3 Frequency	36876	
900D	ON	Noise Frequency	36877	
900E		Multi Colour Mode	Sound Amplitude	36878
900F	Background Colour	Foregnd/Backgnd	Border Colour	36879

### 6522 VIA 1

9110	DSR In	CTS In	DCD* In	RI* In	DTR Out	RTS Out	Data In	37136	
	RS-232 Interface or Parallel User Port								
9111	*Unused - see \$911F							37137	
9112	Data Direction Register B (for \$9110)							37138	
9113	Data Direction Register A (for \$911F)							37139	
9114	T1-L	RS 232 Send Speed:						37140	
9115	T1-H	Tape Write Timing						37141	
9116	T1-Latch L							37142	
9117	T1 Latch H							37143	
9118	T2-L	RS 232 Input Timing						37144	
9119	T2-H							37145	
911A	Shift Register (* unused)							37146	
911B	T1 Control	T2 Ctrl	Shift Register Control		PB LE	PA LE	37147		
911C	CB2: RS 232 Send		CB1 Ctrl	CA2: Tape Motor Ctrl		CA1 Ctrl	37148		
911D	NMI:	T1	T2	CB1: RS 232 In		CA1: RESTORE	37149		
911E	NMI En.	T1 Enab	T2 Enab	CB1 En.		CA1 En.	37150		
911F	ATN Out	Tape Sense	Fire	Joystick Left	Down	Up	Serial Data In	Serial Clock In	37151

### 6522 VIA 2

9120	Joystick Right	Tape Out	Keyboard Row Select	37152			
9121	Keyboard Column Input			37153			
9122	Data Direction Register B (for \$9120)			37154			
9123	Data Direction Register A (for \$9121)			37155			
9124	T1-L	Cassette Tape Read:		37156			
9125	T1-H	Keyboard and Clock		37157			
9126	T1-Latch L	Interrupt Timing		37158			
9127	T1 Latch H			37159			
9128	T2-L	Serial Bus Timing		37160			
9129	T2-H	Tape R/W Timing		37161			
912A	Shift Register (* unused)			37162			
912B	T1 Control	T2 Ctrl	Shift Register Control	PB LE	PA LE	37163	
912C	Serial Bus Data Out		CB1 Ctrl	Serial Clock Line Out		CA1 Ctrl	37164
912D	IRQ:	T1	T2	CB1: SRQ In		CA1: Tape In	37165
912E	IRQ En.	T1 Enab	T2 Enab	CB1 En.		CA1 En.	37166
912F	*Unused (see \$9121)			37167			

# SuperChart: VIC 20 / Commodore 64

DECIMAL	HEX	ASCII	SCREEN	BASIC	6502	DECIMAL
0	00		@	end-line	BRK	0
1	01		A		ORA(I,X)	1
2	02		B			2
3	03	stop	C			3
4	04		D			4
5	05	white	E		ORA Z	5
6	06		F		ASL Z	6
7	07		G			7
8	08	lock	H		PHP	8
9	09	unlock	I		ORA #	9
10	0A		J		ASL A	10
11	0B		K			11
12	0C		L			12
13	0D	car ret	M		ORA	13
14	0E	text	N		ASL	14
15	0F		O			15
16	10		P		BPL	16
17	11	cur down	Q		ORA(I),Y	17
18	12	reverse	R			18
19	13	cur home	S			19
20	14	delete	T			20
21	15		U		ORA Z,X	21
22	16		V		ASL Z,X	22
23	17		W			23
24	18		X		CLC	24
25	19		Y		ORA Y	25
26	1A		Z			26
27	1B		[			27
28	1C	red	\			28
29	1D	cur right	]		ORA X	29
30	1E	green	↑		ASL X	30
31	1F	blue	←			31
32	20	space	space	space	JSR	32
33	21	!	!	!	AND(I,X)	33
34	22					34
35	23	#	#	#		35
36	24	\$	\$	\$	BIT Z	36
37	25	%	%	%	AND Z	37
38	26	&	&	&	ROL Z	38
39	27	'	'	'		39
40	28	(	(	(	PLP	40
41	29	)	)	)	AND #	41
42	2A	*	*	*	ROL A	42
43	2B	+	+	+		43
44	2C	,	,	,	BIT	44
45	2D	-	-	-	AND	45
46	2E	.	.	.	ROL	46
47	2F	/	/	/		47
48	30	0	0	0	BMI	48
49	31	1	1	1	AND(I),Y	49
50	32	2	2	2		50
51	33	3	3	3		51
52	34	4	4	4		52
53	35	5	5	5	AND Z,X	53
54	36	6	6	6	ROL Z,X	54
55	37	7	7	7		55
56	38	8	8	8	SEC	56
57	39	9	9	9	AND Y	57
58	3A	:	:	:		58
59	3B	;	;	;		59
60	3C	<	<	<		60
61	3D	=	=	=	AND X	61
62	3E	>	>	>	ROL X	62
63	3F	?	?	?		63

DECIMAL	HEX	ASCII	SCREEN	BASIC	6502	DECIMAL
64	40	@	␣	@	RTI	64
65	41	A	␣,a	A	EOR(I,X)	65
66	42	B	␣,b	B		66
67	43	C	␣,c	C		67
68	44	D	␣,d	D		68
69	45	E	␣,e	E	EOR Z	69
70	46	F	␣,f	F	LSR Z	70
71	47	G	␣,g	G		71
72	48	H	␣,h	H	PHA	72
73	49	I	␣,i	I	EOR #	73
74	4A	J	␣,j	J	LSR A	74
75	4B	K	␣,k	K		75
76	4C	L	␣,l	L	JMP	76
77	4D	M	␣,m	M	EOR	77
78	4E	N	␣,n	N	LSR	78
79	4F	O	␣,o	O		79
80	50	P	␣,p	P	BVC	80
81	51	Q	␣,q	Q	EOR(I),Y	81
82	52	R	␣,r	R		82
83	53	S	␣,s	S		83
84	54	T	␣,t	T		84
85	55	U	␣,u	U	EOR Z,X	85
86	56	V	␣,v	V	LSR Z,X	86
87	57	W	␣,w	W		87
88	58	X	␣,x	X	CLI	88
89	59	Y	␣,y	Y	EOR Y	89
90	5A	Z	␣,z	Z		90
91	5B	[	␣	[		91
92	5C	⌞	␣	⌞		92
93	5D	]	␣	]	EOR X	93
94	5E	↑	␣	↑	LSR X	94
95	5F	←	␣	←		95
96	60		␣		RTS	96
97	61		␣		ADC(I,X)	97
98	62		␣			98
99	63		␣			99
100	64		␣			100
101	65		␣		ADC Z	101
102	66		␣		ROR Z	102
103	67		␣			103
104	68		␣		PLA	104
105	69		␣		ADC #	105
106	6A		␣		ROR A	106
107	6B		␣			107
108	6C		␣		JMP(I)	108
109	6D		␣		ADC	109
110	6E		␣		ROR	110
111	6F		␣			111
112	70		␣		BVS	112
113	71		␣		ADC(I),Y	113
114	72		␣			114
115	73		␣			115
116	74		␣			116
117	75		␣		ADC Z,X	117
118	76		␣		ROR Z,X	118
119	77		␣			119
120	78		␣		SEI	120
121	79		␣		ADC Y	121
122	7A		␣			122
123	7B		␣			123
124	7C		␣			124
125	7D		␣		ADC X	125
126	7E		␣		ROR X	126
127	7F		␣			127

DECIMAL	HEX	ASCII	SCREEN	BASIC	6502	DECIMAL
128	80		@	END		128
129	81	orange	A	FOR	STA(I,X)	129
130	82		B	NEXT		130
131	83	load & run	C	DATA		131
132	84		D	INPUT#	STY Z	132
133	85	F1	E	INPUT	STA Z	133
134	86	F3	F	DIM	STX Z	134
135	87	F5	G	READ		135
136	88	F7	H	LET	DEY	136
137	89	F2	I	GOTO		137
138	8A	F4	J	RUN	TXA	138
139	8B	F6	K	IF		139
140	8C	F8	L	RESTORE	STY	140
141	8D	car ret	M	GOSUB	STA	141
142	8E	graphics	N	RETURN	STX	142
143	8F		O	REM		143
144	90	black	P	STOP	BCC	144
145	91	cur up	Q	ON	STA(I),Y	145
146	92	rvs off	R	WAIT		146
147	93	clear	S	LOAD		147
148	94	insert	T	SAVE	STY Z,X	148
149	95	brown	U	VERIFY	STA Z,X	149
150	96	lt. red	V	DEF	STX Z,Y	150
151	97	dk. grey	W	POKE		151
152	98	md. grey	X	PRINT#	TYA	152
153	99	lt. green	Y	PRINT	STA Y	153
154	9A	lt. blue	Z	CONT	TXS	154
155	9B	lt. grey	[	LIST		155
156	9C	magenta	]	CLR		156
157	9D	cur left	!	CMD	STA X	157
158	9E	yellow	!	SYS		158
159	9F	cyan	!	OPEN		159
160	A0		!	CLOSE	LDY #	160
161	A1		!	GET	LDA(I,X)	161
162	A2		!	NEW	LDX #	162
163	A3		#	TAB(		163
164	A4		S	TO	LDY Z	164
165	A5		%	FN	LDA Z	165
166	A6		&	SPC(	LDX Z	166
167	A7		(	THEN		167
168	A8		)	NOT	TAY	168
169	A9		)	STEP	LDA #	169
170	AA		+	+	TAX	170
171	AB		+	-		171
172	AC		*	*	LDY	172
173	AD		/	/	LDA	173
174	AE		/	↑	LDX	174
175	AF		/	AND		175
176	B0		0	OR	BCS	176
177	B1		1	>	LDA(I),Y	177
178	B2		2	=		178
179	B3		3	<		179
180	B4		4	SGN	LDY Z,X	180
181	B5		5	INT	LDA Z,X	181
182	B6		6	ABS	LDX Z,Y	182
183	B7		7	USR		183
184	B8		8	FRE	CLV	184
185	B9		9	POS	LDA Y	185
186	BA		:	SQR	TSX	186
187	BB		:	RND		187
188	BC		<	LOG	LDY X	188
189	BD		=	EXP	LDA X	189
190	BE		>	COS	LDX Y	190
191	BF		?	SIN		191

DECIMAL	HEX	ASCII	SCREEN	BASIC	6502	DECIMAL
192	C0		␣	TAN	CPY #	192
193	C1	a	␣,a	ATN	CMP(I),X	193
194	C2	b	␣,b	PEEK		194
195	C3	c	␣,c	LEN		195
196	C4	d	␣,d	STR\$	CPY Z	196
197	C5	e	␣,e	VAL	CMP Z	197
198	C6	f	␣,f	ASC	DEC Z	198
199	C7	g	␣,g	CHR\$		199
200	C8	h	␣,h	LEFT\$	INY	200
201	C9	i	␣,i	RIGHT\$	CMP #	201
202	CA	j	␣,j	MID\$	DEX	202
203	CB	k	␣,k	GO		203
204	CC	l	␣,l		CPY	204
205	CD	m	␣,m		CMP	205
206	CE	n	␣,n		DEC	206
207	CF	o	␣,o			207
208	D0	p	␣,p		BNE	208
209	D1	q	␣,q		CMP(I),Y	209
210	D2	r	␣,r			210
211	D3	s	␣,s			211
212	D4	t	␣,t			212
213	D5	u	␣,u		CMP Z,X	213
214	D6	v	␣,v		DEC Z,X	214
215	D7	w	␣,w			215
216	D8	x	␣,x		CLD	216
217	D9	y	␣,y		CMP Y	217
218	DA	z	␣,z			218
219	DB		␣			219
220	DC		␣			220
221	DD		␣		CMP X	221
222	DE		␣		DEC X	222
223	DF		␣			223
224	E0		␣		CPX #	224
225	E1		␣		SBC(I),X	225
226	E2		␣			226
227	E3		␣			227
228	E4		␣		CPX Z	228
229	E5		␣		SBC Z	229
230	E6		␣		INC Z	230
231	E7		␣			231
232	E8		␣		INX	232
233	E9		␣		SBC #	233
234	EA		␣		NOP	234
235	EB		␣			235
236	EC		␣		CPX	236
237	ED		␣		SBC	237
238	EE		␣		INC	238
239	EF		␣			239
240	F0		␣		BEQ	240
241	F1		␣		SBC(I),Y	241
242	F2		␣			242
243	F3		␣			243
244	F4		␣			244
245	F5		␣		SBC Z,X	245
246	F6		␣		INC Z,X	246
247	F7		␣			247
248	F8		␣		SED	248
249	F9		␣		SBC Y	249
250	FA		␣			250
251	FB		␣			251
252	FC		␣			252
253	FD		␣		SBC X	253
254	FE		␣		INC X	254
255	FF		␣			255

Reverse of ASCII



### 6566 Video Chip C64 Control & Miscellaneous Registers

D011	Extended Clr. Mode	Bit Map	Display Enable	Row Select	Y-Scroll	53265			
D012	Raster Register					53266			
D013	Light Pen Input					53267			
D014	Light Pen Input					53268			
D016	x	x	Reset	Multi Colour	Column Select	X-Scroll	53270		
D018	VM13	VM12	VM11	VM10	Screen	Character Base CB13 CB12 CB11	x	53272	
D019	IRQ	Interrupt Sense:			Light Pen	Spr-Spr Collision	Spr-Back Collision	Raster	53273
D01A	Interrupt Enable:			Light Pen	Spr-Spr Collisions	Spr-Back Collisions	Raster	53274	
<b>Colour Registers</b>									
D020	X	Exterior Colour (Border)				53280			
D021	X	Background Colour #0				53281			
D022	X	Background Colour #1				53282			
D023	X	Background Colour #2				53283			
D024	X	Background Colour #3				53284			
D025	X	Sprite MultiColour #0				53285			
D026	X	Sprite MultiColour #1				53286			

### 6566 Video Chip C64 Sprite Registers

Sprite 0	Sprite 7	Sprite 0	Sprite 7
D000	D00E	X Position	53248
D001	D00F	Y Position	53249
D027	D02E	Sprite Colour	53287

Bit For Sprite*:								
7	6	5	4	3	2	1	0	
D010	X-Position High							53264
D015	Sprite Enable Flags							53269
D017	Y-Expand							53271
D01B	Background Priority							53275
D01C	Sprite MultiColour Mode							53276
D01D	X-Expand							53277
D01E	Interrupt: Sprite Collision							53278
D01F	Interrupt: Background Collision							53279

### CIA 1 (IRQ) (6526)

SDC00	Paddle Sel A B	Fire	Right	Joystick 0 Left	Down	Up	PRA	56320	
Keyboard Row Select (inverted)									
SDC01	Fire	Right	Joystick 1 Left	Down	Up	PRB	56321		
Keyboard Column Read									
SDC02	\$FF - All Output							DDRA	56322
SDC03	\$00 - All Input							DDRB	56323
SDC04	Timer A							TAL	56324
SDC05	Timer A							TAH	56325
SDC06	Timer B							TBL	56326
SDC07	Timer B							TBH	56327
SDC0D	Tape Input	Timer Interrupt A				ICR	56333		
SDC0E	One Shot	Out Mode	Out Mode	PB6 Out	Timer A Start	CRA	56334		
SDC0F	One Shot	Out Mode	Out Mode	PB7 Out	Timer B Start	CRB	56335		

### Processor I/O Port (6510)

\$0000	IN	IN	OUT	IN	OUT	OUT	OUT	OUT	DDR 0
\$0001	Tape Motor		Tape Sense	Tape Write	D-ROM Switch	EF RAM Switch	AB RAM Switch	PR 1	

### SID (6581)

Voice 1	Voice 2	Voice 3	Frequency	L	Voice 1	Voice 2	Voice 3
\$D400	\$D407	\$D40E		H	54272	54279	54286
\$D401	\$D408	\$D40F	Pulse Width	L	54273	54280	54287
\$D402	\$D409	\$D410	0 0 0 0	H	54274	54281	54288
\$D403	\$D40A	\$D411	Voice Type: NSE, PUL, SAW, TRI	Key	54275	54282	54289
\$D404	\$D40B	\$D412	Attack Time 2ms - 8ms	Decay Time 6ms - 24 sec	54276	54283	54290
\$D405	\$D40C	\$D413	Sustain Level	Release Time 6ms - 24 sec	54277	54284	54291
\$D406	\$D40D	\$D414	Voices (write only)				

### CIA 2 (NMI) (6526)

SDD00	Serial IN	Clock IN	Serial OUT	Clock OUT	ATN OUT	RS-232 OUT	VIC II addr 15	VIC II addr 14	PRA	56576	
SDD01	DSR IN	CTS IN	DCD* IN	RI* IN	DTR OUT	RTS OUT	RS-232 IN	PRB	56577		
SDD02	\$3F - Serial								DDRA	56578	
SDD03	\$00 - P.U.P. All Input				or	\$06 - RS-232				DDRB	56579
SDD04	Timer A								TAL	56580	
SDD05	Timer A								TAH	56581	
SDD06	Timer B								TBL	56582	
SDD07	Timer B								TBH	56583	
SDD0D	RS-232 IN				Timer Interrupt B				ICR	56589	
SDD0E					Timer A Start				CRA	56590	
SDD0F					Timer B Start				CRB	56591	

\$D415	0 0 0 0 0	L	54293	
\$D416	Filter Frequency		H	54294
\$D417	Resonance	Filter Voices V3 V2 V1	Ext	54295
\$D418	Passband: V3 off, HI, BP, LO	Master Volume		54296
Filter & Volume (write only)				

\$D419	Paddle X (A/D #1)	54297
\$D41A	Paddle Y (A/D #2)	54298
\$D41B	Noise 3 (random)	54299
\$D41C	Envelope 3	54300
Sense (read only)		

Note: Special Voice Features (TEST, RING MOD, SYNC) are omitted from the above diagram.

\* Connected but not used by ( ).S.









### Commodore B128 ROM Routines

The following is a map of routines and data within the current (September 1983) version of the Commodore B128 computer. Caution: The same routines exist in the B256 but the addresses are not exactly the same.

Table listing Commodore B128 ROM routines with columns for address, routine name, and description. Includes routines like 'bad subscript', 'illegal quantity', 'Float-fixed conversion', 'Get prog key addr', etc.

### 6526 CIA 1

Table for CIA 1 registers DB00-DB0F. Columns include register address, bits (X, IRQ Out), and data type (SEMAPH, Busy, Unused, IP Flag).

### 6526 CIA 2

Table for CIA 2 registers DC00-DC0F. Columns include register address, data type (IEEE Data In/Out, User Port, Data Direction Register, Unused, Timer B, Time Of Day Clock, Alarm, Unused), and units (Sec, Min, Hour).

### 6551 ACIA

Table for ACIA registers DD00-DD03. Columns include register address, data type (Data Register, XTRR Stop, Parity), and bits (IRQ, DSR, DCD, Tx, Rx, OV, Error FR, PA, Speed, Tx, Rx, DTR).

### 6545 CRT Controller

Table for CRT Controller registers D800-D817. Columns include register address, register name (Horizontal Total, Horizontal Char Displayed, etc.), and typical value in decimal.

Most Register are Write Only 14/15 are Read/Write  
16/17 are Read Only  
Registers 10, 14 and 15 change as the cursor moves

### 6581 SID

Table for SID registers DA01-DA18. Columns include register address, data type (Voice 1 Frequency High, Saw Tooth, Ring Mod, Key, Attack, Decay, Release, Voice 3 Modulating Freq Hi, Volume), and register value.





## Commodore Disk Specifications

Model	D9090	D9060	8250	8050	4040	2031	1541
Drives per Unit	1	1	2	2	2	1	1
Heads per Drive	6	4	2	1	1	1	1
<b>Formatted Storage</b>							
Capacity per Unit	7.47 MB	4.98 MB	2.12 MB	1.05 MB	340 KB	170 KB	170 KB
Max. Sequential Files/Drive	7.41 MB	4.94 MB	1.05 MB	521 KB	168 KB	168 KB	168 KB
Max. Relative Files/Drive	7.35 MB	4.90 MB	1.04 MB	183 KB	167 KB	167 KB	167 KB
Disk System Buffer	4 KB	4 KB	4 KB	4 KB	4 KB	2 KB	2 KB
<b>Disk Formats</b>							
Cylinders (Tracks)	153	153	154	77	35	35	35
Sectors per Cylinder	128	192	-	-	-	-	-
Sectors per Track	32	32	23-29	23-29	17-21	17-21	17-21
Bytes per Sector	256	256	256	256	256	256	256
Blocks Free/Unit	29162	19442	8266	4104	1328	664	664
<b>Transfer Rates (bytes per second)</b>							
Internal to Unit	5 MB	5 MB	40 KB	40 KB	40 KB	40 KB	-
IEEE-488 Bus	1.2 KB	1.2 KB	1.2 KB	1.2 KB	1.2 KB	1.2 KB	-
<b>Access Times (milli-seconds)</b>							
Track-to-Track	3	3	5	*	30	30	30
Average Track	153	153	125	**	360	360	360
Head Settling Time	15	15	-	-	-	-	-
Average Latency	8.34	8.34	100	100	100	100	100
RPM	3600	3600	300	300	300	300	300
* Track-to-Track: Micropolis 8050 = 30 ms. Tandon 8050 = 5 ms. ** Average Track : Micropolis 8050 = 750 ms. Tandon 8050 = 125 ms.							
<b>Physical Dimensions</b>							
Height (inches)	5.75	5.75	7.0	7.0	7.0	5.5	3.0
Width (inches)	8.25	8.25	15.0	15.0	15.0	8.0	7.0
Depth (inches)	15.25	15.25	13.75	13.75	13.75	14.25	13.0
Weight (pounds)	21	21	28	28	28	20	10
<b>Electrical</b>							
Power (watts)	200	200	60	50	50	40	35
Voltage (all models)	110 - 120 VAC. 60 Hz						

### Directory-File Header Format

4040, 2031, 1541 Directory Header - Track 18 Sector 00		
Byte#	Data	Description
1-143		Reserved for 4040/2031/1541 BAM
144-161		Diskette name, padded with shifted spaces
162-163		Diskette ID number
164	160	Shifted space
165-166	50, 65	ASCII '2a' identifies DOS version and format
167-170	160	Shifted spaces
171-255	00	Not used
8050, 8250 Directory Header - Track 39 Sector 00		
Byte#	Data	Description
0-1	38, 00	Track/Sector to first BAM block
2	67	ASCII 'c' identifies DOS 2.5 format
3	00	Reserved for future DOS use
4-5		Not used
6-21		Diskette Name, padded with shifted spaces
22-23	160	Shifted spaces
24-25		Diskette ID number
26	160	Shifted space
27-28	50, 67	ASCII '2c' identifies DOS version and format
29-32	160	Shifted spaces
33-255	00	Not used
D9060 / D9090 Directory Header - Track 00 Sector 00		
Byte	Data	Description
0-1		Track/Sector pointer to bad Track and Sector list
2-3	00,255	Identifies DOS 3.0 format
4-5	76, 00	Track/Sector of first Directory block
6-7	00, 00	Not used
8-9	01, 00	Track/Sector of first BAM block

### Directory-File Sector Format

2031 Directory Blocks - Track 18 Sector 01 through 18	
4040 Directory Blocks - Track 18 Sector 01 through 18	
8050 Directory Blocks - Track 39 Sector 01 through 29	
8250 Directory Blocks - Track 39 Sector 01 through 29	
D9060 / D9090 Directory Blocks - Starting on cylinder 76, uses all Tracks Sectors 00 through 31, then expands to additional blocks as needed, providing 'unlimited' Directory size.	
Byte#	Description
0-1	Track/Sector pointer to next Directory block
2	File type
3-4	Track/Sector pointer to first file block
5-20	File name, padded with shifted spaces
21-22	Track/Sector of first side sector if RELative file
23	Record length if relative file
24-27	Reserved for future file information
28-29	Track/Sector pointer for replacement
30-31	Number of blocks used by the file
32-255	Seven more 32-byte file entries (same as 2-31 above, plus two additional unused bytes)
Additional Notes	
1	32 bytes per file entry, except the first entry is 30 bytes
2	Total of eight (8) file entries per Directory block
3	File types are: <ul style="list-style-type: none"> <li>Scratched Files \$00</li> <li>SEQ Sequential Files \$01</li> <li>PRG Program Files \$02</li> <li>USR User-Defined \$03</li> <li>REL Relative Record \$04</li> </ul>
4	File type codes are OR'ed with \$80 when file is properly closed
5	Track value of 00 in byte zero indicates the last used block in the Directory. Sector value then shows next byte to use.

# BAM (Block Allocation Map) Formats

4040, 2031, 1541 BAM Format - Track 18 Sector 00				
Byte#	Description	Data		
0-1	Track/Sector of first Directory block	18-01		
2	ASCII 'a' Identifies DOS 2.0 format	65		
3	Reserved for future DOS use	00		
4-143	BAM : Each Track Controlled By 4 bytes	tracks 1-35		
4	Byte 0: Total Blocks Free In Track:	track 1:		
5	Byte 1: Bit Map Of Sector Allocation	sectors 0-7		
6	Byte 2: Bit Map Of Sector Allocation	sectors 8-15		
7	Byte 3: Bit Map Of Sector Allocation	sectors 16-end		
	A bit ON = 1 represents a FREE Sector A bit OFF = 0 represents an Allocated Sector			
8-143	4 Byte Track Maps repeat for all tracks	tracks 2-35		
144-255	Unused			
180-191	Note: 'BLOCKS FREE nnn' may appear here. Not used.			

8050 BAM Format				
Byte#	Description	Data		
		BAM 1 Tr38 / Sc00	BAM 2 Tr38 / Sc03	
0-1	Track/Sector of next BAM block	38-03	39-01	
2	ASCII 'c' identifies DOS 2.5 format	67	67	
3	Reserved for future DOS use	00	00	
4	Lowest track # mapped in this BAM block	01	51	
5	Highest Track # (+ 1) mapped in this BAM block	51	78	
6-255	BAM : Each Track Controlled By 5 bytes	tracks 1-50	tracks 51-77	
6	Byte 0: Total Blocks Free In Track:	track 1:	track 51:	
7	Byte 1: Bit Map Of Sector Allocation	sectors 0-7	sectors 0-7	
8	Byte 2: Bit Map Of Sector Allocation	sectors 8-15	sectors 8-15	
9	Byte 3: Bit Map Of Sector Allocation	sectors 16-23	sectors 16-23	
10	Byte 4: Bit Map Of Sector Allocation	sectors 24-end	sectors 24-end	
	A bit ON = 1 represents a FREE Sector A bit OFF = 0 represents an Allocated Sector			
11-255	(BAM 2: 11-140) 5 Byte Track Maps repeat for all tracks	tracks 2-50	tracks 52-77	
180-191	Note: 'BLOCKS FREE nnn' may appear here on BAM 2. Not used.			

8250 BAM Format					
Byte#	Description	Data			
		BAM 1 Tr38 / Sc00	BAM 2 Tr38 / Sc03	BAM 3 Tr38 / Sc06	BAM 4 Tr38 / Sc09
0-1	Track/Sector of next BAM block	38-03	38-06	38-09	39-01 (Dir)
2	ASCII 'c' identifies DOS 2.7 format	67	67	67	67
3	Reserved for future DOS use	00	00	00	00
4	Lowest Track # mapped in this BAM block	01	51	101	151
5	Highest Track # (+ 1) mapped in this BAM block	51	101	151	155
6-255	BAM : Each Track Controlled By 5 bytes	tracks 1-50	tracks 51-100	tracks 101-150	tracks 151-154
6	Byte 0: Total Blocks Free In Track:	track 1:	track 51:	track 101:	track 151:
7	Byte 1: Bit Map Of Sector Allocation	sectors 0-7	sectors 0-7	sectors 0-7	sectors 0-7
8	Byte 2: Bit Map Of Sector Allocation	sectors 8-15	sectors 8-15	sectors 8-15	sectors 8-15
9	Byte 3: Bit Map Of Sector Allocation	sectors 16-23	sectors 16-23	sectors 16-23	sectors 16-23
10	Byte 4: Bit Map Of Sector Allocation	sectors 24-end	sectors 24-end	sectors 24-end	sectors 24-end
	A bit ON = 1 represents a FREE Sector A bit OFF = 0 represents an Allocated Sector				
11-255	(BAM 4: 11-25) 5 Byte Track Maps repeat for all tracks	tracks 2-50	tracks 52-100	tracks 102-150	tracks 152-154
180-191	Note: 'BLOCKS FREE nnn' may appear here on BAM 4. Not used.				

D9060 / D9090 BAM Format - Track 1 Sector 0 (normal location)				
Byte#	Description	Data		
0-1	Track/Sector pointer to next BAM block	\$FFFF = last		
2-3	Track/Sector pointer to previous BAM block	\$FFFF = first		
4	Lowest Track # mapped in this BAM block			
5	Highest Track # (+ 1) mapped in this BAM block			
6	Number of blocks unused on this Track			
7-10	Bit Map of available blocks on this Track			
11-255	Bit Map of the next 49 Tracks			

## Disk Sector Recording Format

SYNC	08	ID <sub>1</sub>	ID <sub>2</sub>	Track #	Sector #	Checksum	Gap 1	SYNC	07	Next Track	Next Sector	254 Bytes of Data	Checksum	Gap 2
------	----	-----------------	-----------------	---------	----------	----------	-------	------	----	------------	-------------	-------------------	----------	-------

# Disk Data File Format

Program Files	
Byte#	Description
0-1	Track/Sector pointer to next Program block
2-255	Up to 254 bytes of BASIC Program text. End-of-File is marked by three consecutive bytes of \$00
Sequential and Relative Record Data	
Byte#	Description
0-1	Track/Sector pointer to next sequential data block
2-255	Up to 254 bytes of data
<b>Notes:</b> Track link of \$00 in byte zero indicates last data block (Track 0 is not used by DOS). Sector link is then next byte position to receive data. End of relative record data indicated by ST = 64. Unused Record bytes are padded with CHR\$(0). Relative File terminated with \$FF.	
Relative File Side Sector Format	
Byte#	Description
0-1	Track/Sector pointer to next Side Sector
2	8050/4040/2031/1541: Side Sector number
	5250/D9060/D9090: constant \$FE
3	Relative Record Length
4-5	Track/Sector pointer - First Side Sector
6-7	Track/Sector pointer - Second Side Sector
8-9	Track/Sector pointer - Third Side Sector
10-11	Track/Sector pointer - Fourth Side Sector
12-13	Track/Sector pointer - Fifth Side Sector
14-15	Track/Sector pointer - Sixth Side Sector
16-255	Track/Sector pointers to 120 data blocks. Total of 720 blocks (maximum 182.8 K Bytes) per file
DOS 2.7 and DOS 3.0 Super Side Sector contain Track/Sector pointers to 127 groups of 6 Side Sectors as above for maximum file size of 23.25 MB.	

## Disk Utility-Command Set

Command	Abbreviations	Format
Block-Read	B-R	"B-R:" lf;dr;t;s
Block-Write	B-W	"B-W:" lf;dr;t;s
Block-Execute	B-E	"B-E:" lf;dr;t;s
Buffer-Pointer	B-P	"B-P:" lf;p
Block-Allocate	B-A	"B-A:" dr;t;s
Block-Free	B-F	"B-F:" dr;t;s
Memory-Write	M-W	"M-W" adl/adh/nc/data
Memory-Read	M-R	"M-R" adl/adh/nc
Memory-Execute	M-E	"M-E" adl/adh
User	U	"Ux:" lf;dr;t;s

LF	The Logical File Number in the associated OPEN Statement
DR	The Drive Number: 0 (or 1 on dual drives)
T	The Track Number: 1 through 154 (depending on the model number)
S	The Sector Number: 0 through 192 (depending on the model number)
P	The pointer Position for the Buffer Pointer
ADL	The Low Byte of the Address (use CHR\$(ADL))
ADH	The High Byte of the Address (use CHR\$(ADH))
NC	The Number of Characters: 1 through 34
DATA	The actual data in hexadecimal. this is transmitted by using the CHR\$(17) would send the decimal equivalent of hex 11
X	The index to the user table

## Disk LED Error Diagnostics

Number of Flashes	4040		8050	
	Error Cause	Component, Location	Error Cause	Component, Location
1	Zero Page	6532, C1, E1	Zero Page	6532, C1, E1
2	ROM	H1	ROM	2364, L1
3	ROM	L1	ROM	2364, H1
4	ROM	J1	N/A	
5	Zero Page	6530, K3; 6504, H3	Zero Page	6530, K3; 6502, H3
6	N/A		N/A	
7	RAM	2114, D4, D5	RAM	2114, D4, D5
8	RAM	2114, E4, E5	RAM	2114, E4, E5
9	RAM	2114, F4, F5	RAM	2114, F4, F5
10	ROM	6530, K3; 6504, H3	ROM	6530, K3; 6502, H3

# PET/CBM Disk Access Routines

Action	Hex	Dec	Method To Access From Within Basic
CONCAT	\$FF93	65427	sys65427 "filename",d# to "otherfilename",d#
DOPEN	\$FF96	65430	sys65430 #lf,"filename",d#
DCLOSE	\$FF99	65433	sys65433 alone or followed by #lf
RECORD	\$FF9C	65436	sys65436 #lf,(r#),(pr)
HEADER	\$FF9F	65439	sys65439 "disk name",d#,iid
COLLECT	\$FFA2	65442	sys65442 d#
BACKUP	\$FFA5	65445	sys65445 d# to d#
COPY	\$FFA8	65448	sys65448 "filename",d# to "filename",d#
APPEND	\$FFAB	65451	sys65451 #lf,"filename"
DSAVE	\$FFAE	65454	sys65454 "filename",d#
DLOAD	\$FFB1	65457	sys65457 "filename",d#
CATALOG	\$FFB4	65460	sys65460 d# (same for DIRECTORY)
RENAME	\$FFB7	65463	sys65463 "filename",d# to "newfilename"
SCRATCH	\$FFBA	65466	sys65466 "filename",d#
OPEN	\$FFC0	65472	sys(65472) lf,ua,sa,"d#":filename,type,operation"
CLOSE	\$FFC3	65475	sys(65475) lf
LOAD	\$FFD5	65493	sys(65493) "d#":filename",ua
SAVE	\$FFD8	65496	sys(65496) "d#":filename",ua
VERIFY	\$FFDB	65499	sys(65499) "d#":filename",ua

lf = logical file number  
 sa = secondary address  
 ua = drive unit address  
 d# = drive number  
 r# = record number

pr = pointer within record  
 id = 2 character identifier  
 type = either : s (seq), p (prg), or u (usr)  
 operation = either : w (write), r (read), a (append), or (m) modify

## User Command Jump Table

Standard Syntax	Alternate (1541: n/a)	Function
U0		Reset User Jump Vector
U1	UA	Block-Read replacement
U2	UB	Block-Write replacement
		<b>4040/8X50 1541/2031 2031/D90XX Low-Profile</b>
U3	UC	Jump to \$1300 Jump to \$0500
U4	UD	Jump to \$1303 Jump to \$0503
U5	UE	Jump to \$1306 Jump to \$0506
U6	UF	Jump to \$1309 Jump to \$0509
U7	UG	Jump to \$130C Jump to \$050C
U8	UH	Jump to \$130F Jump to \$050F
U9	UI	Jump to \$10F0 Jump to \$FFFA (NMI)
U:	UJ	Power-Up Vector (reset)

## Sector Distribution By Track

Track Number	Number of Sectors		
	4040	2031	1541
1 - 17	21	21	21
18 - 24	19	19	19
25 - 30	18	18	18
31 - 35	17	17	17

Track Number	Number of Sectors	
	8050	8250
1 - 39	29	29
40 - 53	27	27
54 - 64	25	25
65 - 77	23	23
78 - 116		29
117 - 130		27
131 - 141		25
142 - 154		23

**D9060/D9090** - 153 tracks per recording surface (4 on D9060 and 6 on the D9090) with 32 sectors per track

## GCR Codes

GCR is the method in which disk data is magnetically stored. It is based on transitions (ie. 1 to 0, or 0 to 1) A transition is decoded as 0, no transition decodes to a 1.

Hex	GCR	Binary	Dec	Hex	GCR	Binary	Dec
\$00	01010	0000	0	\$08	01001	1000	8
\$01	01011	0001	1	\$09	11001	1001	9
\$02	10010	0010	2	\$0A	11010	1010	10
\$03	10011	0011	3	\$0B	11011	1011	11
\$04	01110	0100	4	\$0C	01101	1100	12
\$05	01111	0101	5	\$0D	11101	1101	13
\$06	10110	0110	6	\$0E	11110	1110	14
\$07	10111	0111	7	\$0F	10101	1111	15







Table of disk drive commands and their descriptions. Includes commands like DX0000, PRSEQ, X0015, CPYDTP, EXLPO, FIXIT, TRFNME, COPY, COP01, CY, OPIRFL, GIBYTE, RENAME, CHKIN, CHKIO, VERDIR, VALDAT, MRKBAM, NEV/MPV, NEWMAP, ECHKSM, MEM, MEMEX, MEMRD, MEMERR, MEMWRT, USER, USRINT, US10, USREXC, OPNBLK, OB05, BLOCK, BLK10, BLK30, BLK40, BCTAB, BCIMP, BLKPAR, ASCHEX, DECTAB, BLKFRF, BLKALC, BA40, BLKRD2, BLKRD3, BLKRD, UBLKRD, BLKWT, UBLKWT, BLKEXC, BLKPTR, BUPTST, BT15, BKOTST, BLKST, FNDREL, MULPLY, DIV254, DIV120, DIV100, DIV200, ZERRS, ACCX4, ACCX2, ADDRES, USEDTS, FREUSE, BMASK, DBLBUF, PIBYTE, PUT, L42, TSTJOB, RECOV, RECI, OK, AGAIN, NOTYET, WAITD, SETHDR, PUTBYT, PUTB1, INTDRV, INTS, INTDRV, STRDBL.

Table of disk drive commands and their descriptions. Includes commands like RDBUF, WRITBUF, STRTIT, FNDRCH, FNDWCH, TYPFPL, GETPRE, GETBYT, RDBYT, WRITBYT, WRT0, INCNT, INCPTR, SETDRN, GETWCH, GETWCH, GETRCH, GETR2, GBERR, FRECHN, FRECO, FRERD, FRWRRT, RELJNX, RELBUF, GETBUF, FREBUF, CLRCHN, CLDCHN, FNDLX, GBTY, FNDGET, SEOGET, GETS, GETERC, NXTBUF, DTRD, DRTWRT, DRT, OPNIRD, OPNIWR, NXDRBK, SETPNT, FREICH, GETPNT, DRDBYT, BUFIND, SETLJB, SETJOB, TSERR, HEDZTS, TSCHK, VNERR, DOIT, ADDPIL, OPEN, OP02, OP021, OP04, OP041, OP0415, OP05, OP81, OP815, OP82, OP90, OP95, OP115, OP120, OP120, OPREAD, OPWRIT, OPPIN, OPKTM, CKM1, CKT1, APPEND, LOADIR, LD01, LD02, CLOSE, CLS10, CLSALL, CLSCHN, CLSREL, CLSWRT, MARPOT, MARCHK, CLSIDR, OPNRCH, OR30, INTPNT, OPNWCH, PUTSS.

Table of disk drive commands and their descriptions. Includes commands like SCFLG, SETFLG, CLRFLG, TSTFLG, TSTWRT, TSTCHN, SCRUB, SETLAK, GETLNK, NULLNK, SETUO, CLRBLK, GETHDR, WRTAB, RDAB, WRTOUT, RDIN, WRTSS, RDSS, RDSS, RDSS, SJUO, SJUO, RDLNK, BOT0BO, CLRBUF, SSSET, SSDIR, SETSSP, SSPOS, IBRD, IBWT, IBOP, GSSPNT, SCALI, SSCALC, ADDT12, STEST, GETACT, GAFLGS, NXTREC, NRBUF, RELPUT, WRREL, CLREC, SDIRTY, CDIRTY, RDREL, RDREL, SETLST, FNDLST, SSEND, BREAK, RECORD, POSITN, POSBUF, BHERE, FDCA, NULBUF, ADDNR, ADDRLE, AR20, AR25, NEWS, NM1, PATCH, FCHKSM, UBLOCK, kernel nmi, kernel disk initialization, kernel atn irq process.





8050 RAM Memory \$0100-

Table with columns: Location, Label, Description. Contains memory addresses from 0100-01FF to 103F-1040 and their corresponding labels and descriptions.

Table with columns: Location, Label, Description. Contains memory addresses from 1041-1048 to 4342-4343 and their corresponding labels and descriptions.

Table with columns: Location, Label, Description. Contains memory addresses from 4344-4345 to 4400-BFFF and their corresponding labels and descriptions.

8050 Dual Disk ROM Map

Table with columns: Loc., Label, Description. Contains ROM addresses from C000 to C99A and their corresponding labels and descriptions.

Table with columns: Loc., Label, Description. Contains ROM addresses from C99B to D00A and their corresponding labels and descriptions.

Table with columns: Loc., Label, Description. Contains ROM addresses from D00C to D4A7 and their corresponding labels and descriptions.



1541 System Constants

Table with 3 columns: Hex Val, Label, Description. Lists system constants like LED1, NOTRDY, RDMDY, etc.

1541 Disk Memory Map

Table with 3 columns: Hex Val, Label, Description. Lists disk memory constants like TYPMSK, VERERR, CLKOUT, etc.

Table with 3 columns: Hex Val, Label, Description. Lists disk memory constants like TOPRD, TOPWRT, JMPC, etc.

1541 RAM Memory Map with Zero Page Contents at Power Up

Table with 4 columns: Hex Location, Content, CBM Label, Function. Lists RAM memory locations and their functions.

Table with 4 columns: Hex Location, Content, CBM Label, Function. Lists RAM memory locations and their functions.

references to Drive 1 are mostly unused locations

Table with 4 columns: Hex Location, Content, CBM Label, Function. Lists RAM memory locations and their functions.







# Music Symbols

	Above staff: play 1 octave higher (Note = Note x 2) Below staff: play 1 octave lower (Note = Note / 2)		<b>Slight Accent.</b>
	<b>Slur or Bowing:</b> Indicates <i>Legato</i> when connecting a group of notes. Indicates a <i>Tie</i> when connecting 2 notes of the same pitch (2nd note is NOT played - value of 2nd note is added to the value of the 1st note).		<b>Staccato Marks:</b> Shorten duration of note(s).
	<b>Trill:</b> Alternate adjacent notes rapidly.		<b>Moderate Staccato.</b>
	<b>Mordent:</b> Play note, add next higher note and release, holding 1st note.		<b>Metronome Setting.</b>
	<b>Inverted Mordent:</b> Play note, add next lower note and release, holding 1st note.		<b>Clefs:</b> Treble or G, Bass or F, C Clef.
	<b>Pedal:</b> Attack and Release.		<b>Beat Interrupts:</b> Divide the beat into other than the regular notation.
	<b>Pedal Release.</b>		<b>Sharp, Double Sharp.</b>
	<b>Turn.</b>		<b>Flat, Double Flat.</b>
	<b>Dal Segno:</b> Like GOTO (label).		<b>Natural.</b>
	<b>Crescendo:</b> Smoothly increasing intensity.		<b>Meter Signatures:</b> 2/4, 6/8, 3/2, 4/4, 2/2, respectively.
	<b>Decrescendo:</b> Smoothly decreasing intensity.		<b>Whole Rest, Half Rest, Quarter Rest.</b>
	<b>First &amp; Second Endings:</b> Play ending 1, then 2 (omit 1)		<b>1/8 Rest, 1/16 Rest, 1/32 Rest.</b>
	<b>Repeat Marks:</b> Like FOR I = 1 TO 2.		<b>Multiple Measure Rest:</b> Rest for n measures.
	<b>Repeat Measure.</b>		<b>Natural Harmonic:</b> On stringed instruments.
	<b>Fermata or Hold.</b>		<b>Artificial Harmonic</b> on the Violin. Sounds 2 octaves above lower tone.
	Indicates voice line moving from one staff to another.		<b>Notes:</b> Double Whole (breve), Whole (semibreve), Half (minim), Quarter (crotchet).
	<b>Arpeggiate:</b> Play notes in a chord successively from bottom to top, or top to bottom, respectively.		<b>Notes:</b> Eighth (quaver), Sixteenth (semiquaver), Thirty-Second (demisemiquaver).
	<b>Glissando:</b> Slide notes.		<b>Dotted Note:</b> Increment duration by 50%.
	<b>Down-Bow, Up-Bow:</b> For stringed instruments.		<b>Tremolo:</b> Repeat rapidly for duration of note.
	<b>Accent Marks:</b> Intensity or pressure increase on note.		

Octave 4: C D E F G A B C D E F G A B  
Octave 5: C D E F G A B C D E F G A B

Middle C

Octave 2: C D E F G A B C D E F G A B  
Octave 3: C D E F G A B C D E F G A B

C#	D#	F#	G#	A#
D <sub>b</sub>	E <sub>b</sub>	G <sub>b</sub>	A <sub>b</sub>	B <sub>b</sub>
B <sub>x</sub>	F <sub>bb</sub>	E <sub>x</sub>		C <sub>bb</sub>
C	D	E	F	G
B#	C <sub>x</sub>	D <sub>x</sub>	E#	F <sub>x</sub>
D <sub>bb</sub>	E <sub>bb</sub>	F <sub>b</sub>	G <sub>bb</sub>	A <sub>bb</sub>
				B <sub>bb</sub>
				C <sub>b</sub>

C D E F G A B C D E F G A B C D E F G A B C D E F G A B

C	Doh	Tonic
D	Ray	Supertonic
E	Me	Mediant
F	Fah	Subdominant
G	Soh	Dominant
A	Lah	Submediant
B	Te	Leading Note
C	Doh	Tonic

C Major no signature	G Major 1 sharp	D Major 2 sharps	A Major 3 sharps	C Major no signature	F Major 1 flat	B Flat Major 2 flats	E Flat Major 3 flats
E Major 4 sharps	B Major 5 sharps	F Sharp Major 6 sharps	C Sharp Major 7 sharps	A Flat Major 4 flats	D Flat Major 5 flats	G Flat Major 6 flats	C Flat Major 7 flats

## Note Frequency Table

Frequency in Hz

Based on formula:  $\text{Note}_N = \text{Note}_{N-1} \times 2^{1/12}$

(- Octave Not Accessible) (\* Octave Only Partially Accessible)

Note in: For:	Octave:								
	0	1	2	3	4	5	6	7	8
CB2	-	-	-	-	0	1	2	3	-
VIC Voice 1	-	0	1	2	3*	-	-	-	-
VIC Voice 2	-	-	0	1	2	3*	-	-	-
VIC Voice 3	-	-	-	0	1	2	3*	-	-
C64	0	1	2	3	4	5	6	7	-
+4/C16	-	-	0	1	2	3	4	5	6
C	16.3516	32.7032	65.4064	130.813	261.626	523.251	1046.50	2093.00	4186.01
C#	17.3239	34.6478	69.2957	138.591	277.183	554.365	1108.73	2217.46	4434.92
D	18.3540	36.7081	73.4162	146.832	293.665	587.330	1174.66	2349.32	4698.64
D#	19.4454	38.8909	77.7817	155.563	311.127	622.254	1244.51	2489.02	4978.03
E	20.6017	41.2034	82.4069	164.814	329.628	659.255	1318.51	2637.02	5274.04
F	21.8268	43.6536	87.3071	174.614	349.228	698.456	1396.91	2793.83	5587.65
F#	23.1247	46.2493	92.4986	184.997	369.994	739.989	1479.98	2959.96	5919.91
G	24.4997	48.9994	97.9989	195.998	391.995	783.991	1567.98	3135.96	6271.93
G#	25.9565	51.9131	103.826	207.652	415.305	830.609	1661.22	3322.44	6644.88
A	27.5	55.0	110.0	220.0	440.0	880.0	1760.0	3520.0	7040.0
A#	29.1352	58.2705	116.541	233.082	466.164	932.328	1864.66	3729.31	7458.62
B	30.8671	63.7354	127.471	254.942	509.883	1019.767	2039.53	4079.07	8158.13

## Chord Note Derivatives

Notes are shown in diminishing order of importance.

Chord	Major	Minor	Seventh	Minor 7th	Diminished
A <sup>b</sup> / G#	A <sup>b</sup> C E <sup>b</sup>	G# B D#	A <sup>b</sup> C G <sup>b</sup> E <sup>b</sup>	G# B F# D#	G# B D F
A	A C# E	A C E	A C# G E	A C G E	A C E <sup>b</sup> F#
B <sup>b</sup> / A#	B <sup>b</sup> D F	B <sup>b</sup> D <sup>b</sup> F	B <sup>b</sup> D A <sup>b</sup> F	B <sup>b</sup> D <sup>b</sup> A <sup>b</sup> F	B <sup>b</sup> D <sup>b</sup> E G
B / C <sup>b</sup>	B D# F#	B D F#	B D# A F#	B D A F#	B D F A <sup>b</sup>
C / B#	C E G	C E <sup>b</sup> G	C E B <sup>b</sup> G	C E <sup>b</sup> B <sup>b</sup> G	C E <sup>b</sup> F# A
D <sup>b</sup> / C#	D <sup>b</sup> F A <sup>b</sup>	C# E G#	D <sup>b</sup> F C <sup>b</sup> A <sup>b</sup>	C# E B G#	C# E G A#
D	D F# A	D F A	D F# C A	D F C A	D F A <sup>b</sup> B
E <sup>b</sup> / D#	E <sup>b</sup> G B <sup>b</sup>	E <sup>b</sup> G <sup>b</sup> B <sup>b</sup>	E <sup>b</sup> G D <sup>b</sup> B <sup>b</sup>	E <sup>b</sup> G <sup>b</sup> D <sup>b</sup> B <sup>b</sup>	E <sup>b</sup> G <sup>b</sup> A C
E / F <sup>b</sup>	E G# B	E G B	E G# D B	E G D B	E G B <sup>b</sup> D <sup>b</sup>
F / E#	F A C	F A <sup>b</sup> C	F A E <sup>b</sup> C	F A <sup>b</sup> E <sup>b</sup> C	F A <sup>b</sup> B D
E <sup>b</sup> / F#	F# A# C#	F# A C#	F# A# E C#	F# A# E C#	F# A C D#
G	G B D	G B <sup>b</sup> D	G B F D	G B <sup>b</sup> F D	G B <sup>b</sup> D <sup>b</sup> E
Chord	Augmented	Suspended 4th	Major 7th	Major 6th	Major 9th
A <sup>b</sup> / G#	A <sup>b</sup> C E	A <sup>b</sup> D <sup>b</sup> E <sup>b</sup>	A <sup>b</sup> C G E <sup>b</sup>	A <sup>b</sup> C F E <sup>b</sup>	A <sup>b</sup> C B <sup>b</sup> G <sup>b</sup> E <sup>b</sup>
A	A C# F	A D E	A C# G# E	A C# F# E	A C# B <sup>b</sup> G <sup>b</sup> E <sup>b</sup>
B <sup>b</sup> / A#	B <sup>b</sup> D F#	B <sup>b</sup> E <sup>b</sup> F	B <sup>b</sup> D A F	B <sup>b</sup> D G F	B <sup>b</sup> D C A <sup>b</sup> F
B / C <sup>b</sup>	B D# G	B E F#	B D# A# F#	B D# G# F#	B D# C# A F#
C / B#	C E G#	C F G	C E B G	C E A G	C E D B <sup>b</sup> G
D <sup>b</sup> / C#	D <sup>b</sup> F A	D <sup>b</sup> G <sup>b</sup> A <sup>b</sup>	D <sup>b</sup> F C A <sup>b</sup>	D <sup>b</sup> F B <sup>b</sup> A <sup>b</sup>	D <sup>b</sup> F E <sup>b</sup> C <sup>b</sup> A <sup>b</sup>
D	D F# A#	D G A	D F# C# A	D F# B A	D F# E C A
E <sup>b</sup> / D#	E <sup>b</sup> G B	E <sup>b</sup> A <sup>b</sup> B <sup>b</sup>	E <sup>b</sup> G D B <sup>b</sup>	E <sup>b</sup> G C B <sup>b</sup>	E <sup>b</sup> G F D <sup>b</sup> B <sup>b</sup>
E / F <sup>b</sup>	E G# C	E A B	E G# D# B	E G# C# B	E G# F# D B
F / E#	F A C#	F B <sup>b</sup> C	F A E C	F A D C	F A G E <sup>b</sup> C
E <sup>b</sup> / F#	F# A# D	F# B C#	G <sup>b</sup> B <sup>b</sup> F D <sup>b</sup>	G# A# D# C#	F# A# G# E C#
G	G B D#	G C D	G B F# D	G B E D	G B A F D



## VIC 20 Screen & Border Colours

POKE 36879, X:		Border							
Screen	BLK	WHT	RED	CYN	PUR	GRN	BLU	YEL	
BLK	8	9	10	11	12	13	14	15	
WHT	24	25	26	27	28	29	30	31	
RED	40	41	42	43	44	45	46	47	
CYN	56	57	58	59	60	61	62	63	
PUR	72	73	74	75	76	77	78	79	
GRN	88	89	90	91	92	93	94	95	
BLU	104	105	106	107	108	109	110	111	
YEL	120	121	122	123	124	125	126	127	
ORG	136	137	138	139	140	141	142	143	
Lt. ORG	152	153	154	155	156	157	158	159	
PNK	168	169	170	171	172	173	174	175	
Lt. CYN	184	185	186	187	188	189	190	191	
Lt. PUR	200	201	202	203	204	205	206	207	
Lt. GRN	216	217	218	219	220	221	222	223	
Lt. BLU	232	233	234	235	236	237	238	239	
Lt. YEL	248	249	250	251	252	253	254	255	

## Colour Codes

Colour:	VIC	C64	+4	ASCII	Colour:	VIC	C64	+4	ASCII
Black	0	0	1	144	Medium Grey		12		152
White	1	1	2	5	Light Purple	12*			
Red	2	2	3	28	Blue-Green			13	152
Cyan	3	3	4	159	Light Green	13*	13	16	153
Purple	4	4	5	156	Light Blue	14*	14	14	154
Green	5	5	6	30	Dark Blue			15	154
Blue	6	6	7	31	Light Grey		15		155
Yellow	7	7	8	158	Light Yellow	15*			
Orange	8*	8	9	129	* = Not available as a character colour. Colour values for VIC/C64 are POKEd into the appropriate registers (see memory maps). +4 values are used in the COLOR Command (same for C16). ASCII values are PRINTed using CHR\$.				
Brown		9	10	149					
Light Orange	9*								
Pink	10*	10	12	150					
Yellow-Green			11	150					
Dark Grey		11		151					
Light Cyan	11*								

## Table Of Secondary Addresses

Eg. OPEN 4, 4, 7 ; 7 is the Secondary Address on CBM printers that alters line spacing. Once open the new value can be sent. Secondary addresses are not applicable to the VIC 20/Commodore 64 RS-232 routines ('device' 2), keyboard (device 0), screen (device 3), or the CBM 8010 Modem (device 5).

Sec. Addr.	Printer 4	I/O Device & Device Number (DV#)		
		Cassette 1 or 2	Vic/64 Cassette 1	Disk 8
0	Print data exactly as received	seq. read	Load & relocate (dflt)	Load, and Dir read
1	Print data according to previously defined format	Write file + end-of-file marker on Close	Load without relocating	Program Save
2	Format Set-up	Write file + eof + end of tape marker on Close	Write file + eof + end of tape marker on Close	R/W channels are 2-14
3	Set number of lines per page for paging			
4	Enable printer format diagnostics			
5	Define a programmable character			
6	Set spacing between lines			
7	Upper/Lower case			
8	ASCII/Graphics			
9	Suppress Diagnostic Message Printing			
10	Reset Printer			
11	Set Uni-Direction			
12	Reset Uni-Direction			
13	Set Condense mode			
14	Reset Condense mode			
15	Set pseudo letter quality			
21	Reset pseudo letter quality			
17	Storing bit image data			
18	Printing bit data previously written			

Command Ch.

## Commodore 6545 Video Chip

POKE 59520, R#	POKE 59521, Value
R0	Horizontal total number of characters on line (Nht) including horizontal retrace. (true value = number + 1)
R1	Horizontal number of characters displayed (Nhd)
R2	Distance (in characters) from left to right margin of screen + 1
R3	Sync width. Lo nybble is vertical sync width (in lines) Hi nybble is horizontal sync (in characters).
R4	Number of display lines including retrace (Nvt).
R5	Vertical position of the edge of the screen.
R6	Number of display lines on screen (Nvd)
R7	Height of upper edge from bottom of screen (in lines displayed)
R8	Interlace and Skew:- Bit 0 1 = interlaced mode 0 = non interlaced mode Bit 1 if Bit 0 = 1 then interlace and video mode Bit 2 not used Bit 3 not used Bit 4 1 = scan from 32770 in memory Bit 5 1 = scan from 32772 in memory Bit 6 cursor (not implemented on the PET) Bit 7 cursor (not implemented on the PET)
R9	Number of lines between top of one display line and top of the next
R10	Cursor (not implemented on the PET)
R11	Cursor (not implemented on the PET)
R12	Control Register: Bit 0 add 256 to start address (512 for 8032) Bit 1 add 512 to start address (1024 for 8032) Bit 2 invert flyback Bit 3 invert video signal Bit 4 use top half of 4K character generator Bit 5 (not implemented on the PET) Bit 6 (not implemented on the PET) Bit 7 not used
R13	Value + 32768 is address of first character (multiply by 2 for 8032)
R14	Cursor location HI (not implemented on the PET)
R15	Cursor location LO (not implemented on the PET)
R16	Light pen position HI (read only)
R17	Light pen position LO (read only)

## 8032 Control Characters

Most functions can be activated by combinations of simultaneous key depressions, a phenomena of the keyboard hardware. Notice that the CHR\$ values of complimentary functions differ by 128.

Function	CHR\$	ESC/RVS	Keyboard Combination
BELL	7	G	
GRAPHICS TEXT	142 14	Shift N N	Both Shifts + *
SCROLL DOWN	153	Shift Y	Left Shift + TAB + I
SCROLL UP	25	Y	
SET BOTTOM	143	Shift O	Shift + Z + A + L
SET TOP	15	O	Z + A + L
INSERT LINE	149	Shift U	Shift + RVS + A + L
DELETE LINE	21	U	RVS + A + L
ERASE BEGIN	150	Shift V	Shift + TAB + $\boxtimes$ + DEL
ERASE END	22	V	TAB + $\boxtimes$ + DEL
SET/CLR TAB	137	Shift I	Shift + TAB
TAB	9	I	TAB

### 8032 Window POKEs

TOP:224, T where T=0 to 24      LEFT:226, L where L=0 to 79  
BOTTOM:225, B where B=T to 24      RIGHT:213, R where R=L to 79

## VIC 20 Screen Memory

To move the screen: POKE 36869, (PEEK(36869) AND 15) OR X  
POKE 36866, (PEEK(36866) AND 127) OR Y

X	Y	4*(PEEK(36866) AND 128) + 64*(PEEK(36869) AND 112) = Location	
		Decimal (1/2K blocks)	Hexadecimal
128	0	0	\$0000
128	128	512	\$0200
129	0	1024	0400
129	128	1536	0600
130	0	2048	0800
130	128	2560	0A00
131	0	3072	0C00
131	128	3584	0E00
132	0	4096	1000 (dflt w/exp)
132	128	4608	1200
133	0	5120	1400
133	128	5632	1600
134	0	6144	1800
134	128	6656	1A00
135	0	7168	1C00
135	128	7680	1E00 (default)
136	0	8192	2000
136	128	8704	2200
137	0	9216	2400
137	128	9728	2600
138	0	10240	2800
138	128	10752	2A00
139	0	11264	2C00
139	128	11776	2E00
140	0	12288	3000
140	128	12800	3200
141	0	13312	3400
141	128	13824	3600
142	0	14336	3800
142	128	14848	3A00
143	0	15360	3C00
143	128	15872	3E00

## Commodore 64 Screen Memory

To move the screen: POKE 53272, (PEEK(53272) AND 15) OR X

X	(3-PEEK(56576) AND 3) * 16384 + (X*64) = Location For Screen at Bank 0 (default):	
	Decimal	Hexadecimal
0	0	\$0000
16	1024	0400 (default)
32	2048	0800
48	3072	0C00
64	4096	1000
80	5120	1400
96	6144	1800
112	7168	1C00
128	8192	2000
144	9216	2400
160	10240	2800
176	11264	2C00
192	12288	3000
208	13312	3400
224	14336	3800
240	15360	3C00

## Commodore 64 VIC II Address

To move VIC II: POKE 56576, (PEEK(56576) AND 252) OR X ;X=3-Bank#

Bank	X	VIC II Chip Address Range	
		Decimal (16K blocks)	Hexadecimal
0	3	0-16383	\$0000-3FFF (default)
1	2	16384-32767	4000-7FFF
2	1	32768-49151	8000-BFFF
3	0	49152-65535	C000-FFFF

Note: Character ROM only available with VIC II in bank 0 or 2

## VIC 20 Character Base

To move the character base: POKE 36869, (PEEK(36869) AND 240) OR X

X*	32768 + (PEEK(36869) AND 15) * 1024 = Location	
	Decimal (1K blocks)	Hexadecimal
0	32768-34815	\$8000-87FF (dflt)
1	33792-35839	8400-8BFF
2	34816-36863	8800-8FFF
3	35840-37887	8C00-93FF
4	36864-38911	9000-97FF
5	37888-39935	9400-9BFF
6	38912-40959	9800-9FFF
7	39936-41983	9C00-A3FF
8	0-2047	0000-07FF
9	1024-3071	0400-0BFF
10	2048-4095	0800-0FFF
11	3072-5019	0C00-13FF
12	4096-6143	1000-17FF
13	5020-7167	1400-1BFF
14	6144-8191	1800-1FFF
15	7168-9216	1C00-23FF

\* X = PEEK(36869) AND 15

## Commodore 64 Character Base

To move the character base: POKE 53272, (PEEK(53272) AND 240) OR X

X*	(3-PEEK(56576) AND 3) * 16384 + (X*64) = Location For Screen at Bank 0 (default):	
	Decimal (2K blocks)	Hexadecimal
0	0-2047	\$0000-07FF
2	2048-4095	0800-0FFF
4	4096-6143	1000-17FF *1
6	6144-8191	1800-1FFF *2
8	8192-10293	2000-27FF
10	10240-12287	2800-2FFF
12	12288-14335	3000-37FF
14	14336-16383	3800-3FFF

\* - X = PEEK(53272) AND 14

\*1 - Lower 2K of Character ROM (Bank 0 or 2 only) (default)

\*2 - Upper 2K of Character ROM (Bank 0 or 2 only)

## Character ROM Contents

Character ROM is the same in all machines, but only addressable in VIC 20/C64

2K Block	VIC 20		Commodore 64			Contents
	Default Address		Default Address		VIC II Image	
	Dec (1/2K blocks)	Hex	Dec (1/2K blocks)	Hex	Hex	
0	32768-33279	8000-81FF	53248-53759	D000-D1FF	1000-11FF	Upper case characters
	33280-33791	8200-83FF	53760-54271	D200-D3FF	1200-13FF	Graphics characters
	33792-34303	8400-85FF	54272-54783	D400-D5FF	1400-15FF	Reversed upper case characters
	34304-34815	8600-87FF	54784-55295	D600-D7FF	1600-17FF	Reversed graphics characters
1	34816-35327	8800-89FF	55296-55807	D800-D9FF	1800-19FF	Lower case characters
	35328-35839	8A00-8BFF	55808-56319	DA00-DBFF	1A00-1BFF	Upper case and graphics characters
	35840-36351	8C00-8DFF	56320-56831	DC00-DDFF	1C00-1DFF	Reversed lower case characters
	36352-36863	8E00-8FFF	56832-57343	DE00-DFFF	1E00-1FFF	Reversed upper case and graphics

















### + 4 / C16 Screen Map

3072
1 0C00
2 0C28
3 0C50
4 0C78
5 0CA0
6 0CC8
7 0CF0
8 0D18
9 0D40
10 0D68
11 0D90
12 0DB8
13 0DE0
14 0E08
15 0E30
16 0E58
17 0E80
18 0EA8
19 0ED0
20 0EF8
21 0F20
22 0F48
23 0F70
24 0F98
25 0FC0

4071

### + 4 / C16 Colour Table Map

2087
1 0800
2 0828
3 0850
4 0878
5 08A0
6 08C8
7 08F0
8 0918
9 0940
10 0968
11 0990
12 09B8
13 09E0
14 0A08
15 0A30
16 0A58
17 0A80
18 0AA8
19 0AD0
20 0AF8
21 0B20
22 0B48
23 0B70
24 0B98
25 0BC0

3047

# True ASCII Conversion Table

Dec	x256	x256 +32768	Hex	CBM True	Even Parity			Binary	Odd Parity			BCD	EBCDIC
					Dec	Hex	Oct		Dec	Hex	Oct		
0	0	32768	00	NUL	0	00	000	00000000	128	80	200	00000000	00
1	256	33024	01	SOH	129	81	201	00000001	1	01	001	00000001	01
2	512	33280	02	STX	130	82	202	00000010	2	02	002	00000010	02
3	768	33536	03	ETX	3	03	003	00000011	131	83	203	00000011	03
4	1024	33792	04	EOT	132	84	204	00000100	4	04	004	00000100	37
5	1280	34048	05	ENQ	5	05	005	00000101	133	85	205	00000101	2D
6	1536	34304	06	ACK	6	06	006	00000110	134	86	206	00000110	2E
7	1792	34560	07	BEL	135	87	207	00000111	7	07	007	00000111	2F
8	2048	34816	08	BS	136	88	210	00001000	8	08	010	00001000	16
9	2304	35072	09	HT	9	09	011	00001001	137	89	211	00001001	05
10	2560	35328	0A	LF	10	0A	012	00001010	138	8A	212	00001010	25
11	2816	35584	0B	VT	139	8B	213	00001011	11	0B	013	00001011	0B
12	3072	35840	0C	FF	12	0C	014	00001100	140	8C	214	00001100	0C
13	3328	36096	0D	CR	141	8D	215	00001101	13	0D	015	00001101	0D
14	3584	36352	0E	SO	142	8E	216	00001110	14	0E	016	00001110	0E
15	3840	36608	0F	SI	15	0F	017	00001111	143	8F	217	00001111	0F
16	4096	36864	10	DLE	144	90	220	00010000	16	10	020	00010000	10
17	4352	37120	11	DC1	17	11	021	00010001	145	91	221	00010001	11
18	4608	37376	12	DC2	18	12	022	00010010	146	92	222	00010010	12
19	4864	37632	13	DC3	147	93	223	00010011	19	13	023	00010011	13
20	5120	37888	14	DC4	20	14	024	00010100	148	94	224	00010100	14
21	5376	38144	15	NAK	149	95	225	00010101	21	15	025	00010101	3D
22	5632	38400	16	SYN	150	96	226	00010110	22	16	026	00010110	32
23	5888	38656	17	ETB	23	17	027	00010111	151	97	227	00010111	26
24	6144	38912	18	CAN	24	18	030	00010100	152	98	230	00010100	18
25	6400	39168	19	EM	153	99	231	00010110	25	19	031	00010110	19
26	6656	39424	1A	SUB	154	9A	232	00010111	26	1A	032	00010111	3F
27	6912	39680	1B	ESC	27	1B	033	00011000	155	9B	233	00011000	27
28	7168	39936	1C	FS	156	9C	234	00011001	28	1C	034	00011001	22
29	7424	40192	1D	GS	29	1D	035	00011010	157	9D	235	00011010	29
30	7680	40448	1E	RS	30	1E	036	00011011	158	9E	236	00011011	35
31	7936	40704	1F	US	159	9F	237	00011100	31	1F	037	00011100	31
32	8192	40960	20		160	A0	240	00011101	32	20	040	00011101	40
33	8448	41216	21	!	33	21	041	00011110	161	A1	241	00011110	5A
34	8704	41472	22	!"	34	22	042	00011111	162	A2	242	00011111	7F
35	8960	41728	23	#	163	A3	243	00011000	35	23	043	00011000	7B
36	9216	41984	24	\$	36	24	044	00011001	164	A4	244	00011001	5B
37	9472	42240	25	%	165	A5	245	00011010	37	25	045	00011010	6C
38	9728	42496	26	&	166	A6	246	00011011	38	26	046	00011011	50
39	9984	42752	27	'	39	27	047	00011100	167	A7	247	00011100	7D
40	10240	43008	28	(	40	28	050	00011101	168	A8	250	00011101	4D
41	10496	43264	29	)	169	A9	251	00011110	41	29	051	00011110	5D
42	10752	43520	2A	*	170	AA	252	00011111	42	2A	052	00011111	5C
43	11008	43776	2B	+	43	2B	053	00011000	171	AB	253	00011000	4E
44	11264	44032	2C	,	172	AC	254	00011001	44	2C	054	00011001	6B
45	11520	44288	2D	-	45	2D	055	00011010	173	AD	255	00011010	60
46	11776	44544	2E	=	46	2E	056	00011011	174	AE	256	00011011	4B
47	12032	44800	2F	/	175	AF	257	00011100	47	2F	057	00011100	61
48	12288	45056	30	0	48	30	060	00011101	176	B0	260	00011101	F0
49	12544	45312	31	1	177	B1	261	00011110	49	31	061	00011110	F1
50	12800	45568	32	2	178	B2	262	00011111	50	32	062	00011111	F2
51	13056	45824	33	3	51	33	063	00011000	179	B3	263	00011000	F3
52	13312	46080	34	4	180	B4	264	00011001	52	34	064	00011001	F4
53	13568	46336	35	5	53	35	065	00011010	181	B5	265	00011010	F5
54	13824	46592	36	6	54	36	066	00011011	182	B6	266	00011011	F6
55	14080	46848	37	7	183	B7	267	00011100	55	37	067	00011100	F7
56	14336	47104	38	8	184	B8	270	00011101	56	38	070	00011101	F8
57	14592	47360	39	9	57	39	071	00011110	185	B9	271	00011110	F9
58	14848	47616	3A	:	58	3A	072	00011111	186	BA	272	00011111	7A
59	15104	47872	3B	;	187	BB	273	00011000	59	3B	073	00011000	5E
60	15360	48128	3C	<	60	3C	074	00011001	188	BC	274	00011001	4C
61	15616	48384	3D	=	189	BD	275	00011010	61	3D	075	00011010	7E
62	15872	48640	3E	>	190	BE	276	00011011	62	3E	076	00011011	6E
63	16128	48896	3F	?	63	3F	077	00011100	191	BF	277	00011100	6F









Table with 3 columns: Phone Number, State, City. Includes entries for Granville, Hamilton, Kent, Lima, Mansfield, Marysville, North Canton, Parma, Springfield, Toledo, Warren, Youngstown.

Oklahoma (OK) section with 15 entries listing phone numbers and cities like Ardmore, Bethany, Enid, Lawton, Norman, Oklahoma City.

Oregon (OR) section with 12 entries listing phone numbers and cities like Corvallis, Eugene, Medford, Portland, Salem.

Pennsylvania (PA) section with 45 entries listing phone numbers and cities like Allentown, Altoona, Bethlehem, Downingtown, Erie, Greensburg, Harrisburg, Johnstown, King of Prussia, Lancaster, Latrobe, Levittown, New Castle, Norristown, Penn Hills, Philadelphia, Pittsburgh, Scranton, State College, Upper Darby, Valley Forge, Wilkes Barre, York.

Puerto Rico (PR) section with 3 entries listing phone numbers and cities like Mayaguez, Ponce, San Juan.

Rhode Island (RI) section with 10 entries listing phone numbers and cities like Newport, Pawtucket, Providence, Warwick, Woonsocket.

South Carolina (SC) section with 3 entries listing phone numbers and cities like Charleston.

Table with 3 columns: Phone Number, State, City. Includes entries for Charleston, Columbia, Greenville, Spartanburg.

South Dakota (SD) section with 6 entries listing phone numbers and cities like Pierre, Rapid City, Sioux Falls.

Tennessee (TN) section with 15 entries listing phone numbers and cities like Bristol, Chattanooga, Knoxville, Memphis, Nashville, Oak Ridge.

Texas (TX) section with 30 entries listing phone numbers and cities like Abilene, Amarillo, Austin, Baytown, Brownsville, Bryan, College Station, Corpus Christi, Dallas, Denton, El Paso, Ft. Worth, Galveston, Houston, Lubbock, McAllen, Midland, Nederland, Odessa, San Angelo, San Antonio, Waco, Wichita Falls.

Utah (UT) section with 10 entries listing phone numbers and cities like Ogden, Provo, Salt Lake City, Terminal, Tyler.

Virginia (VA) section with 4 entries listing phone numbers and cities like Alexandria, Annandale, Arlington.

Table with 3 columns: Phone Number, State, City. Includes entries for Arlington, Charlottesville, Chesapeake, Fairfax, Newport News, Norfolk, Petersburg, Portsmouth, Richmond, Roanoke, Springfield, Williamsburg.

Vermont (VT) section with 4 entries listing phone numbers and cities like Burlington, Montpelier.

Washington (WA) section with 15 entries listing phone numbers and cities like Auburn, Bellevue, Bellingham, Enumclaw, Longview, Olympia, Richland, Seattle, Tacoma, Vancouver, Wenatchee, Yakima.

Wisconsin (WI) section with 15 entries listing phone numbers and cities like Appleton, Beloit, Brookfield, Eau Claire, Green Bay, La Crosse, Madison, Neenah, Oshkosh, Racine, West Bend.

West Virginia (WV) section with 10 entries listing phone numbers and cities like Charleston, Huntington, Morgantown, Parkersburg, Wheeling.

Wyoming (WY) section with 3 entries listing phone numbers and cities like Casper, Cheyenne.

CompuServe CIS Commodore Information Service

My local CompuServe Number is:

My CompuServe Account Number is:

- T - TOP: TOP menu page. Goes directly page CIS-1
M - MENU: Previous MENU. Goes back to the menu page that points to the current page.
G - GO: Go n... Go directly to page 'n'.
H - HELP: Displays HELP file.
S - SCROLL: S n... SCROLL from item 'n'.
OFF or BYE: These commands will disconnect you from the Information Service immediately.
F - FORWARD: FORWARD a page. Displays the next page in a series of pages.
B - BACKWARD: Returns to the preceding page.
P - PREVIOUS: Go to the PREVIOUS item from last selected menu.
N - NEXT: Go to the NEXT item from last selected menu.
R - RESEND: RESEND the current page.
Control Characters: Control characters are entered by holding down the Control key while at the same time pressing the character key.
1C interrupts display or a program's execution...
1U deletes the line which you are currently typing...
1V redisplay the partial line you are typing...
1H backspaces, deleting the character that was there...
1A temporarily suspends output at the end of the current line...
1S temporarily suspends output immediately...
1Q resumes output after 1A or 1S...
1O stops output which is in process...
1P interrupts output and takes you to a command prompt.

# CompuServe Category Index

SIG = Special Interest Group

Category	Page	Category	Page	Category	Page
AAMSI Communications	AAM	Entertainment SIG	HOM-29	Parenting & Family Life	PFL
AAMSI SIG	SFP-5	Environmental SIG	SFP-38	Pascal SIG	PCS-55
AOPA Forum	AOP	EpsOnLine	PCS-19	Peak Delay Guide	PDG
AP Datastream	SPD-1005	Evans Economic Inc.	EEL	Personal Computing	PCS
AP Videotex, Business	APV	FOI Newslite - FDA Info.	FOI	Personal File Area	CIS-174
AP Videotex, Entertainment	APV	Family Matters SIG	HOM-144	Personality Profile	TMC-17
AP Videotex, Politics	APV	Fantasy	GAM-16	Popular Science, Autos	PSC
AP Videotex, Weather	APV	FasterMind	GAM-17	Popular Science, Energy	PSC
AP Videotex, World News	APV	Fedwatch Newsletter	MMS	Popular Science, New Product	PSP
ASCMD SIG	SFP-7	Feedback to CompuServe	CIS-8	PowerSoft's XTRA-80	PCS-56
ASI Flight Operations	ASI-11	Fifth Avenue Shopper	FTH	Primetime Radio Classics	PRC
ASI Monitor	ASI-10	Financial Forecasts	FIN-4	Product Ordering	CIS-54
ASI Service Difficulty	ASI-12	Financial Services	FIN-20	Programmer's SIG	PCS-158
Academic Amer. Encyclopedia	AAE	Fire Fighters' SIG	SFP-36	Quick Quote	FIN-20
Access Phone Numbers	LOG-50	Firstworld Travel Club	TVL	Quick Reference List	QUICK
Adventure	GAM-8	Food Buylite SIG	HOM-151	RCA SIG	PCS-57
Aircraft Insurance	AVL	Football	GAM-27	Rapaport Diamond Broker	RDC
Alternative Educ. Services	AES	Fur trader	GAM-36	Religion SIG	HOM-33
Altertext Report	ALT	GameSIG Archives	GSA	Republican Forum	HOM-41
American Ski Association	SKI	Candolf's Reports	GAN	Reversi	GAM-40
Apple User Group SIG	PCS-51	Golf	GAM-21	Rick's Arcade Center	ARC
Arcade SIG	HOM-138	Golf SIG	GAM-129	Roulette	GAM-42
Astrology	GAM-45	Gomoku	GAM-22	SAVINGS-SCAN	SAV
Atari SIG	PCS-132	Good Earth SIG	HOM-145	SHO-TIME Movie Catalog	MOV
Athlete's Outfitter	HAN	Government Publications	GPO	Scott Adams' Games	GAM-28
Aunt Nettie	NET	HamNet SIG	HOM-11	Scramble	GAM-43
AutoNet	ATO	Hammurabi	GAM-37	SeaWar	GAM-57
Aviation Rules & Reg.	AVR	Handicapped Users' Database	HUD	Shareholders Freebies	FRE
Aviation SIG (AVSIG)	SFP-6	Hangman	GAM-23	Shawmut Bank of Boston	SHW
Aviation Safety Institute	ASI	Heath User Group SIG	PCS-48	Shop-at-home	HOM-40
Aviation Weather	AWX	Heathkit Catalog	HTH	Ski SIG	HOM-36
Bacchus Data Services	VIN	Hi-Tech Forum SIG	CCC-150	Social Security Administration	SSA
Backgammon	GAM-31	Hollywood Hotline	HHL	Society of Mining Engineers	SME
Banking Services	HOM-45	Home Management	HOM-80	Software Author's SIG	PCS-117
Banshi	GAM-30	Howard Sams' Books	SAM	Space SIG	HOM-127
Belmont Golf Association	BEL	Human Sexuality	HSM	Space Trek	GAM-26
Biorhythms	GAM-29	Huntington National Bank	HNB	Space War	GAM-25
Blackjack	GAM-60	IBM-PC SIG	PCS-131	Sports SIG	HOM-110
Bridge	GAM-18	Incorporating Guide	INC	StL Post-Dispatch, Autos	SPD
Bulletin Board	HOM-30	Index	IND	StL Post-Dispatch, Business	SPD
Business & Law Review	BLR	Industry Standard Databases	TDC-4	StL Post-Dispatch, Jobs	SPD
CB	CB-10	InfoText	IFT	StL Post-Dispatch, Real Est.	SPD
CB Interest Group SIG	HOM-9	InfoWorld	INF	StL Post-Dispatch, Sports	SPD
CB Society	CUP	Information on Demand	IOD	StL Post-Dispatch, U.S. News	SPD
CEMSIG SIG	CEM-5	Intelligence Test	TMC-32	StL Post-Dispatch, Classified	SPD
CP Business Info Wire	BIW	Internal Revenue Services	IRS	Standard & Poor's	FIN-20
CP/M Users Group SIG	PCS-47	Kaypro Users Forum	PCS-25	State Capital Quiz	TMC-44
Calculate A Raise	HOM-15	Kesmai	GAM-46	Stevens Business Reports	SBR
Calculate Net Worth	HOM-16	Legal SIG	PCS-49	TRS-80 Professional Forum	PCS-21
Changing Password	CIS-175	Legal SIG	SFP-40	TRS80 Model 100 SIG	PCS-154
Changing Terminal Type	CIS-9	Literary SIG	HOM-136	TYMNET logon instructions	LOG-11
Checkbook balancer	HOM-14	Loan Amortization	HOM-17	Tandy Newsletter	TRS
Children's Games	TMC-27	Lunar Lander	GAM-24	TeleComm SIG	PCS-52
Civil War	GAM-14	MNET80 SIG	PCS-54	Telenet logon instruct	LOG-20
Clarke School for the Deaf	CSD	MUSUS SIG	PCS-55	Terminal Software	PCS-20
CoalScoop	CMP	Magic Cube Solution	GAM-35	Texas Instruments Forum	PCS-27
College Press Service	CPS	Max Ule's Tickerscreen	TKR	Text Editors	PCS-20
Color Computer SIG	PCS-126	Maze	GAM-38	TBW	TBW
Color Graphics	CIS-91	MegaWars I	GAM-20	TCB	TCB
Columbus Chamber of Commerce	CCC	MegaWars II	GAM-55	EM	EM
Command Summary	CIS-58	MegaWars III	GAM-15	TMC	TMC
Commodore	CBM	MicroQuote	FIN-9	KCS	KCS
Commodore 64 SIG	PCS-156	MicroShope	MCS	NTT	NTT
Commodore Programming Sig	PCS-116	Microsoft SIG	PCS-145	LOTUS	LOTUS
Commodore VIC20 & Pet/CBM	PCS-155	Military Vets Forum	SFP-10	TMC	TMC
Communication Industry	SFP-35	Mine-Equip	MIN-100	ESC	ESC
Comp-U-Store	CUS	Miner's Underground	SFP-44	HOM-157	HOM-157
CompuServe Rates	BIL	Money Market Services	MMS	TRV	TRV
CompuServe logon instruct	LOG	Monthly Charges	MON	TMC	TMC
CompuServe's Softex	PCS-40	Mugwump	GAM-39	UMC	UMC
Computer Art SIG	PCS-157	Multi-Player GameSIG	GAM-300	HOM-152	HOM-152
Computer Job Bank	TDC-4	Music Information Service	MUS	HOM-4	HOM-4
Computer Resume Bank	TDC-4	Music SIG	HOM-150	PCS-16	PCS-16
Computer Wire, The	TDC-4	NOAA Weather Wire	WEA	VID	VID
Computing Across America	CAA	NWS Aviation Weather	AWX	FIN-20	FIN-20
Computing Tutorials	PCS-121	Narrow-Gage Scout	LMC	FIN-18	FIN-18
Concentration	GAM-32	National Issues SIG	HOM-132	SFP-37	SFP-37
Cook's Underground	HOM-109	National Water Well Assoc	WWA	VIC	VIC
DISCOVER ORLANDO	ORL	Netwits Database	WIT	VIF	VIF
DataPac logon instruct	LOG-41	Netwits SIG	WIT-100	TWP-12	TWP-12
Democratic Forum	HOM-39	New Adventure	GAM-59	TWP	TWP
Department of State	DOS	News-A-Tron	NAT	TWP	TWP
Dice	GAM-33	Newspapers	HOM-10	TWP	TWP
Digital Research Inc.	DRI	Node Abbreviations	LOG-51	TWP-15	TWP-15
Direct Connection, The	TDC	OS9 SIG	PCS-18	TWP	TWP
EMAIL	EMA	Official Airline Guide	OAG	TWP	TWP
EMI Flight Planning	EMI	Ohio Scientific SIG	PCS-125	TWP	TWP
Economic News	FIN-10	Orch-90 SIG	HOM-13	WCT	WCT
Educational Research Sig	HOM-28	Outdoor SIG	HOM-38	NEW	NEW
Educators' SIG	HOM-137	PDP-11	PCS-53	WEC	WEC
Edutech	CAI	PGA Official Tour Guide	PGA	WWW	WWW
Election '84	VOT	PR and Marketing Forum	SFP-48	HOM-146	HOM-146
Electronic Bounce Back	EBB	Pan Am Travel Guide	PAN	WWX	WWX
Electronic Gourmet	HMS	Panasonic SIG	PCS-114	GAM-44	GAM-44

# Bulletin Boards By Area Code

24h Denotes 24-hour operation    ↔ Multi-User System    \$ Pay System, Password Required    ♀ Sexually Oriented BBS  
 ● Nighttime Operation    ★ 1200 Baud Allowed    © Password Required    † Religious orientation

201	
□ 201-864-5345	ABBS Apple-Mate, New York, NY
□ 201-835-7228	ABBS CCNJ, Pompton Plains, NJ
□ 201-891-7441	A.C.C.E.S.S., Wyckoff, NJ
□ 201-790-5910	Aphrodite-E, Haledon, NJ
□ 201-627-5151	Conference-Tree Flagship, Rockaway, NJ
□ 201-272-3686	Dial-Your-Match #14, Cranford, NJ
□ 201-462-0435	Dial-Your-Match #21, Freehold, NJ
□ 201-486-2956	Forum-80, Linden, NJ
□ 201-528-6623	Forum-80 Monmouth, Brielle, NJ
□ 201-994-9620	Net-Works The Barn, Livingston, NJ
□ 201-736-4630	Pirates Distributing
□ 201-366-2209	Pirates I/O
□ 201-423-0810	Places Unknown
□ 201-790-6795	Photo-80, Haledon, NJ
□ 201-932-3887	PMS Rutgers Univ. MicroLab, Piscataway, NJ
□ 201-887-8874	RATS System, Whippany, NJ
□ 201-584-9227	RCP/M Flanders, NJ
□ 201-272-1874	RCP/M RBBS Cranford, NJ
□ 201-775-8705	RCP/M RBBS Ocean, NJ
□ 201-747-7301	RCP/M RBBS Paul Bogdanovich, NJ
□ 201-932-3879	RCP/M RBBS Rutgers, New Brunswick, NJ
□ 201-625-1797	RCP/M The C-Line, NJ
□ 201-233-5997	Sherwood Forest
202	
□ 202-364-8617	Aladdin's Lamp
□ 202-276-8342	ARMUDIC Washington, DC
□ 202-363-8165	NWDS
□ 202-337-4694	Program Store of DC, Washington, DC
□ 202-678-9947	Ware-House III
203	
□ 203-744-4644	Bullet-80, Danbury, CT
□ 203-888-7952	Bullet-80, Seymour, CT
□ 203-834-0026	Spectre-80
□ 203-746-5763	Telcom 7, New Fairfield, CT
204	
□ 204-785-8742	Selkirk BBS, Selkirk, MB, CAN
205	
□ 205-492-0373	Bullet-80, Gadsden, AL
□ 205-272-5069	Forum-80, Montgomery, AL
□ 205-972-1685	Pentagon
□ 205-895-6749	RCP/M RBBS NACS/UAH, Huntsville, AL
206	
□ 206-935-9119	ABBS Apple Crate I, Seattle, WA
□ 206-244-5438	ABBS Apple Crate II, Seattle, WA
□ 206-866-9043	A.C.C.E.S.S., Olympia, WA
□ 206-621-8665	Anchor CP/M
□ 206-525-5410	Apple Crate I, Seattle, WA
□ 206-546-6239	ARBB, Seattle, WA
□ 206-524-0203	Call-A.P.P.L.E., Seattle, WA
□ 206-256-6624	Dial-Your-Match #16, Seattle, WA
□ 206-723-3282	Forum-80, Seattle, WA
□ 206-883-0403	JCTS Redmond, WA
□ 206-767-7777	Kingdom of Seven, Seattle, WA
□ 206-527-0897	Mail Board-82, Seattle, WA
□ 206-762-5141	Mini-Bin, Seattle, WA
□ 206-334-7394	MSG-80 Everett, WA
□ 206-743-6021	NWWCUG Edmunds, Seattle, WA
□ 206-783-9798	Pirates of Puget Sound, Seattle, WA
□ 206-486-2368	PMS Software Unlimited, Kenmore, WA
□ 206-357-7400	RCP/M Olympia, WA
□ 206-458-3086	RCP/M RBBS Yelm, Olympia, WA
□ 206-763-8879	Seacom-80, Seattle, WA
207	
□ 207-839-2337	RCP/M Programmers Anonymous, Gorham, ME
209	
□ 209-298-1328	Dial-Your-Match #26, Clovis, CA
212	
□ 212-896-0519	(?) Queens, NY
□ 212-933-9459	Bronx BBS, New York, NY
□ 212-740-5680	Bullet-80, New York, NY
□ 212-897-3392	Comm-80, Queens, NY
□ 212-991-1664	Connection-80, Manhattan, NY
□ 212-441-3755	Connection-80, Woodhaven, NY
□ 212-631-1788	Kracker's Kastle
□ 212-541-5975	MMMMM#2, New York, NY
□ 212-410-0949	Net-Works, Brooklyn, NY
□ 212-626-0375	Nybbles-80, NY
□ 212-997-2488	PMS McGraw-Hill Books, New York, NY
□ 212-255-7240	RCP/M RBBS Manhattan, New York, NY
□ 212-442-3874	Sister, Staten Island, NY
□ 212-799-4649	TCBBS Astrocom, New York, NY
□ 212-362-1040	TCBBS B.A.M.S. New York, NY
213	
□ 213-829-1140	ABBS Computer Conspiracy, Santa Monica, CA
□ 213-459-6400	ABBS Pacific Palisades, Los Angeles, CA
□ 213-537-3378	Access One, CA
□ 213-564-7636	All Night BBS, CA
□ 213-991-1604	Alpha Byte, CA
□ 213-851-0780	Aware II, Los Angeles, CA
□ 213-394-5950	BBS B.R., Los Angeles, CA
□ 213-649-1489	BBS IBM PC, Culver City, CA

□ 213-930-2578	CIA
□ 213-657-1799	Computer Connection, Los Angeles, CA
□ 213-372-4800	Conference-Tree Kelp Bed, Los Angeles, CA
□ 213-394-1505	Conference-Tree, Santa Monica, CA
□ 213-633-5463	Data-Mate, Canoga Park, CA
□ 213-346-1849	Dec-Line, Woodland Hills, CA
□ 213-842-3322	Dial-Your-Match #1, CA
□ 213-990-6830	Dial-Your-Match #22, CA
□ 213-783-2305	Dial-Your-Match #4, CA
□ 213-345-1047	Dial-Your-Match #9, CA
□ 213-347-9780	Dr. Falcon's Retreat, Canoga Park, CA
□ 213-428-5206	Dragon's Game System
□ 213-789-9512	Electric Line Connection, Sherman Oaks, CA
□ 213-840-8066	Fantasy Plaza
□ 213-287-1363	Greene Machine, Temple City, CA
□ 213-445-3591	Greene Machine, Fricaseed Chicken, Arcadia, CA
□ 213-431-1443	Greene Machine, Los Alamitos, CA
□ 213-591-7239	Groundstar System, Long Beach, CA
□ 213-366-1238	HBBS Mog-ur, Granada Hills, CA
□ 213-477-4605	Interface, Los Angeles, CA
□ 213-947-8128	Kluge Computer
□ 213-631-3186	L.A. Interchange, Los Angeles, CA
□ 213-478-5478	Master World, Los Angeles, CA
□ 213-470-5912	Mad Board From Mars, Los Angeles, CA
□ 213-390-3239	MMMMM#1, Santa Monica, CA (line One)
□ 213-450-4580	MMMMM#1, Santa Monica, CA (line Two)
□ 213-452-6111	MMMMM#3, Marina Del Rey, CA
□ 213-821-2257	MMMMM#4, Lawndale, CA
□ 213-336-5535	Net-Works Coin Games, Los Angeles, CA
□ 213-859-0894	Net-Works Computer World, Los Angeles, CA
□ 213-345-3670	Net-Works Encino, CA
□ 213-388-5198	Net-Works Magnetic Fantasies, Los Angeles, CA
□ 213-454-3075	Net-Works Pirate's Inn, CA
□ 213-473-2754	Net-Works Softworx, West Los Angeles, CA
□ 213-881-6880	Novation Co., Los Angeles, CA
□ 213-980-5643	Oracle, North Hollywood, CA
□ 213-784-0204	Outer Limits # 1, Van Nuys, CA
□ 213-782-8390	Outer Limits # 2, Van Nuys, CA
□ 213-360-0211	Phantoms Hollow, Granada Hills, CA
□ 213-472-4287	Pirates Mountain, Los Angeles, CA
□ 213-395-9813	Pirates Paper, Santa Monica, CA
□ 213-331-3574	PMS, Los Angeles, CA
□ 213-368-5801	RBBS, San Fernando, CA
□ 213-395-0460	RBBS, Santa Monica, CA
□ 213-799-1632	RCP/M CBBS, Pasadena, CA
□ 213-360-5053	RCP/M, Granada Hills, CA
□ 213-296-5927	RCP/M, Los Angeles, CA
□ 213-541-2503	RCP/M RBBS GFRN Data Exchange Palos Verdes, CA
□ 213-653-6398	RCP/M RBBS, Hollywood, CA
□ 213-973-2374	RCP/M RBBS IBM PC, Hawthorne, CA
□ 213-577-9947	RCP/M RBBS, Pasadena, CA
□ 213-447-0681	The Frigate
□ 213-375-6137	Torture Chamber, Los Angeles, CA
□ 213-357-2038	Twilight Zone
□ 213-859-2735	Ye Pawn Shoppe, Los Angeles, CA
214	
□ 214-424-3862	ABBS Dallas Info Board, Dallas, TX
□ 214-960-7654	ABBS Teledunion III, Dallas, TX
□ 214-631-7747	ABBS The Pulse, Dallas, TX
□ 214-289-1386	BBS-80 Daltrug, Dallas, TX
□ 214-644-4781	Net-Works Apple Shack, TX
□ 214-361-1386	Net-Works Dallas, TX
□ 214-239-5942	Net-Works Eclectic Computer Systems, Dallas, TX
□ 214-824-7455	Net-Works Winesap, TX
□ 214-931-8274	RCP/M CBBS, Dallas, TX
□ 214-241-1939	RCP/M CBBS Maxicom, Farmers Branch, TX
□ 214-247-5307	RCP/M CBBS Maxicom, Line 2
□ 214-769-3036	TBBS Hawkins, TX
215	
□ 215-364-2180	Bullet-80, Langhorne, PA
□ 215-855-3809	Comnet-80, North Wales, PA
□ 215-563-9815	Datanet 1200 Baud
□ 215-563-9211	Datanet 300 Baud
□ 215-434-3998	Hermes-80, Allentown, PA
□ 215-435-3388	Lehigh Press BBS, Allentown, PA
□ 215-244-0864	Net-Works Galaxy One, PA
□ 215-398-3937	RCP/M RBBS, Allentown, PA
□ 215-446-7670	Video Ace
□ 215-363-0563	Video Fantasies, Langhorne, PA
216	
□ 216-745-7855	ABBS Akron Digital Group, Akron, OH
□ 216-757-3711	BBS Computer Applications Co., Poland, OH
□ 216-729-2769	Bullet-80, Chesterland, OH
□ 216-645-0827	Comnet-80, Akron, OH
□ 216-486-4176	Forum-80, Cleveland, OH
□ 216-845-3179	Genius' Modemline
□ 216-724-2125	Infoex-80, Akron, OH
□ 216-875-4582	Micro-Com, Louisville, OH
□ 216-832-8392	PMS Massillon, OH
□ 216-867-7463	PMS Raug, Akron, OH
217	
□ 217-529-1113	Bullet-80, Springfield, IL
□ 217-877-1544	Hacker's Haven
□ 217-753-4309	MCMS Word Exchange, Springfield, IL
□ 217-429-4738	Net-Works C.A.M.S., Decatur, IL





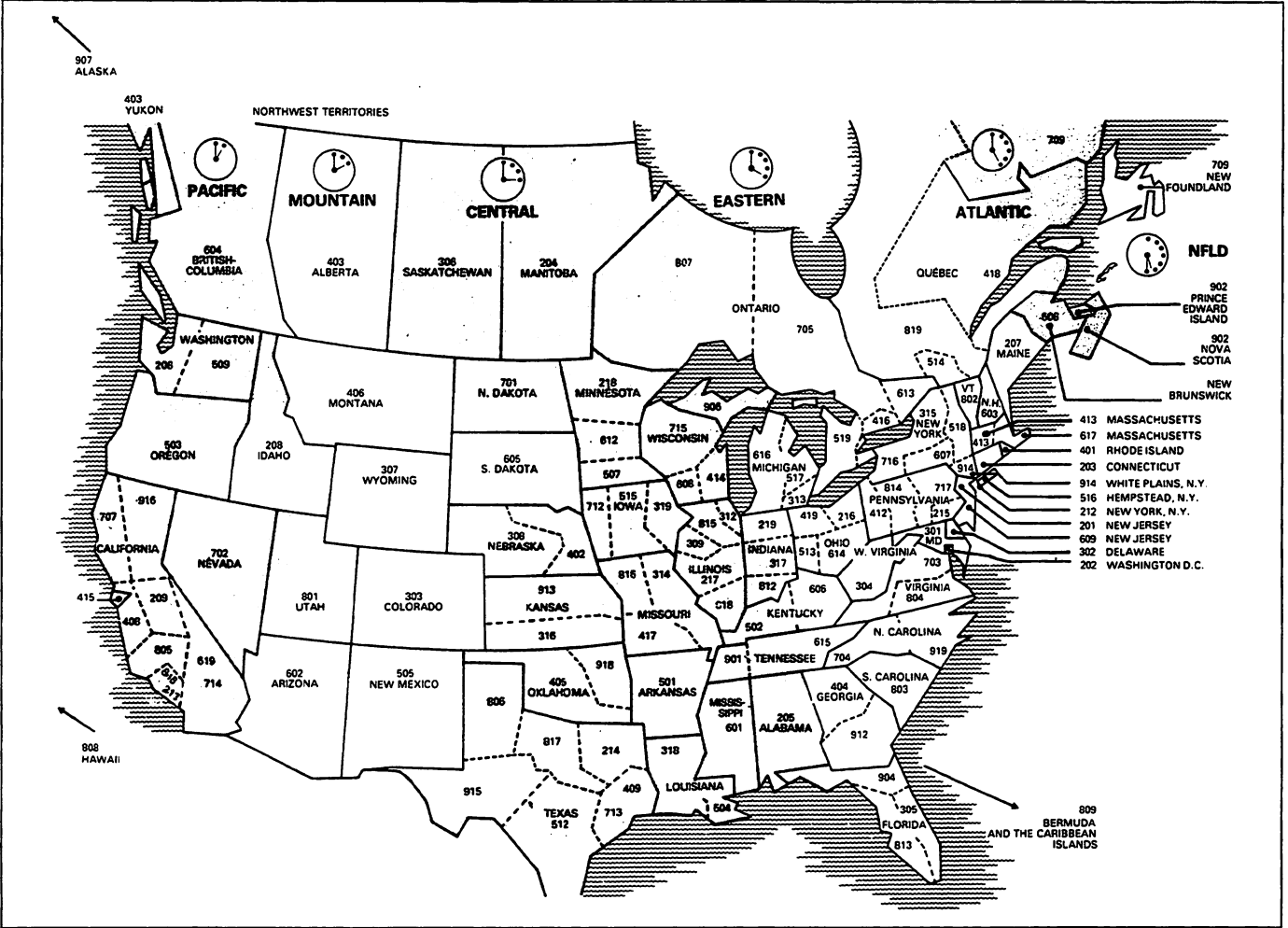






<input type="checkbox"/> 813-866-9945	CBBS St. Petersburg, FL	24h
<input type="checkbox"/> 813-977-0989	Connection-80 Tampa, FL	
<input type="checkbox"/> 813-875-3331	Micro Informer, Tampa, FL	
<input type="checkbox"/> 813-391-5219	PET BBS Commodore, Largo, FL	
<input type="checkbox"/> 813-831-7276	RCP/M RBBS Tampa, FL	
<input type="checkbox"/> 813-381-2394	Remote Northstar Largo, FL	24h
<input type="checkbox"/> 813-839-6746	Tecom-80, Tampa, FL	
<b>814</b>		
<input type="checkbox"/> 814-238-4857	RCP/M CUG-Node, PA State College	24h
<input type="checkbox"/> 814-898-2952	Trade-80 Erie, PA	24h
<b>815</b>		
<input type="checkbox"/> 815-397-4176	Cider City	
<input type="checkbox"/> 815-455-2406	Flynn's Games	
<input type="checkbox"/> 815-838-1020	MCMS J.A.M.S. Lockport, IL	24h
<b>816</b>		
<input type="checkbox"/> 816-587-9543	BBS Atari Amis, Kansas City, MO	24h
<input type="checkbox"/> 816-861-7040	Forum-80 Kansas City, MO	24h *
<input type="checkbox"/> 816-931-9316	Forum-80 Kansas City, MO	*
<input type="checkbox"/> 816-483-2526	Net-Works ABC, Kansas City, MO	
<input type="checkbox"/> 816-232-3153	Net-Works The Silver Tongue, ST. Joseph, MO	
<input type="checkbox"/> 816-252-0232	PMS Apple Bits, Kansas City, MO	24h
<b>817</b>		
<input type="checkbox"/> 817-767-5847	Comnet-80 Wichita Falls, TX	
<input type="checkbox"/> 817-665-3876	Dragonfire	
<input type="checkbox"/> 817-261-4700	Net-Works Compushop FWA, TX	
<input type="checkbox"/> 817-732-1787	Net-Works Computer Pro, Ft. Worth, TX	
<input type="checkbox"/> 817-283-3886	Texas Connection	
<b>901</b>		
<input type="checkbox"/> 901-761-4743	ABBS Computer Lab, Memphis, TN	
<input type="checkbox"/> 901-276-8196	Forum-80 Medical, Memphis, TN	24h
<b>904</b>		
<input type="checkbox"/> 904-243-1257	ABBS Fort Walton Beach, Destin, FL	
<input type="checkbox"/> 904-477-8783	BBS Pensacola, FL	
<input type="checkbox"/> 904-264-0335	Colour-80, Orange Park, FL	24h
<input type="checkbox"/> 904-353-5227	Connection-80 Jacs, Jacksonville, FL	24h
<input type="checkbox"/> 904-932-8271	Net-Works Beach BBS, Pensacola, FL	
<input type="checkbox"/> 904-743-7050	PMS Seb Computer, Jacksonville, FL	
<input type="checkbox"/> 904-725-4995	RCP/M RBBS Jug, Jacksonville, FL	24h *
<b>907</b>		
<input type="checkbox"/> 907-225-6789	ABBS, Ketchikan, AK	
<input type="checkbox"/> 907-344-5251	Conference-Tree, Anchorage, AK	
<input type="checkbox"/> 907-278-4223	Net-Works Alaska	
<input type="checkbox"/> 907-344-8558	PMS Anchorage, AK	

<input type="checkbox"/> 907-337-1984	RCP/M Anchorage, AK	
<b>912</b>		
<input type="checkbox"/> 912-233-0863	Dial-Your-Match #3	97
<input type="checkbox"/> 912-439-7440	Trade-80, Albany, GA	24h
<b>913</b>		
<input type="checkbox"/> 913-676-3613	Experimental-80, Kansas City, MO	
<input type="checkbox"/> 913-648-6071	Net-Works Leawood, KS	
<input type="checkbox"/> 913-432-5544	Online Dickinsons Movie Guide, Mission, KS	24h
<input type="checkbox"/> 913-677-1299	PMS Your Computer Connection, Kansas City, MO	
<input type="checkbox"/> 913-362-9583	RCP/M, Mission, KS	24h *
<input type="checkbox"/> 913-843-4259	RCP/M RBBS Alphanet, Lawrence, KS	24h
<input type="checkbox"/> 913-648-5301	Steve's BBS	24h
<b>914</b>		
<input type="checkbox"/> 914-634-1268	Net-Works Pirate's Lodge NY	
<input type="checkbox"/> 914-592-5385	Nybbles-80, Elmsford, NY	
<input type="checkbox"/> 914-725-4060	OSUNY, Scarsdale, NY	
<input type="checkbox"/> 914-942-2638	RACS III	
<input type="checkbox"/> 914-279-5693	RCP/M RBBS, Brewster, NY	
<input type="checkbox"/> 914-679-8734	RCP/M RBBS, Woodstock, NY	24h *
<input type="checkbox"/> 914-679-6559	RCP/M SJBBS, Bearsville, NY	24h
<input type="checkbox"/> 914-359-1517	Sherwood Forest II	
<input type="checkbox"/> 914-782-7605	ST80-PBB Monroe Camera Shop, Monroe, NY	
<input type="checkbox"/> 914-623-4248	Teleport 64	
<b>915</b>		
<input type="checkbox"/> 915-565-9903	Bullet-80, El Paso, TX	24h
<input type="checkbox"/> 915-755-1000	Forum-80, El Paso, TX	24h
<input type="checkbox"/> 915-593-6655	Net-Works El Paso, TX	
<input type="checkbox"/> 915-533-2202	RCP/M RBBS Comp. Tech. Assoc., El Paso, TX	24h
<input type="checkbox"/> 915-598-1668	RCP/M RBBS, El Paso, TX	24h *
<b>916</b>		
<input type="checkbox"/> 916-393-4459	Aviators Bulletin Board, Sacramento, CA	
<input type="checkbox"/> 916-483-8718	RCP/M CBBS, Sacramento, CA	24h
<b>918</b>		
<input type="checkbox"/> 918-838-8698	Infoex-80, Tulsa, OK	24h
<input type="checkbox"/> 918-749-0059	TBBS, Tulsa, OK	24h
<b>919</b>		
<input type="checkbox"/> 919-362-0676	Dial-Your-Match #20	97
<b>Foreign</b>		
<input type="checkbox"/> 613-762-5088	RCP/M CBBS Micom, Melbourne, VIC, Australia	24h
<input type="checkbox"/> 1 0-997-1018	RCP/M Software Tools, Sydney, Australia	24h
<input type="checkbox"/> 4-1 399-2136	CBBS, London, England	(European Standard)
<input type="checkbox"/> 44 482859169	Forum-80, Hull, England	(Country Code = 011)













# Computer Clubs

User clubs are very nomadic. The listing may show inactive clubs, but the addresses might still be useful for locating others.

## Canada

### Alberta

Calgary Commodore Users Group  
John Hazard  
37 Castlebridge Dr., N.E.  
Calgary, Alberta  
Canada T3J 1P4

CCCC (Canadian Commodore Computer Club)  
Roger Olanson  
c/o Strictly Commodore  
47 Coachwood Place  
Calgary, Alberta  
T3H 1E1  
Canada

Bonnyville VIC Cursors  
Ed Wittchen  
Box 2100  
Bonnyville, Alberta  
TOA 0L0 403-826-3992  
Canada

### British Columbia

VIC-TIMS  
Greg Goss  
2-830 Helena St.  
Trail, BC  
V1R 3X2 604-368-9970  
Canada

Castlegar Commodore Computer Club  
Robert Doolay  
SS1, S37, C7  
Castlegar, BC  
V1N 3H7 604-365-3889  
Canada

Commodore Computer Club  
PO Box 91164  
West Vancouver, BC  
V7V 3N6 604-738-3311  
Canada

### Manitoba

W.P.U.G.  
Larry Neufeld  
9-300 Enniskillen Ave.  
Winnipeg, Manitoba  
R2V 0H9  
Canada

### New Brunswick

C-64 Users Group  
Don Shea  
PO Box 9  
Rothesay, NB  
E0G 2W0  
Canada

Club 64  
Cass Howorth  
120 Liverpool St.  
Fredericton, NB  
E3B 4V5 506-454-9730  
Canada

### Nova Scotia

Nova Scotia Commodore Computer Group  
Phil Cummings  
PO Box 3426  
Halifax South  
Halifax, NS  
B3J 3J1  
Canada

### Ontario

Fledging Barrie User Group (BUG)  
58 Steel St  
Barrie, Ontario  
Canada L4M 2E9

PET Educators Group  
PO Box 454  
Station A  
Windsor, Ontario  
Canada N9A 6L7

Commodore Users Club of Sudbury  
938 Brookfield Ave.  
Sudbury, Ontario  
Canada P3A 4K4

Toronto PET Users Group, Inc.  
Chris Bennett 416-782-8900  
1912A Avenue Rd., Ste. 1  
Toronto, Ontario  
M5M 4A1 416-782-9252  
Canada

### London Commodore Users Club (LCUC)

Dennis Trankner  
28 Barrett Cres.  
London, Ontario  
N6E 1T5 519-681-5059  
Canada

Mr. Walter Scholz  
568 Mornington St.  
Stratford, Ontario  
N5A 5G9 519-271-5704  
Canada

D. Lerch  
Arva Hackers, Medway High School  
Arva, Ontario  
N0M 1C0  
Canada

Cambridge Commodore Users Group  
William McLean  
c/o Badcock & Wilcox Canada Ltd.  
581 Coronation  
Cambridge, Ontario  
N1R 5V3  
Canada

Cornwall Computer Club  
David King  
1510 Second St. East  
Cornwall, Ontario  
K6H 2C3  
Canada

Cambridge Commodore Users Group  
William McLean  
c/o Badcock & Wilcox Canada Ltd.  
581 Coronation  
Cambridge, Ontario  
N1R 5V3  
Canada

PET Users Club  
Mr. Brown  
Valley Heights Secondary School  
Box 159  
Langton, Ontario  
N0E 1G0  
Canada

C-64 Users Group  
Susan Timar  
1122 Wilson Dr.  
Sarnia, Ontario  
N7S 3J6 519-542-2534  
Canada

Brockville Users Group (B.U.G.)  
Bill Maxwell  
72 Murray St.  
Brockville, Ontario  
K6V 2X1  
Canada

### Quebec

COMVIC  
PO Box 1688  
St. Laurent  
Montreal, Quebec  
Canada H4L 4Z2

C-64 Users Group Of Montreal (C.U.G.O.M.)  
Gary Letovsky  
Snowdon PO Box 792  
Montreal, Quebec  
H3X 3K9  
Canada

### Saskatchewan

Compu-Dom of Southern Saskatchewan  
Joel Champagne  
308 Coldwell Rd.  
Regina, Saskatchewan  
S4R 4L5  
Canada

The Regina Commodore Club  
K.H. Jones  
76 Dolphin Bay  
Regina, Saskatchewan  
S4S 4Z8 584-2968  
Canada

## United States

### Alaska

Alaska 84 Computer Club  
c/o Line 49 Management  
PO Box 6043  
Anchorage, AK  
99502

COMPOOH-T  
PO Box 118  
Old Harbor, AK  
99643 907-286-2213

First City Users Group  
James Llanos  
PO Box 6692  
Ketchikan, AK  
99901 907-225-5695

1st City Users Group  
James Llanos  
PO Box 6692  
Ketchikan, AK  
99901 907-225-5695

### Alabama

Shoals Commodore Users Group (SCUG)  
G. Taylor  
209 Lakeshore Dr.  
Muscle Shoals, AL  
35661

William Autry  
1734 S. Atmore Ave.  
Whistler, AL  
36612 205-452-9740

Howard Crider  
1920-A Avenue C  
Brookly  
Mobile, AL  
36615 205-661-1973

Wiregrass Micro-Computer Society  
Bill Brown  
Commodore SIG  
109 Key Bernd Rd.  
Enterprise, AL  
36330 205-347-7564

Commodore Club of Mobile  
Tom Wyatt  
3868-H Rue Maison  
Mobile, AL  
36608 205-343-1178

CC & Me  
Bill Freeman  
PO Box 324  
Pinson, AL  
35126 205-854-0650

Riverchase Commodore Users Group  
Ken Browning  
617 Grove St.  
Birmingham, AL  
35209 205-988-1078

Tiger Byte: E. Alabama CBM 64 Users Group  
Jack Parsons  
c/o The Computer Store, Inc.  
Midway Plaza  
Opelika, AL  
36801

Huntsville PET Users Club  
Hal Carey  
9002 Berclair Rd.  
Huntsville, AL  
35802

The Birmingham Commodore Computer Club  
Harry Jones  
Birmingham, AL

### Arkansas

Booneville 64 Club  
Mary Taff  
c/o A.R. Hederich Elem. School  
401 W. 5th St.  
Booneville, AR  
72927

Commodore/PET Users Club  
Geneva Bowlin  
Conway Middle School  
Davis St.  
Conway, AR  
72032

The Siloam Commodore Computer Club  
Ken Emanuelson  
PO Box 88  
Siloam Springs, AR  
72761 501-524-5624

### Arkansas River Valley Commodore Users

Bob Brazal  
401 S. Arlington Dr.  
Russellville, AR  
72801 501-967-1868

Commodore Computer Club of Ft. Smith, AR  
Joe Ragsdale  
PO Box 6000  
So. Station  
Ft. Smith, AR  
72906

P.I.C. Club  
Bob Reed  
c/o Hatfield Public Schools  
Box 130  
Hatfield, AR  
71945 501-389-6164

### Arizona

VIC Users Group  
Paul Muffuletto  
2612 E. Covina  
Mesa, AZ  
85203

ACUG  
Dan Deacon  
c/o Home Computer Service  
2028 W. Camelback Rd.  
Phoenix, AZ  
85015 602-249-1186

Catalina Commodore Computer Club  
George Pope  
2012 Avenida Guillermo  
Tucson, AZ  
85710 602-296-6766

West Mesa VIC  
Kenneth Epstein  
2351 S. Standage  
Mesa, AZ  
85202

Arizona VIC 20-64 Users Club  
Donald Kipp  
232 W. 9th Place North  
Mesa, AZ  
85201

Central Arizona PET People  
Roy Schaher  
842 W. Calle del Norte  
Chandler, AZ  
85224 602-899-3622

Arizona VIC & 64 Users  
Tom Monson  
904 W. Marlboro Circle  
Chandler, AZ  
85224 602-963-6149

Canyon De Chelly - Four Corners Users Group  
Larry DiLucchio  
c/o Calumet Consulting  
Box 1945  
Chino, AZ  
86503 602-674-3421

### California

The Valley Computer Club  
2006 Magnolia Blvd.  
Burbank, CA  
91506

San Diego Commodore (PET) User Group  
Jane Campbell  
Box 86531  
San Diego, CA  
92138 619-277-7214

SIG (Special Interest Group)  
Brian R. Klotz  
1135 Coronet Ave.  
Pasadena, CA  
91107

Sixty Fourum  
John Damiano  
PO Box 16098  
Fresno, CA  
93755

Pomona Valley Vic Users Group  
Mark Joerger  
1401 W. 9th, #77  
Pomona, CA  
91766 714-620-8889

Valley Computer Club  
PO Box 310  
Denair, CA  
95316

### Southern California PET Users Group

c/o Data Equipment Supply Corp.  
8315 Firestone Blvd.  
Downey, CA  
90241 213-923-9361

Port Townsend Computer Club  
Doug Nash  
PO Box 233  
Port Townsend, CA  
98368

The Exchange  
Michael C. Joseph, MD  
PO Box 9189  
Long Beach, CA  
90810 213-595-1771

Walnut Creek PET Users Club  
1815 Ygnacio Valley Rd.  
Walnut Creek, CA  
94596

Jurupa Wizards  
Walter J. Scott  
8700 Galena St.  
Riverside, CA  
92509 781-1731

Robyn Graves  
8120 Sundance Dr.  
Orangevale, CA  
95662 916-969-2028

Commodore 64 West Computer Club  
Don Campbell  
2917 Colorado Ave.  
Santa Monica, CA  
90404 213-828-9308

PET on the Air  
Max J. Babin, Secretary  
525 Crestlake Dr.  
San Francisco, CA  
94132

Diablo Valley Commodore Users Group  
PO Box 27155  
Concord, CA  
94520 415-838-2838

San Fernando Valley Commodore Users Group  
Tom Lynch  
21208 Nashville  
Chatsworth, CA  
91311 213-709-4736

Antelope Valley Commodore Users Group  
James Haner  
POB 4436  
Lancaster, CA  
93539 805-942-2626

Bay Area Home Computer Asso.  
Cliff Downing  
1332 Pine St.  
Walnut Creek, CA  
94598 415-932-5447

San Francisco Commodore Users Group  
Roger Tierce  
278 - 27th Ave. #103  
San Francisco, CA  
94121 415-387-0225

Commodore Users Group  
Gilbert Vela  
4237 Phumeria Ct.  
Santa Maria, CA  
93455 805-937-4174

Commodore Users Group of Riverside (CUGR)  
Ken Brown  
PO Box 8748  
Riverside, CA  
92515 714-689-1452

Marin Commodore Computer Club  
620 Del Ganado Rd.  
San Rafael, CA

Lincoln Computer Club  
John Fung, Advisor  
750 E. Yosemite  
Manteca, CA  
95336

NVCUG  
Jim Banks, Jr.  
PO Box 1925  
Chico, CA  
95927 916-343-4611

Sacramento Commodore Users Group  
Robyn Graves  
8120 Sundance Dr.  
Orangevale, CA  
95662 916-969-2028

PALS (PETS Around Livermore Society)  
J. Johnson  
886 South K  
Livermore, CA  
94550 415-449-1084

SPHINX  
Bill MacCracken  
267 Arlington Ave.  
Kensington, CA  
94707 415-527-9286

Commodore Tech. Users Group C-TUG  
PO Box 1497  
Costa Mesa, CA  
92626

Sixty Fourum  
Deb Christensen  
4413 E. Iowa  
Fresno, CA  
93702 209-252-0392

C-64/VIC 20 Users Group  
Chuck Cypher  
Pasadena City College  
Cicadian Room  
Pasadena, CA

20/64 Users Group  
Don Cracraft  
PO Box 18473  
San Jose, CA  
95158

Peninsula Commodore Users Group  
Timothy Very  
549 Old County Rd.  
San Carlos, CA  
94070 415-593-7697

VIC-Club: San Francisco (VCSF)  
Colin Johnston  
1503A Dolores  
San Francisco, CA  
94110

Humboldt Commodore Group  
R. Turner  
c/o R. Turner  
PO Box 570  
Arcata, CA  
95521

Commodore 64 West  
Charles P. Santos  
PO Box 346  
Culver City, CA  
90232 213-398-0913

20/64  
PO Box 18473  
San Jose, CA  
95158 408-978-0546

PALS (Pets Around Livermore Society)  
John Rambo  
886 South K  
Livermore, CA  
94550

Commodore Interest Association  
Mark Finley  
c/o Computer Data  
14660 La Paz Dr.  
Victorville, CA  
92392

VIC 20 Software Exchange  
Vincent Beltz  
7660 Western Ave.  
Buena Park, CA  
90620

Software 64  
Mario Abad  
353 California Dr.  
Burlingame, CA  
94010 415-340-7115

Amateurs and Artesians Computing  
PO Box 682  
Cobb, CA  
95426

PUG of Silicon Valley  
22355 Rancho Ventura Rd.  
Cupertino, CA  
95014

VIC 20 Software Exchange Club  
Daniel Upton  
10530 Sky Circle  
Grass Valley, CA  
95945

Southern California Edison Commodore Club  
Jerry Van Norton  
PO Box 800  
Rosemead, CA  
91770

S.D. East County C-64 User Group  
Linda Schwartz  
c/o Linda Schwartz  
6353 Lake Apogka Place  
San Diego, CA  
92119 619-698-7814

Manteca VIC 20 Users Organization  
Gene Rong  
429 N. Main St.  
Manteca, CA  
95336

Suisun/Vacaville Commodore Users Group  
Charles D. Alula  
1410 Pelican Way  
Suisun City, CA  
94585 707-426-2077

Sequoia Computer Users  
Dave Demanty  
3005 Seeger Avenue  
Visalia, CA  
93277

South Bay Commodore Users Group  
Lloyd Lehrer  
401 - 9th St.  
Manhattan Beach, CA  
90266 213-374-1247

The Diamond Bar R.O.P. Users Group  
Don McIntosh  
2644 Amelgado  
Haciendo Hgts., CA  
91745 213-333-2645

CA. Area Commodore Terminal Users Society  
Darrell Hall  
C.A.C.T.U.S.  
PO Box 1277  
Alta Loma, CA  
91701

VIC TORII-The VIC 20 Users Group  
Wesley Clark  
PSC #1, Box 23467  
APO San Francisco, CA  
96230

South Bay Commodore 64 Users Group  
PO Box 3193  
San Ysidro, CA  
95073

C-64 West Orange County Users Group  
Philip Putman  
PO Box 1457  
Huntington Beach, CA  
92647 714-842-4484

Santa Rosa Commodore 64 Users Group  
Garry Palmer  
333 East Robles Ave.  
Santa Rosa, CA  
95407 707-584-7009

San Luis Obispo Commodore Computer Club  
Joan Rinehart  
1766 9th St.  
Los Osos, CA  
93402 805-528-3371

Stockton Commodore Users Group  
Andrew Smith  
2555 Alexa Way  
Stockton, CA  
95209 209-478-8419

Computer Using Educators  
Leanne Patterson  
PO Box 18547  
San Jose, CA  
95158

LOGIKS Commodore Computer Club  
Elmer Johnson  
c/o Christ Presbyterian Church  
620 Del Ganado Rd.  
San Rafael, CA  
94903 415-479-0426

Computer Bam Computer Club  
S. Mark Vanderbilt  
319 Main St.  
Suite #2  
Salinas, CA  
93901 757-0788

Napa Valley Commodore Computer Club  
Mick Winter  
c/o Liberty Computerware  
2680 Jefferson St.  
Napa, CA  
94558 707-252-6281  
night ph. 707-944-2797

The Commodore Connection  
Bud Massey  
2301 Mission St.  
Santa Cruz, CA  
95060 408-425-8054

**Colorado**

VICKIMPET Users Group  
Louis Roehrs  
4 Waring Lane, Greenwood Village  
Littleton, CO  
80121

Commodore Users Group  
Ray Brooks  
Box 377  
Aspen, CO  
81612 303-925-5604

Vicore Users Group  
Wayne Sundstrom  
326 Emery Dr.  
Longmont, CO  
80501 303-772-2821

Aurora Market Users Group  
Roger Oberdier  
c/o Computer Market Place  
15200 E. 6th Ave.  
Aurora, CO  
80012 303-367-0901

Colorado Commodore Computer Club  
Jack Moss at 986-0577  
2187 S. Golden Ct.  
Or CONTACT: John Adams at 494-0705.  
Denver, CO  
80227

**Connecticut**

John F. Garbarino  
Skill Lane Masons Island  
Mystic, CT  
06355 203-536-9789

New London County Commodore Club  
Dr. Walter Doolittle  
Doolittle Road  
Preston, CT  
06360

Fairfield County Commodore Users Group  
Linda Retter  
PO Box 212  
Danbury, CT  
06810

Commodore Users Group  
Daniel G. Spaneas  
Wethersfield High School  
411 Wolcott Hill Rd.  
Wethersfield, CT  
06109

Capitol Region Commodore Computer Club  
Prudence Schifley  
57 Carter Dr.  
Tolland, CT  
06084

VIC Users Club  
Edward Barszczewski  
22 Tunxis Rd.  
West Hartford, CT  
06107

The Commodore East Users Group  
165 B S. Bigelow Rd.  
Hampton, CT  
06247 203-455-0108

Commodore Users Group of Stratford  
Dan Kern-Ekins  
PO Box 1213  
Stratford, CT  
06497 203-377-8373

PEEK & POKE Computer Software Club  
Bob J. Pipolo  
PO Box 98, 528 Main St.  
Cromwell, CT  
06416 203-267-2113

CT Computer Society  
Harry Hill  
180 Bloomfield Ave.  
Hartford, CT  
06105 203-233-3373

**District of Columbia**

USO Computer Club  
Steven Gunther  
USO Outreach Center  
207 Beyer Rd., SW  
Washington, DC  
20332

**Delaware**

The Diamond State Users Group  
Michael Butler  
Box 892, RD 2  
Felton, DE  
19943 302-284-4495

Brandywine Users Group  
Rick Jeandell  
PO Box 10943  
Wilmington, DE  
19850 302-362-6162

Newark Commodore Users Group (NCUG)  
Bob Black  
210 Durso Dr.  
Newark, DE  
19711 302-737-4686

**Florida**

South Tampa Commodore 64 Users Group  
Ronald S. Clement  
736 F Second Dr.  
Macdill AFB, FL  
33621

Tampa Bay Commodore Computer Club  
10208 N. 30th St.  
Tampa, FL  
33612 813-977-0877

E! Shift OH  
Mike Schnoke  
PO Box 548  
Cocoa, FL  
32922

Sanibel Commodore Users Group (SCUG)  
Phil Belanger  
1119 Periwinkle  
Box 73  
Sanibel, FL  
33957 813-472-3471

The Ultimate 64 Experience  
Sandy Cueto  
5740 S.W. 56th Terrace  
Miami, FL  
33143

Tampa Commodore Users Group  
PO Box 8713  
Tampa, FL  
33674 813-237-2100

64 Educators Users Group North  
Robert Figueroa  
16330 N.E. 2nd Ave.  
North Miami Beach, FL  
33162 305-944-5548

Ram Rom 84  
Nancy Kenneally  
1620 Morning Dove Lane  
Englewood, FL  
33533 813-474-9450

Commodore Users Group  
Jim Neill  
545 E. Park Ave.  
Apt. #2  
Tallahassee, FL  
32301 904-224-6286

Lakeland VIC 20 Users Group  
West Hartford, CT  
Mulberry, FL  
33860

Brandon Users Group  
Paul Daugherty  
108 Anglemore Dr.  
Brandon, FL  
33511 813-685-5138

Brandon Commodore Users Group  
414 E. Lumsden Rd.  
Brandon, FL  
33511

64 Educators Users Group South  
Dr. Eydle Stoane  
FDLRS-South  
9220 S.W. 52nd Terrace  
Miami, FL  
33165 305-274-3501

Miami 20/64  
12911 S.W. 49th St.  
Miami, FL  
33175 305-226-1185

VIC Users Club  
Ray Thigpen  
4071 Edgewater Dr.  
Orlando, FL  
32804

PF 's and Friends  
Richard Plummer  
129 NE 44th St.  
Miami, FL  
33137

South Florida PET Users Group  
Dave Young  
7170 S.W. 11th St.  
West Hollywood, FL  
33023 305-987-6982

Commodore Computer Club  
David Phillips  
PO Box 9726  
Jacksonville, FL  
32208 904-764-5457

Commodore 64/VIC 20 User Group  
Mr. Earl Preston (305)  
Martin Marietta Aerospace  
PO Box 5837, MP 142  
Orlando, FL  
32855 352-3252/2266

Gainesville Commodore Users Club  
Louis Wallace  
3604-20A SW 31st Dr.  
Gainesville, FL  
32608

Bay Commodore Users Group  
Richard Scofield  
c/o Gulf Coast Computer Exchange  
241 N. Tyndall Pkwy., PO Box 6215  
Panama City, FL  
32401 904-785-6441

Volusia Ct. Commodore Program Exchange  
Rick Stidham  
1612 Reynolds Rd.  
DeLeon Springs, FL  
32028

Suncoast 64s  
Curtis Miller  
c/o Little Professor Book Center  
2395 U.S. 19 North  
Palm Harbor, FL  
33563 813-785-1036

VIC/64 Heartland Users Group  
Tom Keough  
1220 Bartow Rd. #23  
Lakeland, FL  
33801 813-666-2132

Charlotte County Commodore Club (CCCC)  
Lee Truax  
567 N. Ellicott Circle  
Port Charlotte, FL  
33952 813-625-1277

Broward Commodore Users Group  
Lewis Horn  
13 Spinning Wheel Lane  
Tamarac, FL  
33319 305-726-4390

Richard Prestien  
6278 SW 14th St.  
Miami, FL  
33144

Commodore Computer Club  
Chuck Fechko  
PO Box 21138  
St. Petersburg, FL  
33742 813-522-2547

The Class of 64  
Joe Statofora  
c/o The Computer Corner  
5208 - 66th St., North  
St. Petersburg, FL  
33709 813-541-1185

Jacksonville Area PET Society  
401 Monument Rd. #177  
Jacksonville, FL  
32211

Sun Coast VICs  
Mark Weddell  
PO Box 1042  
Indian Rocks Beach, FL  
33535

The Commodore Advantage  
Deanna Owens  
PO Box 18490  
Pensacola, FL  
32523 904-456-6554

Cleanwater Commodore Club  
Gary Gould  
1532 Lemon St.  
Cleanwater, FL  
33516 813-442-0770

Commodore Connection  
PO Box 6684  
West Palm Beach, FL  
33405

The Commodore Connection  
PO Box 6684  
West Palm Beach, FL  
33405

Gainesville Commodore Users Group  
James E. Birdsell  
Santa Fe Community College  
Gainesville, FL  
32602









Southwestern Ohio Commodore Users Group  
 PO Box 399117  
 Cincinnati, OH  
 45239

Licking County 64 Users Group  
 323 Schuler St.  
 Newark, OH  
 43055 614-345-1327

Commodore Users Group  
 Carl Skala  
 18813 Harlan Dr.  
 Maple Heights, OH  
 44137 216-581-3099

Dayton Area Commodore Users Group  
 Charles Tobin  
 679 Murray Hill Dr.  
 Xenia, OH  
 45385 513-372-4077

Commodore Users of Blue Chip (Cincinnati)  
 Ted Stalats  
 816 Beecher St.  
 Cincinnati, OH  
 45206 513-981-6582

**Oklahoma**

Commodore Users  
 Monte Maker, President  
 Box 268  
 Oklahoma City, OK  
 73101

Commodore Users Group  
 Steve Ford  
 Muskogee Computer Society  
 202 S. 12th St.  
 Muskogee, OK  
 74401

Commodore Users of Norman  
 Matt Hager  
 209 Brookwood  
 Noble, OK  
 73068

Southwest Oklahoma Computer Club  
 c/o Commodore Chapter  
 PO Box 6646  
 Lawton, OK  
 73504

Commodore Oklahoma Users Club  
 Stanley B. Dow  
 4000 NW 14th St.  
 Oklahoma City, OK  
 73107 405-943-1370

Commodore Hobby Users Group (CHUG)  
 Annette Hinshaw  
 Box 15238  
 Tulsa, OK  
 74158 918-834-5658

Greater Oklahoma Commodore Club  
 Randy Hill  
 1401 N. Rockwell  
 Oklahoma City, OK  
 73127 405-789-3229

**Oregon**

United States Commodore Users Group  
 Richard Tsukiji  
 PO Box 2310  
 Roseburg, OR  
 97470 503-672-7591

NW PET Users Group  
 John F. Jones  
 2134 N.E. 45th Ave.  
 Portland, OR  
 97213

US Commodore Users Group  
 Richard Tsukiji  
 1385 Cleveland Loop Dr.  
 Roseburg, OR  
 97470

Southern Oregon VIC/64 Users Group  
 James Powell  
 3600 Madrona Lane  
 Medford, OR  
 97501 503-779-7631

Jefferson State Computer Users Group-JUG  
 John Newman  
 2355 Camp Baker Rd.  
 Medford, OR  
 97501

**Pennsylvania**

G.R.C. User Club  
 Bill Bolt  
 300 Whitten Hollow Rd.  
 New Kensington, PA  
 15068

Bellwood - Altoona Users Group  
 D.N. Dantof  
 1433 - 13th Ave.  
 Altoona, PA  
 16603 814-942-9565

Commodore Users Group  
 Jim Mathers  
 3021 Ben Venue Dr.  
 Greensburg, PA  
 15601 412-836-2224

Commodore Users Group  
 Matt Malulaitis  
 781 Dick Ave.  
 Warminster, PA  
 18974

VIC 20 Programers, Inc.  
 Robert Gougher  
 c/o Watson Woods  
 115 Old Spring Rd.  
 Coatesville, PA  
 19320

Clifton Heights Users Group  
 PO Box 235  
 Clifton Heights, PA  
 19018

VIC Software Development Club  
 Tracy Lee Thomas  
 440 W. Sedgwick  
 Apt. A-1  
 Philadelphia, PA  
 19119 215-844-4328

G/C Computer Owners Group  
 Jo Lambert 215-775-2600  
 c/o Gilbert Associates, Inc.  
 PO Box 1498  
 Reading, PA  
 19607 Extention 6472

Gene Planchak  
 4820 Anne Lane  
 Sharpshill, PA  
 15150 412-962-9682

The Commodore Users Club of S.E. Pittsburgh  
 Charles Groves  
 c/o Groves Appliance & TV  
 2407 Pennsylvania Ave.  
 West Mifflin, PA  
 15122

Main Line Commodore Users Group (MLCUG)  
 Emil Volcheck  
 1046 General Allen Lane  
 West Chester, PA  
 19380 215-388-1581

Oxford Circle 64 User Group  
 Roger Nazaley 215-535-9021  
 Trinity Church  
 6900 Rising Sun Ave.  
 Philadelphia, PA  
 19111 215-743-8999

Bits & Bytes  
 Dave Boodye  
 1015 Dale Rd.  
 Secane, PA  
 19018 215-544-5875

CACC (Capitol Area Commodore Club)  
 Geoffrey Hebert  
 PO Box 333  
 Lemoynne, PA  
 17043 717-732-5255

Penn Conference Computer Club  
 Dan R. Knepp  
 c/o Penn Conference of SDA  
 720 Museum Rd.  
 Reading, PA  
 19611

PET User Group  
 Gene Beals  
 PO Box 371  
 Montgomeryville, PA  
 18936

A-K 64 Users Group  
 Alton E. Glubish  
 1762 Fairmont St.  
 New Kensington, PA  
 15068 412-335-9070

PACS Commodore Users Group  
 Stephen Longo  
 LaSalle College  
 20th & Olney Ave.  
 Philadelphia, PA  
 19141 215-951-1258

Lincoln Technical Inst.  
 Alan Karpe  
 5151 Tilghman  
 Allentown, PA

PPG (Pittsburgh PET Group)  
 Joel A. Casar, DMD  
 2015 Garrick Dr.  
 Pittsburgh, PA  
 15235 412-371-2882

Westmoreland Commodore Users Club  
 Jim Mathers  
 c/o DJ & Son Electronics  
 Colonial Plaza  
 Latrobe, PA  
 15650

Boeing Employees Personal Compute Club  
 Jim McLaughlin  
 The Boeing Vertol Co.  
 PO Box 16858  
 Philadelphia, PA  
 19142 215-522-2257

Worldwide Commodore Users Group  
 David Walter  
 PO Box 337  
 Blue Bell, PA  
 19422

Upper Buxmont C-64 Users  
 Don Roques  
 655 Bergey Rd.  
 Telford, PA  
 18969 215-723-7039

CACCC-Centre Area Commodore Computer Club  
 Bill Hillner  
 214 Computer Building  
 University Park, PA  
 16802 814-237-5912

Scranton Commodore Users Group  
 PO Box 211  
 Clarks Summit, PA  
 18411

NADC Commodore Users Club  
 Norman McCrary  
 248 Oakdale Ave.  
 Horsham, PA  
 19044

MARGA  
 Mindy Skelton  
 PO Box 76  
 Mount Holly Springs PA  
 17065 717-486-3274

COMPSTARS  
 Mike Norm  
 130 Blue Teel Circle  
 Audubon, PA  
 19403

Upper Buxmont C-64 Users  
 Don Roques  
 655 Bergey Rd.  
 Telford, PA  
 18969 215-723-7039

CACCC-Centre Area Commodore Computer Club  
 Bill Hillner  
 214 Computer Building  
 University Park, PA  
 16802 814-237-5912

Scranton Commodore Users Group  
 PO Box 211  
 Clarks Summit, PA  
 18411

NADC Commodore Users Club  
 Norman McCrary  
 248 Oakdale Ave.  
 Horsham, PA  
 19044

MARGA  
 Mindy Skelton  
 PO Box 76  
 Mount Holly Springs PA  
 17065 717-486-3274

COMPSTARS  
 Mike Norm  
 130 Blue Teel Circle  
 Audubon, PA  
 19403

**Puerto Rico**

CUJ of Puerto Rico  
 Ken Burch  
 RFD #1, Box 13  
 San Juan, PR  
 00914

VIC 20 User Group  
 Robert Morales, Jr.  
 655 Hernandez St.  
 Miramar, PR  
 00907

**Rhode Island**

Newport VIC/64 Users  
 Dr. Matt McConeghy  
 10 Maitland Ct.  
 Newport, RI  
 02840 401-849-2684

Irving B. Silverman, CPA  
 Michelle Chavani  
 160 Taunton Ave.  
 E. Providence, RI  
 02914

Commodore Users Group  
 Victor Moffett  
 c/o Data Co.  
 978 Tiogue Ave.  
 Coventry, RI  
 02816 401-828-7385

The VIC 20 Users Club  
 Tom Davey  
 Warwick, RI  
 02886

**South Carolina**

Spartanburg Commodore Users Group  
 James Pasley  
 803 Lucerne Dr.  
 Spartanburg, SC  
 29302 803-582-5897

The Charleston Computer Society  
 Jack Furr  
 PO Box 5264  
 N. Charleston, SC  
 29406 803-747-0310

Lords of BASIC  
 Robert L. Whisonant  
 PO Box 459  
 Ladson, SC  
 29456

Beaufort Technical College  
 Dean of Instruction  
 100 S. Ribaut Rd.  
 Beaufort, SC  
 29902

Commodore Computer Club of Columbia  
 Chuck Howard-Sect./Tres.  
 PO Box 2775  
 Cayce  
 West Columbia, SC  
 29171

The Executive Touch C-64 & VIC 20 Users  
 Patricia Walkins  
 208 Hwy 15  
 Myrtle Beach, SC  
 29577 448-8428

Commodore Users Society of Greenville(CUS)  
 Bo Jeanes  
 Horizon Records-Home Computers  
 347 S. Pleasantburg Dr.  
 Greenville, SC  
 29607 803-235-7922

**South Dakota**

VIC/64 Users Club  
 Larry Lundeen  
 608 West 5th  
 Pierre, SD  
 57501 605-224-4863

PET User Group  
 Jim Dallas  
 515 South Duff  
 Mitchell, SD  
 57301 605-996-8277

**Tennessee**

Memphis Commodore Users Club  
 Harry Ewart  
 2476 Redvers Ave.  
 Memphis, TN  
 38127 901-358-5823

ET 64 Users Group  
 Walt Turner  
 PO Box 495  
 Knoxville, TN  
 37901 615-966-8478

Jackson Commodore Users Group  
 Rick Crone  
 31 Carriage House Dr.  
 Jackson, TN  
 38305 901-668-8958

River City Computer  
 Hobbyists  
 Memphis, TN

Memphis Commodore Users Group  
 Harry Ewart  
 2476 Redvers Ave.  
 Memphis, TN  
 38127 901-358-5823

Nashville Commodore Users Group  
 Dave Rushing  
 PO Box 121282  
 Nashville, TN  
 37212 615-331-5408

Metro-Knoxville Commodore Users Club  
 Ed Pritchard  
 7405 Oxmoor Rd., Rt.# 20  
 Knoxville, TN  
 37931 615-938-3773

Commodore User Club  
 Metro Computer Center  
 1800 Dayton Blvd.  
 Chattanooga, TN  
 37405

**Texas**

PET Users  
 2001 Bryan Tower  
 Suite 3800  
 Dallas, TX  
 75201

CHUG (Commodore Houston Users Group)  
 John Walker  
 8738 Wildforest  
 Houston, TX  
 77088 713-999-3650

Interface Computer Club  
 M.E. Garza, President  
 814 North Sabinas  
 San Antonio, TX  
 78207

Mid-Cities Commodore Club  
 Bruce Nelson  
 413 Chisolm Trail  
 Hurst, TX  
 76053

Corpus Christi Commodores  
 Bob McKelvy  
 PO Box 6541  
 Corpus Christi, TX  
 78411 512-852-7665

PET User Group  
 John Bowen  
 Texas A & M  
 Microcomputer Club  
 Texas A & M, TX

64 Users Group  
 Stan Grodin  
 2421 Midnight Circle  
 Plano, TX  
 75075

The Great Northwest CBM 64 Users Group  
 Randy  
 6302 War Hawk Dr.  
 San Antonio, TX  
 78238 647-3881

VIC Users Group  
 3817 64th St.  
 Lubbock, TX  
 79413

Larry Williams  
 PO Box 652  
 San Antonio, TX  
 78293

Fantasy Commodore Club  
 Ed Howdershelt  
 1913 Dalworth St.  
 Grand Prairie, TX  
 75050

ICUG (Irving Commodore Users Group)  
 Robert Hayes  
 3237 Northgate #1289  
 Irving, TX  
 75062 214-252-7017

Commodore Users Group  
 Danny Miller  
 624 Bellview St.  
 Sulphur Springs, TX  
 75482

VIC 20 Users Group  
 Jeff Southerland  
 6416 Brookhaven Trail  
 Ft. Worth, TX  
 76133 817-346-1407

Compugild  
 Johnathan Witt  
 2211 South Lipscomb  
 Amarillo, TX  
 79109

Mid-Cities Commodore Club  
 Garry Wordelman  
 413 Chisolm Trail  
 Hurst, TX  
 76053

SCOPE  
 PO Box 3095  
 Richardson, TX  
 75083

Gulf Coast Commodore Users Group  
 Lawrence Hernandez  
 PO Box 128  
 Corpus Christi, TX  
 78403 512-887-4577

James Meeker  
 1110 Texas Ave.  
 Mart, TX  
 76664 817-876-2710

The Woodlands Commodore Users Group  
 Andrew Gardner  
 3 Splitrock Rd.  
 The Woodlands, TX  
 77380 713-292-8987

Savid Computer Club  
 Davi Jordan, Chairman  
 312 West Alabama  
 Suite 2  
 Houston, TX  
 77006

Commodore Users Group (Austin)  
 Dr. Jerry D. Frazee  
 PO Box 49138  
 Austin, TX  
 78765

64 Users Group  
 S. G. Grodin  
 2421 Midnight Circle  
 Plano, TX  
 75075



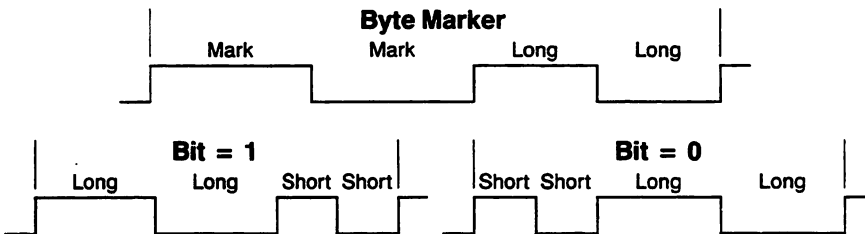
# IEEE Standard Definitions

Capitalized Mnemonics represent interface states and remote messages, lowercase represent local messages received. From "IEEE Std 488-1978".

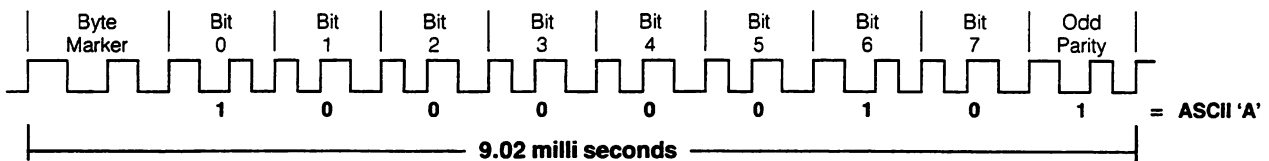
Name	Definition	Name	Definition	Name	Definition
<b>AC</b>	Addressed command	<b>L or LE</b>	Listener or extended listener	<b>RWLS</b>	Remote With Lockout State
<b>ACDS</b>	Accept data state	<b>LACS</b>	Listener active state	<b>SACS</b>	System control active state
<b>ACG</b>	Addressed command group	<b>LADS</b>	Listener Addressed State	<b>SCG</b>	Secondary Command Group
<b>ACRS</b>	Acceptor ready state	<b>LAG</b>	Listener Address Group	<b>SDC or (SDC)</b>	Selected Device Clear
<b>AD</b>	Addressed	<b>LIDS</b>	Listener idle state	<b>SDYS</b>	Source delay state
<b>AH</b>	Acceptor handshake	<b>LLO</b>	Local lockout	<b>SE</b>	Secondary
<b>AH1</b>	Complete capability	<b>LOCS</b>	Local state	<b>SGNS</b>	Source generate state
<b>AH0</b>	No capability	<b>lon</b>	Listener only	<b>SH</b>	Source Handshake
<b>AIDS</b>	Acceptor idle state	<b>LPAS</b>	Listener Primary Addressed State	<b>SIAS</b>	System central interface clear active state
<b>ANRS</b>	Acceptor not ready state	<b>(lpe)</b>	Local Poll Enable	<b>sic</b>	Send Interface Clear
<b>ANSI</b>	American National Standard's Institute	<b>LPIS</b>	Listener Primary Idle State	<b>SIDS</b>	Source idle state
<b>APRS</b>	Affirmative Poll Response State	<b>ltn</b>	Listen	<b>SIIS</b>	System control interface clear idle state
<b>ATN</b>	Attention	<b>lun</b>	Local unlisten	<b>SINS</b>	System control interface clear not active state
<b>AWNS</b>	Acceptor Wait for New cycle State	<b>LWLS</b>	Local With Lockout State	<b>SIWS</b>	Source Idle Wait State
<b>C</b>	Controller	<b>M</b>	Multiline	<b>SNAS</b>	System control not active state
<b>CACS</b>	Controller addressed state	<b>MLA or (MLA)</b>	My Listen Address	<b>SPAS</b>	Serial Poll Active State
<b>CADS</b>	Controller idle state	<b>MSA or (MSA)</b>	My Secondary Address	<b>SPD</b>	Serial Poll Disable
<b>CAWS</b>	Controller active wait state	<b>MTA or (MTA)</b>	My Talk Address	<b>SPE</b>	Serial Poll Enable
<b>CIDS</b>	Controller idle state	<b>nba</b>	New Byte Available	<b>SPIS</b>	Serial Poll Idle State
<b>CPPS</b>	Controller parallel poll state	<b>NDAC</b>	Not Data Accepted	<b>SPMS</b>	Serial Poll Mode State
<b>CPWS</b>	Controller parallel wait state	<b>NPRS</b>	Negative Poll Response State	<b>SR</b>	Service Request
<b>CSBS</b>	Controller standby state	<b>NRFD</b>	Not Ready For Data	<b>SRAS</b>	System control remote enable active state
<b>CSNS</b>	Controller service not requested state	<b>NUL</b>	Null byte	<b>sre</b>	Send Remote Enable
<b>CSRS</b>	Controller service requested state	<b>OSA</b>	Other Secondary Address	<b>SRIS</b>	System control remote enable idle state
<b>CSWS</b>	Controller synchronous wait state	<b>OTA</b>	Other Talk Address	<b>SRNS</b>	System control remote enable not active state
<b>CTRS</b>	Controller transfer state	<b>PAÇS</b>	Parallel poll addressed to configure state	<b>SRQ</b>	Service request
<b>DAB</b>	Data byte	<b>PCG</b>	Primary Command Group	<b>SRQS</b>	Service request state
<b>DAC</b>	Data accepted	<b>POFS</b>	Power off	<b>ST</b>	Status
<b>DAV</b>	Controller Data valid	<b>pon</b>	Power on	<b>STB</b>	Status Byte
<b>DC</b>	Device clear	<b>PP</b>	Parallel Poll	<b>STRS</b>	Source Transfer State
<b>DCAS</b>	Device clear active state	<b>PPAS</b>	Parallel Poll Active State	<b>SWNS</b>	Source wait for new cycle state
<b>DCIS</b>	Device clear idle state	<b>PPC</b>	Parallel Poll configure	<b>T or (TE)</b>	Talker or extended talker
<b>DCL</b>	Device clear	<b>PPD or (PPD)</b>	Parallel Poll Disable	<b>T</b>	Active true
<b>DD</b>	Device Dependent	<b>PPE or (PPE)</b>	Parallel Poll Enable	<b>(T)</b>	Passive True
<b>DIO</b>	Data input	<b>PPIS</b>	Parallel Poll Idle State	<b>TACS</b>	Talker active state
<b>DT</b>	Device trigger	<b>PPR</b>	Parallel Poll Response	<b>TADS</b>	Talker addressed state
<b>DTAS</b>	Device Trigger Active State	<b>PPSS</b>	Parallel Poll Standby State	<b>TAG</b>	Talk Address Group
<b>DTIS</b>	Device trigger state	<b>PPU</b>	Parallel Poll Unconfigure	<b>tca</b>	Take Control Asynchronously
<b>END</b>	End	<b>PUCS</b>	Parallel poll unaddressed to configure state	<b>tcs</b>	Take Control Synchronously
<b>EOI</b>	End Or Identity	<b>rdy</b>	Ready (for next message)	<b>TCT or (TCT)</b>	Take control
<b>EOS</b>	End Of String	<b>REMS</b>	Remote state	<b>TIDS</b>	Talker idle state
<b>F</b>	Active false	<b>REN</b>	Remote enable	<b>ton</b>	Talk only
<b>(F)</b>	Passive False	<b>RFD</b>	Ready For Data	<b>TPAS</b>	Talker Primary Addressed State
<b>GET</b>	Group Execute Trigger	<b>RL</b>	Remote Local	<b>U</b>	Uniline message
<b>GTL</b>	Go To Local	<b>rpp</b>	Request Parallel Poll	<b>UC</b>	Universal Command
<b>gts</b>	Go To Standby	<b>RQS</b>	Request service	<b>UCG</b>	Universal Command Group
<b>IDY</b>	Identify	<b>rsc</b>	Request System Control	<b>UNL</b>	Unlisten
<b>IFC</b>	Interface clear	<b>rsv</b>	Request service	<b>UNT</b>	Untalk
<b>ist</b>	Individual status	<b>rtl</b>	Return To Local		

## Tape Recording Format

- Leader** = 50 cycles of shorts
- Mark** = 342 micro seconds of 1.46 KHz half cycle
- Short** = 182 micro seconds of 2.75 KHz half cycle
- Long** = 262 micro seconds of 1.91 KHz half cycle



### Recorded Byte



### Program File

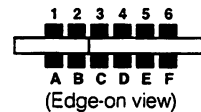
Leader	Header (192 Bytes)	Repeated Header	Program	Repeated Program	End (192 Bytes)	Repeated End
--------	--------------------	-----------------	---------	------------------	-----------------	--------------

### Tape File Format

### Data File

Leader	Header (192 Bytes)	Repeated Header	Data Block (192 Bytes)	Repeated Data Block	Data Block	Repeated Data Block (etc. to end of file)	End (192 Bytes)	Repeated End
--------	--------------------	-----------------	------------------------	---------------------	------------	---	-----------------	--------------

## Cassette Port



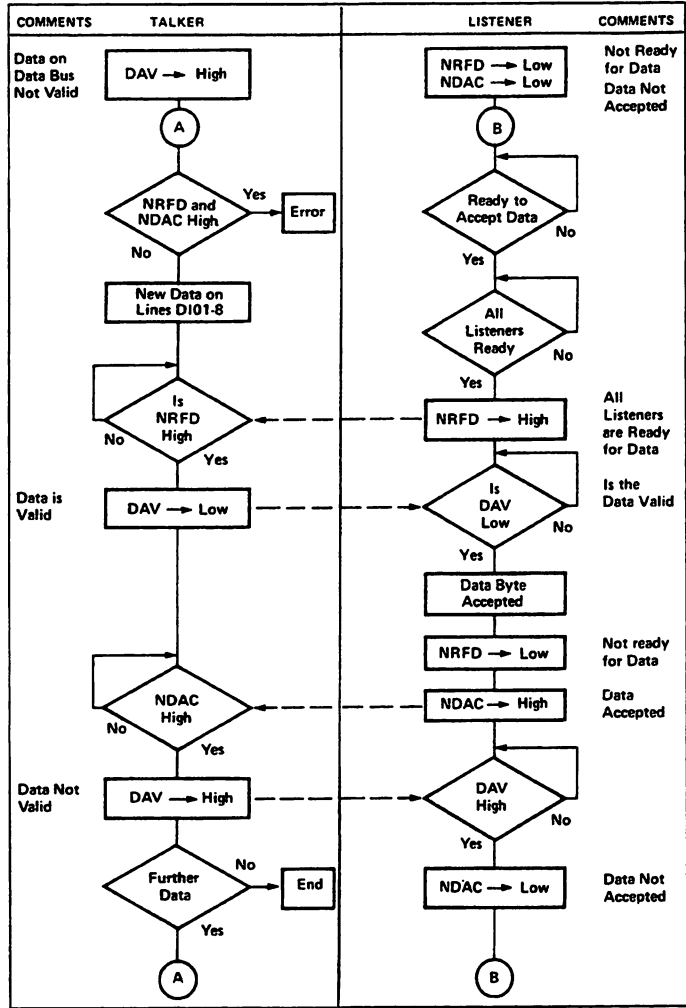
Pin#	Name	Description
A-1	GND	Digital Ground
B-2	+5V	+ 5 Volts to operate cassette circuitry only
C-3	Motor	Computer controlled + 6V for cassette motor
D-4	Read	Read line from cassette
E-5	Write	Write line cassette
F-6	Sense	Monitors closure of any locking type cassette switch

Note: Upper and Lower cassette pins are shorted

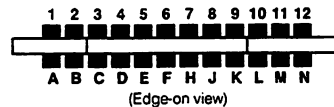
# IEEE 488 Bus Signals

Manager	ATN	Attention	The controller (PET/CBM/B) sets this signal low while it is sending commands on the data bus. When ATN is low, only peripheral addresses and control messages are on the data bus. When ATN is high, only previously assigned devices can transfer data.
Transfer	DAV	Data Valid	When DAV is low, this signifies that data is valid on data bus.
Manager	EOI	End or Identify	When the last byte of data is being transferred, the talker has the option of setting EOI low. The controller always sets EOI low while the last data byte is being transferred from the controller.
Manager	IFC	Interface Clear	The controller sends its internal reset signal as IFC low (true) to initialize all devices to the idle state. When the controller is switched on or reset, IFC goes low for about 100 milliseconds.
Transfer	NDAC	Data Not Accepted	This signal is held low (true) by the listener while reading. When the data byte has been read, the listener sets NDAC high. This signals the talker that data has been accepted.
Transfer	NRFD	Not Ready for Data	When NRFD is low (true), one or more listeners are not ready for the next byte of data. When all devices are ready, NRFD goes high.
Manager	SRQ	Service Request	Not implemented in BASIC, but available to the user.
Manager	REN	Remote Enable	REN is held low by the bus controller. The PET/CBM has a pin grounded that keeps REN permanently low.
Data	D101-8	Data Input/Output Lines 1-8	These signals represent the bits of information on the data bus. When a D10 signal is low, it represents 1 and when high 0.
General	GND	Ground	Ground connections: There are six control and management signal ground returns, one data signal ground return and one chassis shield ground lead.

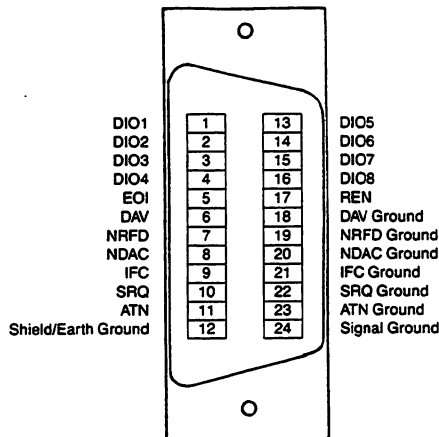
# IEEE Byte Transfer Sequence



# IEEE Port Pinouts



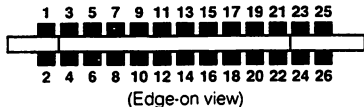
# IEEE Connectors Pins



Pin #	Pin#*	Mnemonic	Definition
1	1	DIO1	Data Input/Output Line #1
2	2	DIO2	Data Input/Output Line #2
3	3	DIO3	Data Input/Output Line #3
4	4	DIO4	Data Input/Output Line #4
5	5	EOI	End or Identify
6	6	DAV	Data Valid
7	7	NRFD	Not Ready For Data
8	8	NDAC	Data not Accepted
9	9	IFC	Interface Clear
10	10	SRQ	Service Request
11	11	ATN	Attention
12	12	GND	Chassis Ground (IEEE cable shield)
A	13	DIO5	Data Input/Output Line #5
B	14	DIO6	Data Input/Output Line #6
C	15	DIO7	Data Input/Output Line #7
D	16	DIO8	Data Input/Output Line #8
E	17	REN	Remote Enable
F	18	GND	DAV Ground
H	19	GND	NRFD Ground
J	20	GND	NDAC Ground
K	21	GND	IFC Ground
L	22	GND	SRQ Ground
M	23	GND	ATN Ground
N	24	GND	Data Ground (DIO1-8)

\* Pin Numbers for Standard IEEE Cable Connector

## PET/CBM User Port



Pin#	Function	Description
1	Ground	System Ground
2	TV Video	Video Out for external displays
3	SRQ	Connected to IEEE SRQ
4	EOI	Connected to IEEE EOI
5	Diag Sense	Held low causes power up to Diagnostic routines
6	READ 1	Connected to cassette 1 read line
7	READ 2	Connected to cassette 2 read line
8	Write	Diagnostic tape write verify
9	Vert	TV Vertical for external displays
10	Horiz	TV Horizontal for external displays
11	GND	
12	GND	
A	GND	
B	CA1	Edge sensitive input of 6522 VIA
C	PB0	PB0-7 are independently programmable for Input or Output
D	PB1	
E	PB2	
F	PB3	
H	PB4	
J	PB5	
K	PB6	
L	PB7	
M	CB2	Special I/O pin of VIA
N	GND	Digital Ground

## Commodore 64 User Port



Pin#	Function	Description
1	Ground	System Ground
2	+5V	(100 ma maximum)
3	RESET	Cold Start. Memory is NOT destroyed
4	CNT1	Serial Port counter from CIA #1
5	SP1	Serial Port from CIA #1
6	CNT2	Serial Port counter from CIA #2
7	SP2	Serial Port from CIA #2
8	PC2	Handshaking line from CIA #2
9	Serial ATN	Connected to Serial Bus ATN Line
10	9 VAC +Phase	Transformer output (50 ma. maximum)
11	9 VAC -Phase	Transformer output (50 ma. maximum)
12	GND	
A	GND	
B	FLAG2	
C	PB0	PB0-7 are independently programmable for Input or Output
D	PB1	
E	PB2	
F	PB3	
H	PB4	
J	PB5	
K	PB6	
L	PB7	
M	PA2	Special I/O pin of CIA
N	GND	

## C64 / VIC 20 Keyboard Matrix

ROW	Column (bit in location 56321)							
	7	6	5	4	3	2	1	0
\$FE	dn	F5	F3	F1	F7	rt	rtrn	DEL
\$FD	l. shft	E	S	Z	4	A	W	3
\$FB	X	T	F	C	6	D	R	5
\$F7	V	U	H	B	8	G	Y	7
\$EF	N	O	K	M	0	J	I	9
\$DF	:	@	=	.	-	L	P	+
\$BF	/	↑	:	r. shf	HOME	:	.	⌂
\$7F	STOP	Q	C=	SPACE	2	CTRL	←	1

Notes:  
 1) The Shift Lock Key is connected to the left shift key.  
 2) The RESTORE Key is not part of the keyboard matrix, but is directly wired to generate an NMI Interrupt when struck.

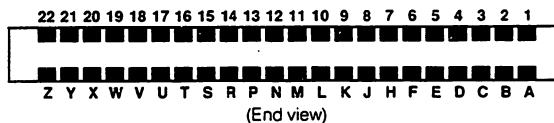
## 6522 Registers

2 8-Bit I/O Ports, 4 Control Lines, 2 16-Bit Counter/Timers, 1 8-Bit Shift Register

Reg#	Register Function
0	I/O Port B Data register
1	I/O Port A Data register, with handshaking
2	I/O Port B Data Direction
3	I/O Port A Data Direction
4	Read: Timer 1 Counter low. Resets T1 Int. Flag (IFR Bit6) Write: Timer 1 Latch low. T1 Latch low xferred to T1 Counter low on writin Reg 5
5	Read: Timer 1 Counter high. Write: Timer 1 Latch high. Latch high transferred to T1 on writing
6	Write: Timer 1 Latch low. Contents transferred to Reg 4 Read: Timer 1 Latch low. Does not reset T1 Int. Flag
7	Write: Timer 1 Latch high. Start up value, no transfer Read: Timer 1 Latch high.
8	Write: Timer 2 low. Read: Timer 2 low.
9	Write: Timer 2 high. Transfers T2 Latch low to T2 Counter low. Resets T2 Int. Flag (IFR Bit5)
10	Serial I/O shift register. Shift OUT: Bit 7 first out, then rotated to Bit 0 Shift IN: Bit 0 loaded first, rotated towards Bit 7
11	Auxiliary Control register
12	Peripheral Control register
13	Interrupt Flag Register (IFR)
14	Interrupt Enable Register (IER)
15	I/O Port A Data, no handshaking

DDRA/B: Bit = 0 Input, Bit = 1 Output (Remember: NOT I/O)

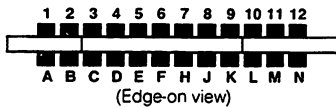
## Commodore 64 Expansion Port



Pin#	Name	Description
1	GND	System Ground.
2	+5 VDC	Total User Port and Cartridge devices
3	+5 VDC	can draw no more than 450ma.
4	IRQ	Interrupt Request line to 6510 (active low).
5	R/W	Read/Write.
6	Dot	
	Clock	8.18 MHz video dot clock.
7	I/O 1	I/O Block 1 @ \$DE00-\$DEFF (active low) unbuffered I/O.
8	GAME	Active low TTL input.
9	EXROM	Active low TTL input.
10	I/O 2	I/O Block 2 @ \$DF00-\$DFFF (active low) buffered TTL output.
11	ROM L	8K decoded RAM/ROM block @ \$8000 (active low) buffered TTL output.
12	BA	Bus Available signal from the VIC II chip - unbuffered - 1 is maximum load.
13	DMA	Direct Memory Access request line (active low input) is TTL input.
14	D7	Data bus bit 7 *
15	D6	Data bus bit 6 *
16	D5	Data bus bit 5 *
17	D4	Data bus bit 4 *
18	D3	Data bus bit 3 *
19	D2	Data bus bit 2 *
20	D1	Data bus bit 1 *
21	D0	Data bus bit 0 *
21	GND	System ground.
A	GND	System Ground
B	ROM H	8K decoded RAM/ROM Block @ \$E000 buffered.
C	RESET	6510 RESET pin (active low) buffered TTL out/unbuffered in.
D	NMI	6510 Non-Maskable Interrupt (active low) buffered TTL out, unbuffered in.
E	φ2	Phase 2 system clock.
F	A15	Address bus bit 15 *
H	A14	Address bus bit 14 *
J	A13	Address bus bit 13 *
K	A12	Address bus bit 12 *
L	A11	Address bus bit 11 *
M	A10	Address bus bit 10 *
N	A9	Address bus bit 9 *
P	A8	Address bus bit 8 *
R	A7	Address bus bit 7 *
S	A6	Address bus bit 6 *
T	A5	Address bus bit 5 *
U	A4	Address bus bit 4 *
V	A3	Address bus bit 3 *
W	A2	Address bus bit 2 *
X	A1	Address bus bit 1 *
Y	A0	Address bus bit 0 *
Z	GND	System Ground

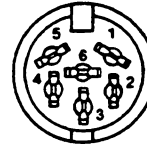


## VIC 20 User Port



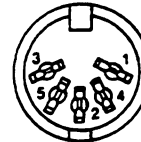
Pin#	Name	Description
1	Ground	System Ground
2	+5V	(100 ma maximum)
3	RESET	Cold Start. Memory is destroyed
4	JOY 0	Joystick Switch 0
5	JOY 1	Joystick Switch 1
6	JOY 2	Joystick Switch 2
7	PEN	Light Pen Input. Also Joystick Fire Button
8	SENSE	Cassette Switch sense line
9	Serial ATN	Connected to Serial Bus ATN Line
10	9 VAC +Phase	Transformer output (50 ma. maximum)
11	GND	
12	GND	
A	GND	
B	CB1	
C	PB0	PB0-7 are independently programmable for Input or Output
D	PB1	
E	PB2	
F	PB3	
H	PB4	
J	PB5	
K	PB6	
L	PB7	
M	CB2	Special I/O pin of VIA
N	GND	

## VIC 20 / Commodore 64 Serial Port



Pin#	Name	Description
1	SRQ	Serial SRQ in (active low)
2	GND	System Ground
3	ATN	Serial ATN In/Out
4	CLK	Serial Clock In/Out
5	DATA	Serial Data In/Out
6	RESET	Resets all devices on Serial bus (active low)

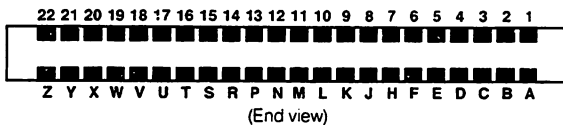
## VIC 20 Audio/Video Port



Pin#	Name	Description	Colour
1	+5V	10 ma. maximum	Red
2	GND	System Ground	-
3	AUD	Audio Out	Grey
4	VID L	Video Low	Black
5	VID H	Video High	White

Colour refers to Radio Shack Part# 42-2394

## VIC 20 Expansion Port



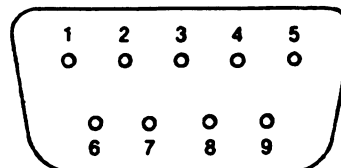
Pin#	Name	Description
1	GND	System ground
2	CD0	Data bus bit 0 *
3	CD1	Data bus bit 1 *
4	CD2	Data bus bit 2 *
5	CD3	Data bus bit 3 *
6	CD4	Data bus bit 4 *
7	CD5	Data bus bit 5 *
8	CD6	Data bus bit 6 *
9	CD7	Data bus bit 7 *
10	BLK1	8k decoded RAM/ROM block 1 @ \$2000 (active low)
11	BLK2	8k decoded RAM/ROM block 2 @ \$4000 (active low)
12	BLK3	8k decoded RAM/ROM block 3 @ \$6000 (active low)
13	BLK5	8k decoded ROM block 5 @ \$A000 (active low)
14	RAM1	1k decoded RAM block @ \$0400 (active low)
15	RAM2	1k decoded RAM block @ \$0800 (active low)
16	RAM3	1k decoded RAM block @ \$0C00 (active low)
17	V R/W	Read/Write line from VIC Chip (high-read, low-write)
18	C R/W	Read/Write line from CPU (high-read, low-write)
19	IRQ	Interrupt Request line to 6502 (active low)
20	NC	
21	+5v	
22	GND	
A	GND	
b	CA0	Address bus bit 0 *
C	CA1	Address bus bit 1 *
D	CA2	Address bus bit 2 *
E	CA3	Address bus bit 3 *
F	CA4	Address bus bit 4 *
H	CA5	Address bus bit 5 *
J	CA6	Address bus bit 6 *
K	CA7	Address bus bit 7 *
L	CA8	Address bus bit 8 *
M	CA9	Address bus bit 9 *
N	CA10	Address bus bit 10 *
P	CA11	Address bus bit 11 *
R	CA12	Address bus bit 12 *
S	CA13	Address bus bit 13 *
T	I/O 2	I/O block 2 (located at \$9600)
U	I/O 3	I/O block 3 (located at \$9C00)
V	Φ02	Phase 2 system clock
W	NMI	6502 Non-Maskable Interrupt (active low)
X	RESET	6502 Reset pin (active low)
Y	NC	
Z	GND	

\* = Unbuffered, 1 low power Schottky TTL load max.

## Commodore 64 Audio/Video Port

Pin#	Name	Description
1	LUM	Luminance
2	GND	System Ground
3	AUD	Audio Out
4	COMP	Composite Video
5	JACK	Audio In
6	CHR	Chroma out
7	N/C	No connection
8	N/C	No connection

## VIC 20 / Commodore 64 Joystick Ports



Pin#	Name	Description
1	JOY 0	
2	JOY 1	
3	JOY 2	
4	JOY 3	
5	POT Y	
6	FIRE	Also the Light Pen input. (C64 port 1 only)
7	+5V	100 ma. maximum
8	GND	System Ground
9	POT X	

Note: See Memory Map for reading Joystick Ports

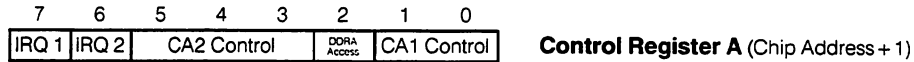
# 6520 PIA Registers

2 8-Bit I/O Ports, 4 Control Lines.  
Control Register Bit 2 is used to select Data or Direction Registers

Reg#	CRA Bit 2 =	Register Function
0	0	I/O Port A Data Direction Register (DDRA)
0	1	Peripheral I/O Port A Data register (PA)
1		Control Register A (CRA)
Reg#	CRB Bit 2 =	Register Function
2	0	I/O Port B Data Direction Register (DDRB)
2	1	Peripheral I/O Port B Data register (PB)
3		Control Register B (CRB)

DDRA/B: Bit = 0 Input, Bit = 1 Output (Remember: NOT I/O)

## PIA Control Registers

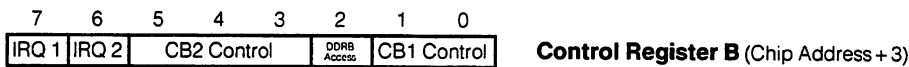


0	0	IRQ1 = 1 on CA1 going low, $\overline{IRQ}$ disabled
0	1	IRQ1 = 1 on CA1 going low, $\overline{IRQ}$ enabled
1	0	IRQ1 = 1 on CA1 going high, $\overline{IRQ}$ disabled
1	1	IRQ1 = 1 on CA1 going high, $\overline{IRQ}$ enabled
IRQ1 cleared by reading Peripheral I/O Register A		

0	Chip Address + 0 = DDRA
1	Chip Address + 0 = Peripheral I/O Port A

0	0	0	IRQ2 = 1 on CA2 going low, $\overline{IRQ}$ disabled
0	0	1	IRQ2 = 1 on CA2 going low, $\overline{IRQ}$ enabled
0	1	0	IRQ2 = 1 on CA2 going high, $\overline{IRQ}$ disabled
0	1	1	IRQ2 = 1 on CA2 going high, $\overline{IRQ}$ enabled
IRQ2 cleared by reading Peripheral I/O Register A			
1	0	0	CA2 set on IRQ1 = 1 (CA1 transition)
1	0	0	CA2 cleared after PA read
1	0	1	CA2 set on every $\Phi 2$ negative edge
1	0	1	CA2 goes low for 1 $\Phi 2$ cycle after PA read
1	1	0	Manual control: CA2 low
1	1	1	Manual control: CA2 high

X	X	IRQ Flags Set by external events
X	X	IRQ Flags Cleared by reading I/O ports



0	0	IRQ1 = 1 on CB1 going low, $\overline{IRQ}$ disabled
0	1	IRQ1 = 1 on CB1 going low, $\overline{IRQ}$ enabled
1	0	IRQ1 = 1 on CB1 going high, $\overline{IRQ}$ disabled
1	1	IRQ1 = 1 on CB1 going high, $\overline{IRQ}$ enabled
IRQ1 cleared by reading Peripheral I/O Register B		

0	Chip Address + 2 = DDRB
1	Chip Address + 2 = Peripheral I/O Port B

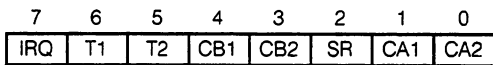
0	0	0	IRQ2 = 1 on CB2 going low, $\overline{IRQ}$ disabled
0	0	1	IRQ2 = 1 on CB2 going low, $\overline{IRQ}$ enabled
0	1	0	IRQ2 = 1 on CB2 going high, $\overline{IRQ}$ disabled
0	1	1	IRQ2 = 1 on CB2 going high, $\overline{IRQ}$ enabled
IRQ2 cleared by reading Peripheral I/O Register B			
1	0	0	CB2 set on IRQ1 = 1 (CB1 transition)
1	0	0	CB2 cleared after PB write
1	0	1	CB2 set on every $\Phi 2$ negative edge
1	0	1	CB2 goes low for 1 $\Phi 2$ cycle after PB write
1	1	0	Manual control: CB2 low
1	1	1	Manual control: CB2 high

X	X	IRQ Flags Set by external events
X	X	IRQ Flags Cleared by reading I/O ports

## Auxiliary Control Register (Chip Address + 11)

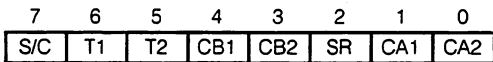
7	6	5	4	3	2	1	0	
Timer 1 Ctrl		CA2 Control	Shift Reg Control		Latch Ctrl			
				0	0	PA Latch disabled, PB Latch disabled		
				1	1	PA Latch enabled, PB Latch enabled		
				0	0	Shift Register disabled		
				0	0	Shift IN: shift rate controlled by Timer 2		
				0	1	Shift IN: shift rate controlled by $\Phi 2$		
				0	1	Shift IN: shift rate controlled by External Clock source		
				1	0	Shift OUT: Free-Running Mode, rate controlled by Timer 2		
				1	0	Shift OUT: rate controlled by Timer 2		
				1	1	Shift OUT: rate controlled by $\Phi 2$		
				1	1	Shift OUT: rate controlled by External Clock source		
				0	Decrement Counter 2 at $\Phi 2$ clock rate (in one-shot mode)			
				1	Decrement Counter 2 on pulses from PB6			
				0	One-Shot Mode			
				1	Free-Running Mode			

0	PB7 disabled
1	PB7 enabled



## Interrupt Flag Register (Chip Address + 13)

Flag Set	Flag Cleared
Transition at CA2	Reading/Writing I/O Port A
Transition at CA1	Reading/Writing I/O Port A
8 Bits Shifted IN/OUT	Reading/Writing Shift Reg
Transition at CB2	Reading/Writing I/O Port B
Transition at CB1	Reading/Writing I/O Port B
Timer 2 Timeout	Reading T2 low / Writing T2 High
Timer 1 Timeout	Reading T1 low / Writing T1 High
Interrupt Occuring	Clearing any interrupt



## Interrupt Enable Register (Chip Address + 14)

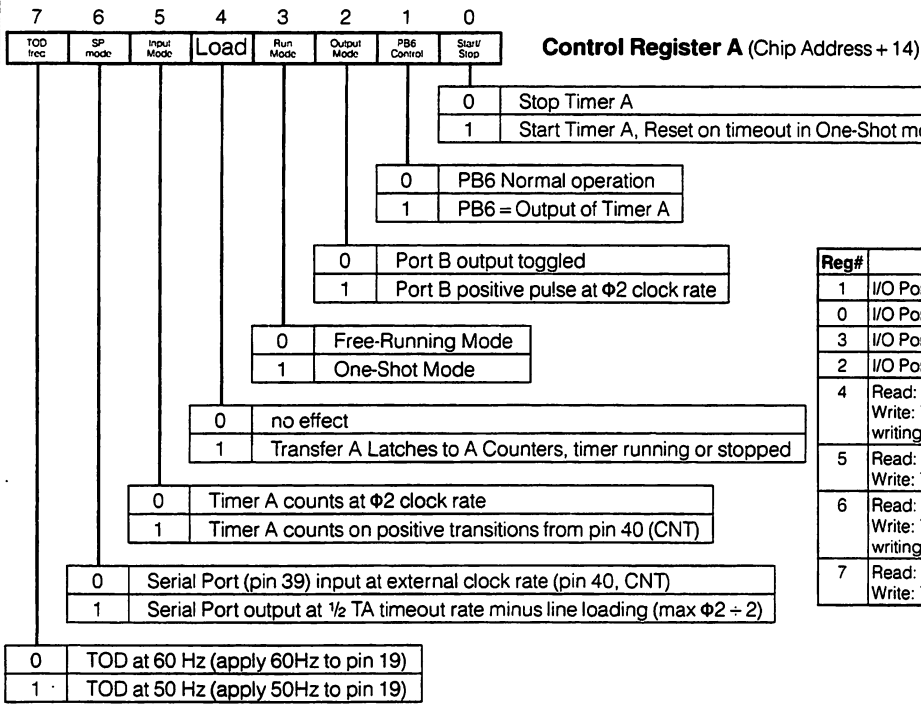
0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1

Interrupt Disabled  
Interrupt Enabled

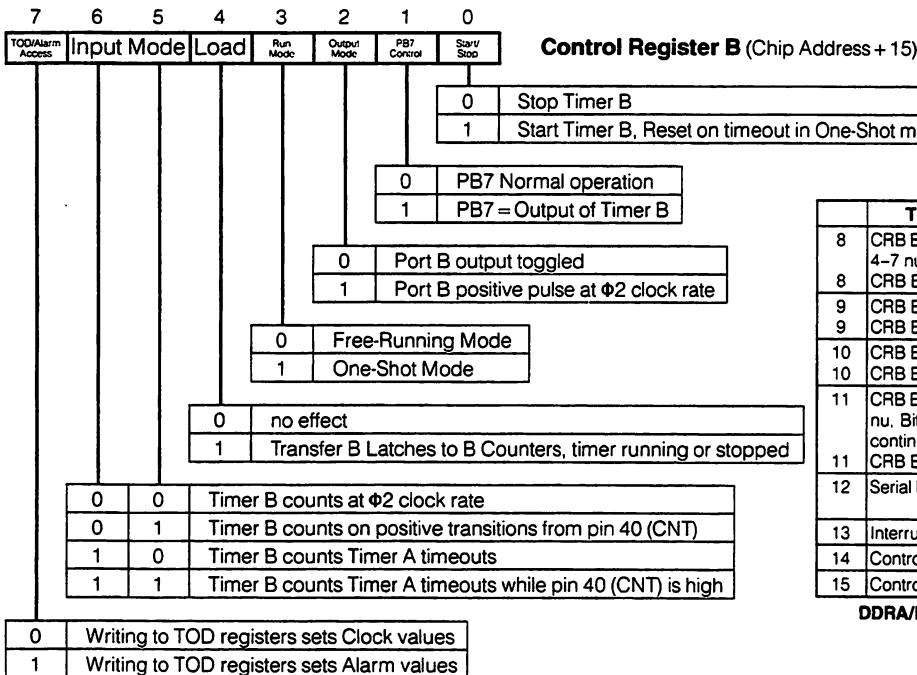
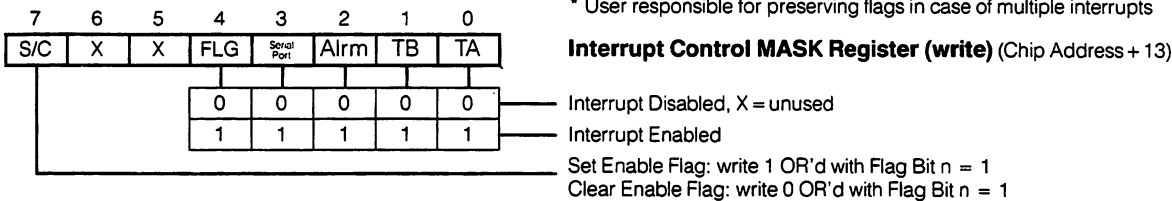
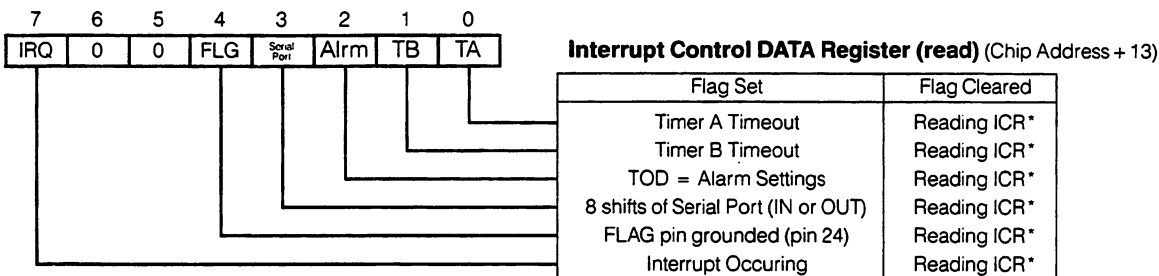
Set Enable Flag: write 1 OR'd with Flag Bit n = 1  
Clear Enable Flag: write 0 OR'd with Flag Bit n = 1

## Peripheral Control Register (Chip Address + 12)

7	6	5	4	3	2	1	0
CB2 Control		CB1 I/O Control	CA2 Control		CA1 I/O Control		
				0	Interrupt Flag Reg Bit1 = 1 on CA1 going low		
				1	Interrupt Flag Reg Bit1 = 1 on CA1 going high		
				Interrupt Flag Reg Bit1 cleared by reading I/O Port A			
0	0	0	Input Mode: IFR Bit0 = 1 on CA2 going low (IFR Bit0 cleared by read/write of I/O Port A)				
0	0	1	Independent Int. Input Mode: IFR Bit0 = 1 on CA2 going low (IFR Bit0 is not cleared by read/write of I/O Port A)				
0	1	0	Input Mode: IFR Bit0 = 1 on CA2 going high (IFR Bit0 cleared by read/write of I/O Port A)				
0	1	1	Independent Int. Input Mode: IFR Bit0 = 1 on CA2 going high (IFR Bit0 is not cleared by read/write of I/O Port A)				
1	0	0	Output Mode w/Handshaking: CA2 goes low on reading/writing I/O Port A (CA2 goes high on pulse from CA1)				
1	0	1	Pulse Output Mode: CA2 goes low for one $\Phi 2$ cycle on reading/writing I/O Port A				
1	1	0	Manual Output: CA2 set low				
1	1	1	Manual Output: CA2 set high				
				0	Interrupt Flag Reg Bit4 = 1 on CB1 going low		
				1	Interrupt Flag Reg Bit4 = 1 on CB1 going high		
				Interrupt Flag Reg Bit4 cleared by reading I/O Port B			
0	0	0	Interrupt Input Mode: IFR Bit3 = 1 on CB2 going low (IFR Bit3 cleared by reading/writing I/O Port B)				
0	0	1	Independent Int. Input Mode: IFR Bit3 = 1 on CB2 going low (IFR Bit3 is not cleared by reading/writing I/O Port B)				
0	1	0	Input Mode: IFR Bit3 = 1 on CB2 going high (IFR Bit3 cleared by reading/writing I/O Port A)				
0	1	1	Independent Int. Input Mode: IFR Bit3 = 1 on CB2 going high (IFR Bit3 is not cleared by reading/writing I/O Port A)				
1	0	0	Output Mode w/Handshaking: CB2 goes low on reading/writing I/O Port A (CB2 goes high on pulse from CB1)				
1	0	1	Pulse Output Mode: CB2 goes low for one $\Phi 2$ cycle on reading/writing I/O Port A				
1	1	0	Manual Output: CB2 set low				
1	1	1	Manual Output: CB2 set high				



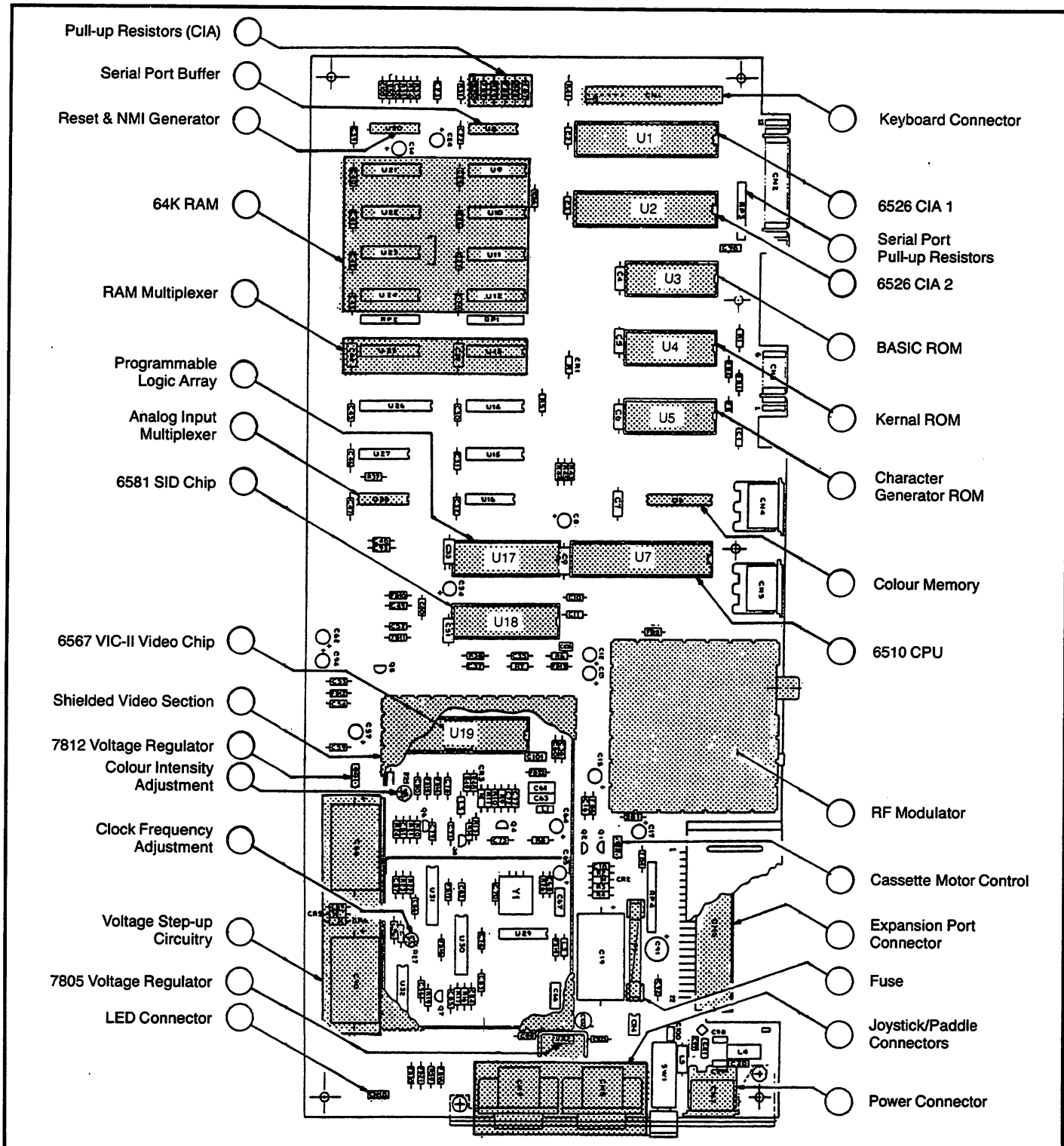
Reg#	Register Function
1	I/O Port A Data register
0	I/O Port B Data register
3	I/O Port A Data Direction
2	I/O Port B Data Direction
4	Read: Timer A Counter low. Resets TA Int. Flag (ICR Bit0) Write: Timer A Latch low. TA Latch low xferred to TA Counter low on writing Reg 5
5	Read: Timer A Counter high. Write: Timer A Latch high. Latch high transferred to TA on writing
6	Read: Timer B Counter low. Resets TB Int. Flag (ICR Bit1) Write: Timer B Latch low. TB Latch low xferred to TA Counter low on writing Reg 7
7	Read: Timer B Counter high. Write: Timer B Latch high. Latch high transferred to TB on writing



Time Of Day Clock, Read or Write		nu = not used
8	CRB Bit7 = 0: TOD 10ths. Bits 0-3 hold 10ths of seconds in BCD (bits 4-7 nu). Writing Reg 8 starts clock.	
8	CRB Bit7 = 1: Alarm 10ths, same format, write only.	
9	CRB Bit7 = 0: TOD Secs in BCD (Bits 0-3 + Bits 4-6 x 10, B7 nu)	
9	CRB Bit7 = 1: Alarm Seconds, same format, write only.	
10	CRB Bit7 = 0: TOD Mins in BCD (Bits 0-3 + Bits 4-6 x 10, B7 nu)	
10	CRB Bit7 = 1: Alarm Minutes, same format, write only.	
11	CRB Bit7 = 0: TOD Hours in BCD (Bits 0-3 + Bit 4 x 10, Bits 5 and 6 nu, Bit 7 = AM/PM) Reading Reg 11 latches TOD values, but clock continues. Reading Reg 8 (10ths) disables latch.	
11	CRB Bit7 = 1: Alarm Hours, same format, write only.	
12	Serial Data Reg. Shift OUT: Bit7 first out. Shift IN: Bit0 first in, shifted towards Bit7.	
13	Interrupt Control Register (ICR)	
14	Control Register A (CRA)	
15	Control Register B (CRB)	

DDRA/B: Bit = 0 Input, Bit = 1 Output (Remember: NOT I/O)

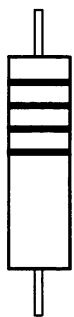
At least 3 circuit boards exist, but differences are minor in most cases.



Hardware: C64 Board Layout

The Complete Commodore Inner Space Anthology

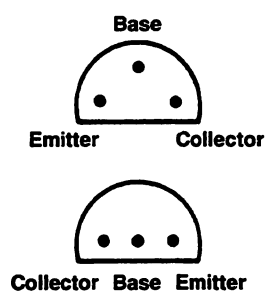
## Resistor Colour Codes



1st Band: 1st Digit  
2nd Band: 2nd Digit  
3rd Band: Multiplier (# of Zeros)  
4th Band: Tolerance

Colour	Value	"Remember:"	Fractional Multipliers	
Black	0	Bad	Colour	Multiply by:
Brown	1	Boys	Gold	0.10
Red	2	Rape	Silver	0.01
Orange	3	Our	<b>Tolerance Percents</b>	
Yellow	4	Young	No Band	±20%
Green	5	Girls	Silver	±10%
Blue	6	But	Gold	±5%
Violet	7	Violet		
Grey	8	Gives		
White	9	Willinlv		

## Transistor Leads



Collector Base Emitter

# ACIA / VIC 20 / Commodore 64 / B / + 4 RS 232 Control

Features not common to all machines are so noted.

OPEN LF, 2, SA, CHR\$( (7 6 5 4 3 2 1 0 ) ) + CHR\$( (7 6 5 4 3 2 1 0 ) )

SA	B Series:
1	Open Output Channel
2	Open Input Channel
3	Open Input/Output Channel
129	Output Channel, Convert CBM to ASCII
130	Input Channel, Convert ASCII to CBM
131	Input/Output, Convert ASCII=CBM

ACIA / VIC 20 / C64 / B / + 4 RS 232 Status							
7	6	5	4	3	2	1	0
ST: Status Variable = Status Register							
1 = Parity Error							
1 = Framing Error							
1 = Receiver Buffer Overrun							
ACIA: 1 = Receiver Register Full							
VIC/64: 0 = Receiver Buffer Empty							
ACIA: 1 = Transmitter Register Empty							
VIC/64: 1 = CTS Signal Missing							
1 = Carrier Detected							
1 = Data Set Not Ready							
1 = Interrupt Has Occurred							

### Notes

- The Command Register is optional for VIC/64 + 4
- If the LF# is 128 or greater, a Line Feed will be sent after each Carriage Return
- The Secondary Address SA does not affect RS 232 operation
- Before Closing the channel, check output buffer for data with:  
VIC/64 : 100 IF PEEK(669)<>PEEK(670) THEN 100

### ASCII Definitions

<b>ACK</b> Acknowledge	<b>FS</b> File Separator
<b>BS</b> Backspace	<b>FF</b> Form Feed
<b>BEL</b> Bell	<b>GS</b> Group Separator
<b>CAN</b> Cancel	<b>HT</b> Horizontal Tab
<b>CR</b> Carriage Return	<b>LF</b> Line Feed
<b>DLE</b> Data Link Escape	<b>NAK</b> Negative Ack
<b>DEL</b> Delete	<b>NUL</b> Null
<b>DC1</b> Device Control 1	<b>RS</b> Record Separator
<b>DC2</b> Device Control 2	<b>SI</b> Shift In
<b>DC3</b> Device Control 3	<b>SO</b> Shift Out
<b>DC4</b> Device Control 4	<b>SOH</b> Start Of Heading
<b>EM</b> End of Medium	<b>STX</b> Start of Text
<b>EOT</b> End Of Transmission	<b>SUB</b> Substitute
<b>ETB</b> End of Xmission block	<b>SYN</b> Synchronous Idle
<b>ETX</b> End of Text	<b>US</b> Unit Separator
<b>ENQ</b> Enquiry	<b>VT</b> Vertical Tab
<b>ESC</b> Escape	

### Pin Assignments For RS 232C Connector

Secondary Transmitted Data	14	•	•	1	Ground
Transmit Clock	15	•	•	2	Transmitted Data
Secondary Received Data	16	•	•	3	Received Data
Receiver Clock	17	•	•	4	Request To Send (RTS)
Unassigned	18	•	•	5	Clear To Send (CTS)
Secondary Request To send	19	•	•	6	Data Set Ready (DSR)
Data Terminal Ready (DTR)	20	•	•	7	Logic Ground
Signal Quality Detect	21	•	•	8	Carrier Detect
Ring Detect	22	•	•	9	Reserved
Data Rate Select	23	•	•	10	Reserved
Transmit Clock	24	•	•	11	Unassigned
Unassigned	25	•	•	12	Secondary Carrier Detect
				13	Secondary Clear To Send

### Control Register

Baud	
0 0 0 0	User*
0 0 0 1	50
0 0 1 0	75
0 0 1 1	110
0 1 0 0	134.5
0 1 0 1	150
0 1 1 0	300
0 1 1 1	600
1 0 0 0	1200
1 0 0 1	2400
1 0 1 0	2400
1 0 1 1	3600*
1 1 0 0	4800*
1 1 0 1	7200*
1 1 1 0	9600*
1 1 1 1	19200*

\* VIC/64: not implemented  
B/+4: User = 1/16 External

RCVR Clock	ACIA/B/+4
VIC/64	0 External
X	Not Used
	1 Internal

### Word Length

0 0	8 Bits
0 1	7 Bits
1 0	6 Bits
1 1	5 Bits

### Stop Bits

0 1	Stop Bit
1 1	2 Stop Bits

### Command Register

B Series:			
+ CHR\$(0) + CHR\$(0) not used but necessary			
B Series			
X	X	X	X
Not Used			
VIC/64 Handshake		ACIA/+4 Data Terminal Ready	
0	3 Line	Disable Rcv/Xmit (DTR high)	
1	X Line	Enable Rcv/Xmit (DTR low)	
VIC/64			
X	X	X	Not Used
ACIA and +4 Receiver Interrupt			
0	Enable IRQ from Status Reg Bit 0		
1	Disable IRQ Interrupt		
ACIA and +4 Transmitter Controls			
	Transmit Interrupt	RTS Level	Other
0 0	Disabled	High	—
0 1	Enabled	Low	—
1 0	Disabled	Low	—
1 1	Disabled	Low	Transmit BRK
Duplex			
0	Full		
1	Half		
Parity			
X	X	0	Disabled
0	0	1	Odd
0	1	1	Even
1	0	1	Mark
1	1	1	Space

### RS 232 User Port Lines



VIC 20 RS 232 is controlled by VIA 1 (6522) at \$9110  
C64 RS 232 is controlled by CIA 2 (6526) at \$DD00  
SuperPET RS 232 is controlled by ACIA (6551) at \$EFFF  
B Series RS 232 is controlled by ACIA (6551) at \$DD00  
+ 4 RS 232 is controlled by ACIA (6551) at \$FD00

Pin#	Chip	Description	Abv	Dir.	Modes
A	GND	Protective Ground	GND		1 2
B	FLAG2	Received Data	S <sub>n</sub>	IN	1 2
C	PB0	Received Data	S <sub>n</sub>	IN	1 2
D	PB1	Request to Send	RTS	OUT	1* 2
E	PB2	Data Terminal Ready	DTR	OUT	1* 2
F	PB3	Ring Indicator	RI	IN	3
H	PB4	Received line Signal	DCD	IN	2
J	PB5	Unassigned		IN	3
K	PB6	Clear To Send	CTS	IN	2
L	PB7	Data Set Ready	DSR	IN	2
M	PA2	Transmitted Data	S <sub>out</sub>	OUT	1 2
N	GND	Signal Ground	GND		1 2 3

#### Available Modes

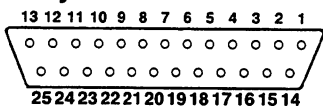
- 3-Line interface (S<sub>n</sub>, S<sub>out</sub>, GND)
  - X-Line interface.
  - User available only (unused in code)
- \* these lines are held high during 3-line mode.

## Cartridge Connector



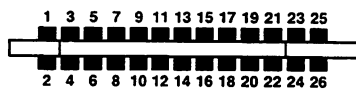
Pin	Name	Pin	Name
1	RO	A	BD0
2	A1	B	BD1
3	A2	C	BD2
4	A3	D	BD3
5	A4	E	BD4
6	A5	F	BD5
7	A6	H	BD6
8	A7	J	BD7
9	A8	K	GND
10	A9	L	GND
11	A10	M	SR/W
12	A11	N	SO2
13	A12	P	CSBANK 1
14	+5 VDC	R	CSBANK 2
15	+5 VDC	S	CSBANK 3

## Keyboard Connector



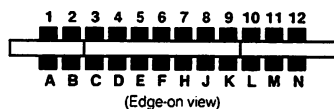
Pin	Name	Pin	Name
1	PA0	14	PA1
2	PA2	15	PA3
3	PA4	16	PA5
4	PA6	17	PA7
5	PB0	18	PC0
6	PB1	19	PC1
7	PB2	20	PC2
8	PB3	21	PC3
9	PB4	22	GND
10	PB5	23	GND
11	PB6	24	GND
12	PB7	25	PC4
13	PC5		

## User Connector



Pin	Name	Pin	Name
1	GND	2	PB2
3	GND	4	PB3
5	PC	6	FLAG
7	2D7	8	2D6
9	2D5	10	2D4
11	2D3	12	2D2
13	2D1	14	2D0
15	1D7	16	1D6
17	1D5	18	1D4
19	1D3	20	1D2
21	1D1	22	1D0
23	CNT	24	+5 VDC
25	IRQ	26	SP

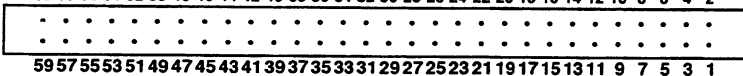
## IEEE Connector



Pin	Name	Pin	Name
1	D1	A	D5
2	D2	B	D6
3	D3	C	D7
4	D4	D	D8
5	EOI	E	REN
6	DAV	F	GND
7	NRFD	H	GND
8	NDAC	J	GND
9	IFC	K	GND
10	SRQ	L	GND
11	ATN	M	GND
12	SHIELD	N	GND

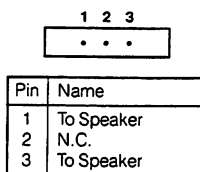
## B Series Connectors

60 58 56 54 52 50 48 46 44 42 40 38 36 34 32 30 28 26 24 22 20 18 16 14 12 10 8 6 4 2

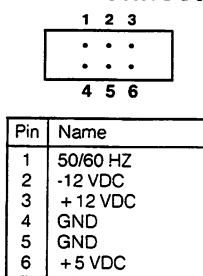


59 57 55 53 51 49 47 45 43 41 39 37 35 33 31 29 27 25 23 21 19 17 15 13 11 9 7 5 3 1

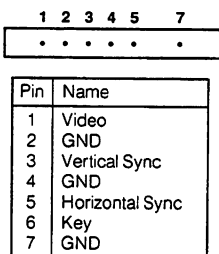
### Audio Jack



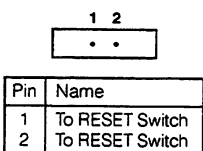
### Power Connector



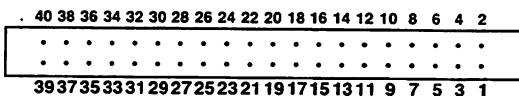
### Video Connector



### RESET Connector



### Co-Processor Connector

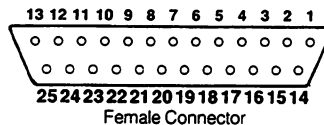


Pin	Name	Pin	Name
1	EXTMA	2	DRAM00
3	EXTMA2	4	DRAM01
5	EXTMA7	6	DRAM02
7	EXTMA6	8	DRAM03
9	EXTMA5	10	DRAM04
11	EXTMA4	12	DRAM05
13	EXTMA1	14	DRAM06
15	EXTMA0	16	DRAM07
17	GND	18	GND
19	GND	20	GND
21	GND	22	BUSY 1
23	GND	24	P2REFREQ
25	GND	26	P2REFGRNT
27	GND	28	BP0
29	GND	30	BP1
31	GND	32	BP2
33	N.C.	34	BP3
35	PROCRES	36	BUSY
37	EXTBUFRW	38	ERAS
39	DRAM R/W	40	ECAS

## Expansion Connector

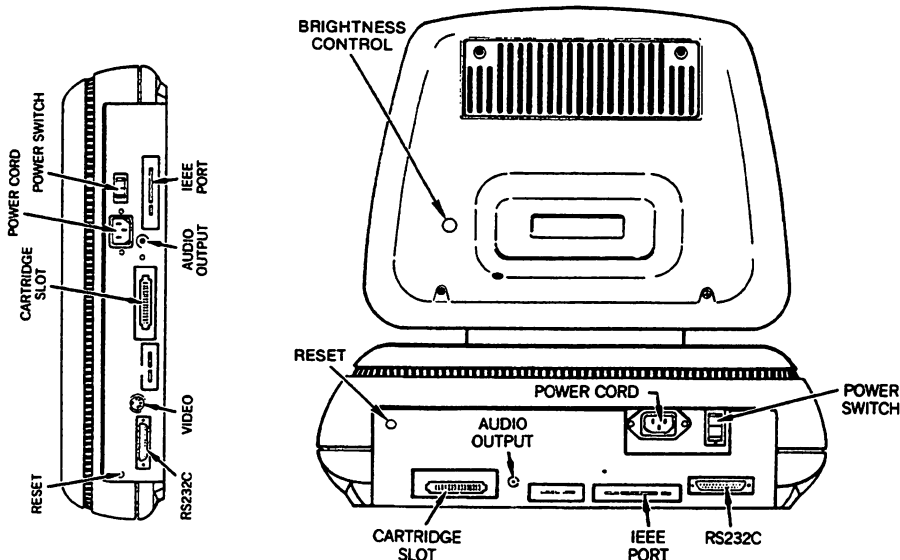
Pin	Name	Pin	Name
1	+5 VDC	2	+5 VDC
3	+5 VDC	4	+5 VDC
5	GND	6	GND
7	GND	8	GND
9	GND	10	GND
11	BRAS	12	IRQ3
13	-12 VDC	14	EXTRES
15	+12 VDC	16	S.O.
17	RES	18	LPEN
19	SR/W	20	EXTBUFCS
21	TODCLK	22	DISKROMCS
23	BOOTCLK	24	N.C.
25	S02	26	BCAS
27	S01	28	CST
29	BD3	30	EXTPRTCS
31	BD4	32	BD2
33	BD5	34	BD1
35	DB7	36	BD0
37	BA13	38	BD7
39	BA14	40	BA15
41	BA1	42	BA0
43	BA2	44	BA11
45	BA3	46	BA10
47	BA12	48	BA4
49	BA9	50	BA5
51	BA8	52	BA6
53	BP0	54	BA7
55	BP1	56	BP2
57	NMI	58	BP3
59	RDY	60	IRQ

### RS 232C Connector



Pin	Name
1	SHIELD
2	T x D
3	R x D
4	RTS
5	CTS
6	DSR
7	GND
8	DCD
11	+5 VDC
18	-12 VDC
20	DTR
24	R x C

(all others N.C.)



# Chip Pinouts

### 6502 CPU

V <sub>ss</sub>	1	40	Reset
RDY	2	39	$\Phi_2$ OUT
$\Phi_1$ OUT	3	38	S.O.
IRQ	4	37	$\Phi_0$ IN
N.C.	5	36	N.C.
NMI	6	35	N.C.
SYNC	7	34	R/W
V <sub>cc</sub>	8	33	DB0
AB0	9	32	DB1
AB1	10	31	DB2
AB2	11	30	DB3
AB3	12	29	DB4
AB4	13	28	DB5
AB5	14	27	DB6
AB6	15	26	DB7
AB7	16	25	AB15
AB8	17	24	AB14
AB9	18	23	AB13
AB10	19	22	AB12
AB11	20	21	V <sub>ss</sub>

### 6509 CPU

Ready	1	40	$\Phi_0$ IN
IRQ	2	39	Reset
SYNC	3	38	$\Phi_0$ OUT
NMI	4	37	R/W
AEC	5	36	D0
V <sub>oo</sub>	6	35	D1
A0	7	34	D2
A1	8	33	D3
A2	9	32	D4
A3	10	31	D5
A4	11	30	D6
A5	12	29	D7
A6	13	28	S.O.
A7	14	27	P0
A8	15	26	P1
A9	16	25	P2
A10	17	24	P3
A11	18	23	A15
A12	19	22	A14
A13	20	21	V <sub>ss</sub>

### 6510 CPU

Clk 0 IN	1	40	Reset
Ready	2	39	$\Phi_2$
IRQ	3	38	R/W
NMI	4	37	D0
AEC	5	36	D1
V <sub>cc</sub>	6	35	D2
A0	7	34	D3
A1	8	33	D4
A2	9	32	D5
A3	10	31	D6
A4	11	30	D7
A5	12	29	P0
A6	13	28	P1
A7	14	27	P2
A8	15	26	P3
A9	16	25	P4
A10	17	24	P5
A11	18	23	A15
A12	19	22	A14
A13	20	21	GND

### Z-80 CPU

A11	1	40	A10
A12	2	39	A9
A13	3	38	A8
A14	4	37	A7
A15	5	36	A6
$\Phi$	6	35	A5
D4	7	34	A4
D3	8	33	A3
D5	9	32	A2
D6	10	31	A1
+5V	11	30	A0
D2	12	29	GND
D7	13	28	RFSH
D0	14	27	MT
D1	15	26	Reset
INT	16	25	BUS RQ
NMI	17	24	WAIT
HALT	18	23	BUSAK
MREQ	19	22	WR
IORQ	20	21	RD

### 6520 PIA

(Peripheral Interface Adapter)

V <sub>ss</sub>	1	40	CA1
PA0	2	39	CA2
PA1	3	38	IRQA
PA2	4	37	IRQB
PA3	5	36	RS0
PA4	6	35	RS1
PA5	7	34	Reset
PA6	8	33	D0
PA7	9	32	D1
PB0	10	31	D2
PB1	11	30	D3
PB2	12	29	D4
PB3	13	28	D5
PB4	14	27	D6
PB5	15	26	D7
PB6	16	25	$\Phi_2$
PB7	17	24	CS1
CB1	18	23	CS2
CB2	19	22	CS0
V <sub>cc</sub>	20	21	R/W

### 6522 VIA

(Versatile Interface Adapter)

V <sub>ss</sub>	1	40	CA1
PA0	2	39	CA2
PA1	3	38	RS0
PA2	4	37	RS1
PA3	5	36	RS2
PA4	6	35	RS3
PA5	7	34	Reset
PA6	8	33	D0
PA7	9	32	D1
PB0	10	31	D2
PB1	11	30	D3
PB2	12	29	D4
PB3	13	28	D5
PB4	14	27	D6
PB5	15	26	D7
PB6	16	25	$\Phi_2$
PB7	17	24	CS1
CB1	18	23	CS2
CB2	19	22	R/W
V <sub>cc</sub>	20	21	IRQ

### 6526 CIA

(Complex Interface Adapter)

V <sub>ss</sub>	1	40	CNT
PA0	2	39	SP
PA1	3	38	RS0
PA2	4	37	RS1
PA3	5	36	RS2
PA4	6	35	RS3
PA5	7	34	Reset
PA6	8	33	DB0
PA7	9	32	DB1
PB0	10	31	DB2
PB1	11	30	DB3
PB2	12	29	DB4
PB3	13	28	DB5
PB4	14	27	DB6
PB5	15	26	DB7
PB6	16	25	$\Phi_2$
PB7	17	24	FLAG
PC	18	23	CS
TOD	19	22	R/W
V <sub>cc</sub>	20	21	IRQ

### 6525 TPI

(Tri-Port Interface)

V <sub>ss</sub>	1	40	DB7
PA0	2	39	DB6
PA1	3	38	DB5
PA2	4	37	DB4
PA3	5	36	DB3
PA4	6	35	DB2
PA5	7	34	DB1
PA6	8	33	DB0
PA7	9	32	PC7
PB0	10	31	PC6
PB1	11	30	PC5
PB2	12	29	PC4
PB3	13	28	PC3
PB4	14	27	PC2
PB5	15	26	PC1
PB6	16	25	PC0
PB7	17	24	RS0
CS	18	23	RS1
R/W	19	22	RS2
V <sub>oo</sub>	20	21	Reset

### 6529 SPI

(Single Port Interface)

R/W	1	20	V <sub>oo</sub>
P0	2	19	CS
P1	3	18	DB0
P2	4	17	DB1
P3	5	16	DB2
P4	6	15	DB3
P5	7	14	DB4
P6	8	13	DB5
P7	9	12	DB6
V <sub>ss</sub>	10	11	DB7

### 6581 - SID CHIP

(Sound Interface Device)

CAP1A	1	28	V <sub>oo</sub>
CAP1B	2	27	Audio OUT
CAP2A	3	26	EXT IN
CAP2B	4	25	V <sub>cc</sub>
Reset	5	24	POT X
$\Phi_2$	6	23	POT Y
R/W	7	22	D7
CS	8	21	D6
A0	9	20	D5
A1	10	19	D4
A2	11	18	D3
A3	12	17	D2
A4	13	16	D1
GND	14	15	D0

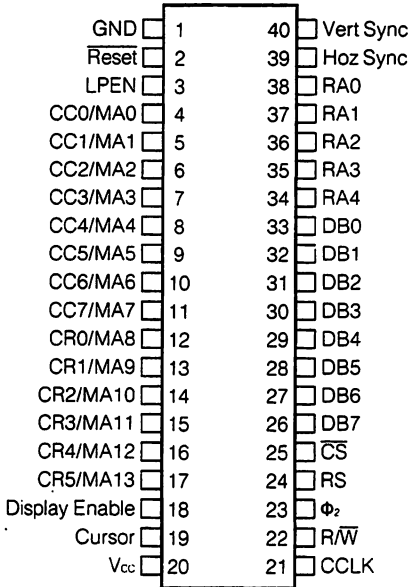
### 6551 - ACIA

(Async Communications Interface Adapter)

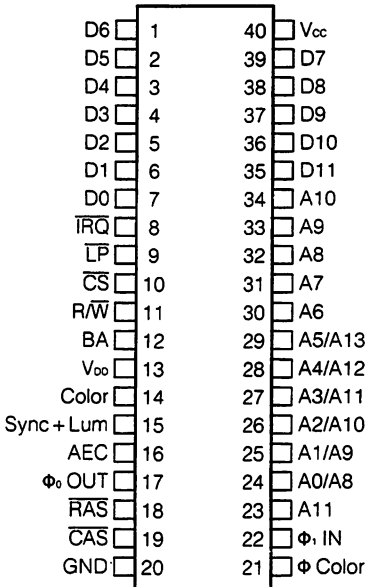
GND	1	28	R/W
CS0	2	27	$\Phi_2$
CS1	3	26	IRQ
Reset	4	25	DB7
RxC	5	24	DB6
XTAL1	6	23	DB5
XTAL2	7	22	DB4
RTS	8	21	DB3
CTS	9	20	DB2
TxD	10	19	DB1
DTR	11	18	DB0
RxD	12	17	DSR
RS0	13	16	D $\overline{C}$ D
RS1	14	15	V <sub>cc</sub>



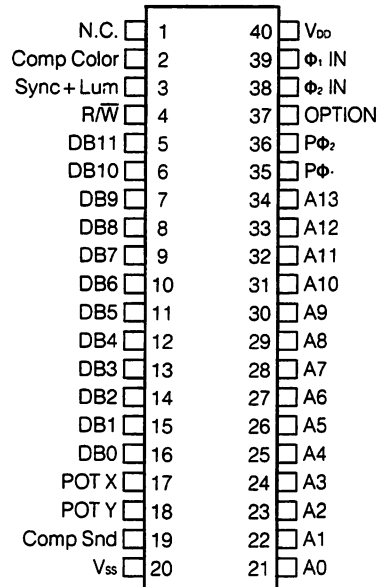
**6545-1 CRT Controller**



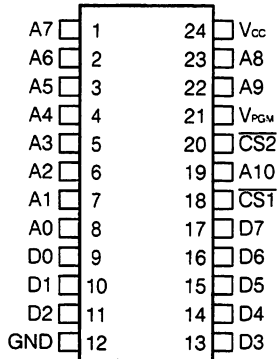
**6567 VIC CHIP**  
(Video Interface Chip)



**650/61 VIC II CHIP**  
(Video Interface Chip)

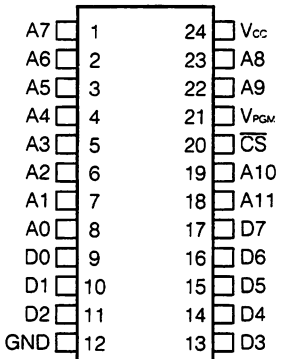


**2516 EPROM**  
2K x 8 Bits



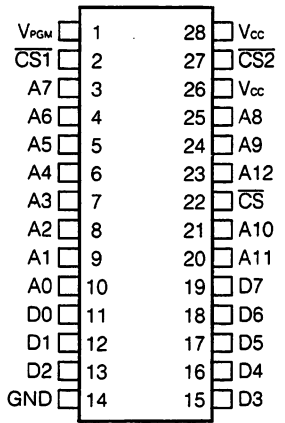
Low power operation when CS lines high.  
 V<sub>PGM</sub>: Apply +25 volts to program chip memories.

**2532 EPROM**  
4K x 8 Bits



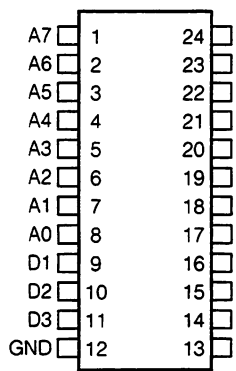
Low power operation when CS lines high.  
 V<sub>PGM</sub>: Apply +25 volts to program chip memories.

**2564 EPROM**  
8K x 8 Bits

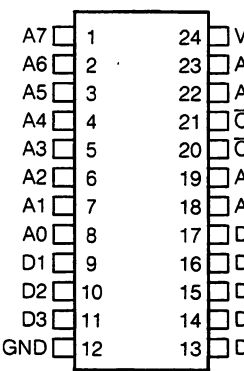


Low power operation when CS lines high.  
 V<sub>PGM</sub>: Apply +25 volts to program chip memories.

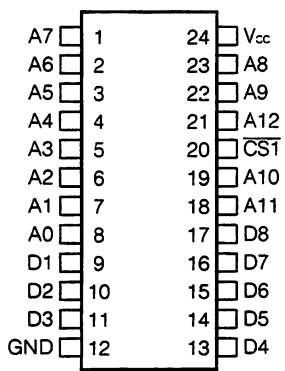
**2316 2K Static ROM**  
2K x 8 Bits



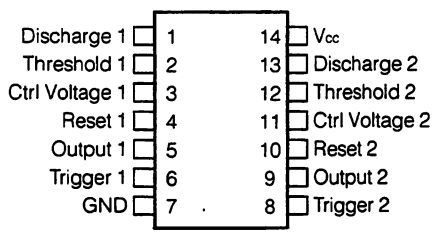
**2332 4K Static ROM**  
4K x 8 Bits



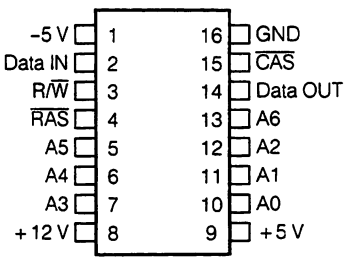
**2364 8K Static ROM**  
8K x 8 Bits



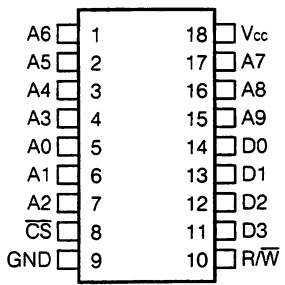
**556 Dual Timer**



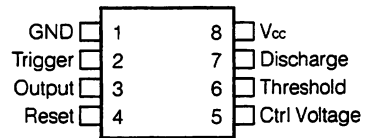
**4116 16K Dynamic RAM**



**2114 Static RAM**  
1K x 4 Bits



**555 Timer**



# Checking Semiconductors with an Ohmmeter

<b>P-N Diodes (including Zener, Photodiodes, or any simple P-N junction)</b>		
Ohmmeter + lead to	Ohmmeter -lead to	Operational Results
Anode (forward bias)	Cathode	short or low resistance (10-1000 ohms depending on diode type)
Cathode (reverse bias)	Anode	open or high resistance (Germanium: 1M ohm typical. Silicon: 10M ohm or greater)
<b>Tunnel Diodes</b>		
Ohmmeter + lead to	Ohmmeter -lead to	Operational Results
Anode (forward bias)	Cathode	short or low resistance
Cathode (reverse bias)	Anode	same, slightly lower with Cathode on +
<b>Photoconductive Cells</b>		
Ohmmeter + lead to	Ohmmeter -lead to	Operational Results
Either end	Either end	Ohmmeter reading should be equal in either direction. Resistance should increase as light decreases.
<b>Photodiodes, LEDs, Photovoltaic Cells (LED: Short Lead = Cathode)</b>		
Ohmmeter + lead to	Ohmmeter -lead to	Operational Results
Anode (forward bias)	Cathode	short or low resistance (10-1000 ohms depending on diode type)
Cathode (reverse bias)	Anode	open or high resistance (Germanium: 1Mohm typical. Silicon: 10M ohm or greater)
<b>NPN Transistors</b>		
Ohmmeter + lead to	Ohmmeter -lead to	Operational Results
Emitter	Base	High resistance, unless ohmmeter voltage exceed breakdown voltage
Base	Emitter	Low resistance (forward biased junction)
Collector	Base	High resistance
Base	Collector	Low resistance, usually not as low as Emitter-Base junction since Collector is lightly doped
Emitter	Collector	High resistance, about 10-50 times less than Emitter-Base reverse bias resistance
Collector	Emitter	High resistance, slightly higher with Collector on +
<b>PNP Transistors</b>		
Ohmmeter + lead to	Ohmmeter -lead to	Operational Results
Emitter	Base	Low resistance (forward biased junction)
Base	Emitter	High resistance, unless ohmmeter voltage exceed breakdown voltage
Collector	Base	Low resistance, usually not as low as Emitter-Base junction since Collector is lightly doped
Base	Collector	High resistance
Emitter	Collector	High resistance, slightly higher with Emitter on +
Collector	Emitter	High resistance, about 10-50 times less than Base-Emitter resistance
<b>Four-Layer Diodes, Silicon Unilateral Switches (SUS)</b>		
Ohmmeter + lead to	Ohmmeter -lead to	Operational Results
Anode (forward bias)	Cathode	High resistance (1Mohm or greater)
Cathode (reverse bias)	Anode	High resistance, greater than Anode-Cathode, but immeasurable without accurate meter
<b>DIAC, SBS</b>		
Ohmmeter + lead to	Ohmmeter -lead to	Operational Results
Either end	Either end	High resistance, 1M ohm or greater
<b>SCR (including light-activated SCR), GCS (gate-controlled switch)</b>		
Ohmmeter + lead to	Ohmmeter -lead to	Operational Results
Anode (forward bias)	Cathode	High resistance, 1M ohm or greater, slightly less for hi-current SCRs
Cathode (reverse bias)	Anode	High resistance, 1M ohm or greater, usually higher than Anode-Cathode direction
Gate	Cathode	High resistance (same as P-N Diode)
Cathode	Gate	Low resistance (same as P-N Diode)
Gate	Anode	High resistance, 1M ohm or greater
Anode	Gate	High resistance, 1M ohm or greater
<b>TRIAC</b>		
Ohmmeter + lead to	Ohmmeter -lead to	Operational Results
Either Anode 1 or 2	Either Anode 2 or 1	High resistance, 1M ohm or greater, slightly less for hi-current SCRs
Gate	Anode 1	Low resistance
Anode 1	Gate	Low resistance
Gate	Anode 2	High resistance
Anode 2	Gate	High resistance

<b>UJT (Unijunction Transistor)</b>		
Ohmmeter + lead to	Ohmmeter -lead to	Operational Results
Base 1	Base 2	Typically 4K-10K ohms
Base 2	Base 1	Same, 4K-10K ohms
Emitter (forward bias)	Base 1	Typically 3K-15K ohms
Base 1	Emitter	High resistance, 1M ohm or greater
Emitter (forward bias)	Base 2	Typically 2K-10K ohms, usually less than Emitter-Base 1
Base 2	Emitter	High resistance, 1M ohm or greater
<b>Complementary UJT</b>		
Ohmmeter + lead to	Ohmmeter -lead to	Operational Results
Base 1	Base 2	Typically 4K-10K ohms
Base 2	Base 1	Same, 4K-10K ohms
Base 1	Emitter (forward bias)	Typically 3K-15K ohms
Emitter	Base 1	High resistance, 1M ohm or greater
Base 2	Emitter (forward bias)	Typically 2K-10K ohms, usually less than Base 1-Emitter
Emitter	Base 2	High resistance, 1M ohm or greater
<b>Programmable UJT (PUT)</b>		
Ohmmeter + lead to	Ohmmeter -lead to	Operational Results
Anode	Cathode	High resistance, 1M ohm or greater
Cathode	Anode	High resistance, 1M ohm or greater
Anode	Gate	Low resistance (forward bias)
Gate	Anode	High resistance
Gate	Cathode	High resistance
Cathode	Gate	High resistance
<b>N-Channel JFET (Field Effect Transistor)</b>		
Ohmmeter + lead to	Ohmmeter -lead to	Operational Results
Drain	Source	Typically 500-5K ohms
Source	Drain	Same, 500-5K ohms
Gate	Drain	Low resistance (forward biased P-N junction)
Gate	Source	Low resistance (forward biased P-N junction)
Drain	Gate	High resistance, 10M ohm or greater, unless Ohmmeter voltage exceeds JFET breakdown voltage
Source	Gate	High resistance, 10M ohm or greater, unless Ohmmeter voltage exceeds JFET breakdown voltage
<b>P-Channel JFET</b>		
Ohmmeter + lead to	Ohmmeter -lead to	Operational Results
Source	Drain	Typically 500-5K ohms
Drain	Source	Same, 500-5K ohms
Drain	Gate	Low resistance (forward biased P-N junction)
Source	Gate	Low resistance (forward biased P-N junction)
Gate	Drain	High resistance, 10M ohm or greater, unless Ohmmeter voltage exceeds JFET breakdown voltage
Gate	Source	High resistance, 10M ohm or greater, unless Ohmmeter voltage exceeds JFET breakdown voltage
<b>Enhancement MOSFET (Metal Oxide Semiconductor FET)</b>		
Ohmmeter + lead to	Ohmmeter -lead to	Operational Results
Drain	Source	High resistance, 10M ohm or greater
Source	Drain	High resistance, 10M ohm or greater
Gate	Drain	High resistance, 100M ohm or greater, either direction
Gate	Source	High resistance, 100M ohm or greater, either direction
<b>Depletion MOSFET</b>		
Ohmmeter + lead to	Ohmmeter -lead to	Operational Results
Drain	Source	Typically 500-5K ohms
Source	Drain	Same, 500-5K ohms
Gate	Drain	High resistance, 100M ohm or greater, either direction
Gate	Source	High resistance, 100M ohm or greater, either direction

Inch Fractions				
in Decimal & Millimeters				
Inches		Decimal	Millimeters	
1/64		0.0156	0.397	
2/64	1/32	0.0313	0.794	
3/64		0.0469	1.191	
4/64	1/16	0.0625	1.588	
5/64		0.0781	1.985	
6/64	3/32	0.0938	2.381	
7/64		0.1094	2.778	
8/64	1/8	0.1250	3.175	
9/64		0.1406	3.572	
10/64	5/32	0.1563	3.969	
11/64		0.1719	4.366	
12/64	3/16	0.1875	4.762	
13/64		0.2031	5.159	
14/64	7/32	0.2188	5.556	
15/64		0.2344	5.953	
16/64	1/4	0.2500	6.350	
17/64		0.2656	6.747	
18/64	9/32	0.2813	7.144	
19/64		0.2969	7.541	
20/64	5/16	0.3125	7.937	
21/64		0.3281	8.344	
22/64	11/32	0.3438	8.731	
23/64		0.3594	9.128	
24/64	3/8	0.3750	9.525	
25/64		0.3906	9.922	
26/64	13/32	0.4063	10.319	
27/64		0.4219	10.716	
28/64	7/16	0.4375	11.112	
29/64		0.4531	11.509	
30/64	15/32	0.4688	11.906	
31/64		0.4844	12.303	
32/64	1/2	0.5000	12.700	
33/64		0.5156	13.097	
34/64	17/32	0.5313	13.494	
35/64		0.5469	13.891	
36/64	9/16	0.5625	14.287	
37/64		0.5781	14.684	
38/64	19/32	0.5938	15.081	
39/64		0.6094	15.478	
40/64	5/8	0.6250	15.875	
41/64		0.6406	16.272	
42/64	21/32	0.6563	16.669	
43/64		0.6719	17.067	
44/64	1 1/16	0.6875	17.463	
45/64		0.7031	17.860	
46/64	23/32	0.7188	18.238	
47/64		0.7344	18.635	
48/64	3/4	0.7500	19.049	
49/64		0.7656	19.446	
50/64	25/32	0.7813	19.842	
51/64		0.7969	20.239	
52/64	13/16	0.8125	20.636	
53/64		0.8281	21.033	
54/64	27/32	0.8438	21.430	
55/64		0.8694	21.827	
56/64	7/8	0.8750	22.224	
57/64		0.8906	22.621	
58/64	29/32	0.9063	23.018	
59/64		0.9219	23.415	
60/64	15/16	0.9375	23.812	
61/64		0.9531	24.209	
62/64	31/32	0.9688	24.606	
63/64		0.9844	25.004	
64/64	1.0	1.0000	25.400	

International System of Units (SI)					
Units Prefixes					
Prefix	Symbol	Multiplier	Prefix	Symbol	Multiplier
Exa	E	10 <sup>18</sup>	Deci	d	10 <sup>-1</sup>
Peta	P	10 <sup>15</sup>	Centi	c	10 <sup>-2</sup>
Tera	T	10 <sup>12</sup>	Milli	m	10 <sup>-3</sup>
Giga	G	10 <sup>9</sup>	Micro	μ	10 <sup>-6</sup>
Mega	M	10 <sup>6</sup>	Nano	n	10 <sup>-9</sup>
Kilo	k	10 <sup>3</sup>	Pico	p	10 <sup>-12</sup>
Hecto	h	10 <sup>2</sup>	Femto	f	10 <sup>-15</sup>
Deca	da	10 <sup>1</sup>	Atto	a	10 <sup>-18</sup>

SI Base Units		
Quantity	SI Unit	Symbol
Length	Meters	m
Mass	Kilograms	kg
Time	Seconds	s
Electric Current	Amperes	A
Temperature	Degrees Kelvin	K
Amount of Substance	Moles	mol
Luminous Intensity	Candela	cd

SI Supplementary Units		
Quantity	SI Unit	Symbol
Plane Angle	Radians	rad
Solid Angle	Steradians	sr

SI Units Without Special Names		
Quantity	SI Unit	Symbol
Area	Square Meters	m <sup>2</sup>
Volume	Cubic Meters	m <sup>3</sup>
Linear Velocity (Speed)	Meters/Second	m/s
Angular Velocity	Radians/Second	rad/s
Linear Acceleration	Meters/Second Squared	m/s <sup>2</sup>
Angular Acceleration	Radians/Second Squares	rad/s <sup>2</sup>
Wavelength	Meters	m
Density	Kilogram/Cubic Meter	kg/m <sup>3</sup>
Concentration	Moles/Cubic Meter	mol/m <sup>3</sup>
Specific Volume	Cubic Meters/Kilogram	m <sup>3</sup> /kg
Luminance	Candela/Square Meter	cd/m <sup>2</sup>
Dynamic Viscosity	Pascal Seconds	Pa · s
Kinematic Viscosity	Square Meters/Second	m <sup>2</sup> /s
Moment of Force	Newton Meters	N × m
Surface Tension	Newton/Meter	N/m
Irradiance (Heat Flux Density)	Watts/Square Meter	W/m <sup>2</sup>
Entropy (Heat Capacity)	Joules/Kelvin	J/K
Specific Entropy	Joules/Kilogram-Kelvin	J/(kg × K)
Specific Energy	Joules/Kilogram	J/kg
Thermal Conductivity	Watts/Meter-Kelvin	W/(m × K)
Energy Density	Joules/Cubic Meter	J/m <sup>3</sup>
Electric Field Strength	Volts/Meter	V/m
Electric Charge Density	Coulombs/Cubic Meter	C/m <sup>3</sup>
Surface Density of Charge (Flux Density)	Coulombs/Square Meter	C/m <sup>2</sup>
Permittivity	Farads/Meter	F/m
Current Density	Amperes/Square Meter	A/m <sup>2</sup>
Magnetic Field Strength	Amperes/Meter	A/m
Permeability	Henries/Meter	H/m
Molar Energy	Joules/Mole	J/mol
Molar Entropy	Joules/Mole Kelvin	J/(mol × K)
Radiant Intensity	Watts/Steradian	W/sr
Radiance	Watts/Square Meter Steradian	W/(m <sup>2</sup> × sr)
Exposure	Coulombs/Kilogram	C/kg
Absorbed Dose Rate	Grays/Second	Gy/s

SI Units With Special Names			
Quantity	SI Unit	Symbol	Derivative
Frequency	Hertz	Hz	1/s or s <sup>-1</sup>
Force	Newtons	N	m × kg/s <sup>2</sup>
Pressure, Stress	Pascals	Pa	N/m <sup>2</sup>
Energy, Work, Quantity of Heat	Joules	J	N × m
Quantity of Heat	Calories	cal	
Power, Radiant Flux	Watt	W	J/s
Quantity of Electricity, Electric Charge	Coulombs	C	s × A
Electric Potential, Potential Difference			
Electromotive Force	Volts	V	W/A
Electric Capacitance	Farads	F	C/V
Electric Resistance	Ohms	Ω	V/A
Electric Conductance	Siemens	S	A/V
Magnetic Flux	Webers	Wb	V × s
Magnetic Flux Density	Tesla	T	Wb/m <sup>2</sup>
Inductance	Henries	H	Wb/A
Luminous Flux	Lumens	lm	cd × sr
Illuminance	Lux	lx	lm/m <sup>2</sup>
Activity of Radionuclides	Becquerels	Bq	s <sup>-1</sup>
Absorbed Dose of Ionising Radiation	Grays	Gy	J/kg

### Names For Large Numbers

Name	French & U.S. Equivalent	Number of Zeros	British & German Equivalent	Number of Zeros
million	1000 thousands	6	1000 thousands	6
million	1000 thousands	6	1000 thousands	6
milliard	1000 millions	9	1000 millions	9
billion	1000 millions	9	1,000,000 millions	12
trillion	1000 billions or 1,000,000 millions	12	1,000,000 billions or 1,000,000 million millions	18
quadrillion	1000 trillions	15	1,000,000 trillions	24
quintillion	1000 quadrillions	18	1,000,000 quadrillions	30
sextillion	1000 quintillions	21	1,000,000 quintillions	36
septillion	1000 sextillions	24	1,000,000 sextillions	42
octillion	1000 septillions	27	1,000,000 septillions	48
nonillion	1000 octillions	30	1,000,000 octillions	54
decillion	1000 nonillions	33	1,000,000 nonillions	60
undecillion	1000 decillions	36	1,000,000 decillions	66
duodecillion	1000 undecillions	39	1,000,000 undecillions	72
tredecillion	1000 duodecillions	42	1,000,000 duodecillions	78
quattuordecillion	1000 tredecillions	45	1,000,000 tredecillions	84
quintdecillion	1000 quattuordecillions	48	1,000,000 quattuordecillions	90
sexdecillion	1000 quintdecillions	51	1,000,000 quintdecillions	96
septdecillion	1000 sexdecillions	54	1,000,000 septdecillions	102
octodecillion	1000 septdecillions	57	1,000,000 octodecillions	108
novemdecillion	1000 octodecillions	60	1,000,000 novemdecillions	114
vigintillion	1000 novemdecillions	63	1,000,000 novemdecillions	120

### Roman Numerals

I	1	XI	11	XXX	30	CD	400
II	2	XII	12	XL	40	D	500
III	3	XIII	13	L	50	DC	600
IV	4	XIV	14	LX	60	DCC	700
V	5	XV	15	LXX	70	DCCC	800
VI	6	XVI	16	LXXX	80	CM	900
VII	7	XVII	17	XC	90	M	1000
VIII	8	XVIII	18	C	100	MCM	1900
IX	9	XIX	19	CC	200	MM	2000
X	10	XX	20	CCC	300	V	5000

**Rules:**  
 1. An overhead line indicates the value multiplied by 1000.  
 2. Repeating a letter repeats its value (XX = 20, CCC = 300)

### Boolean Truth Table

AND	OR	NOT	XOR
1 AND 1 = 1	1 OR 1 = 1	NOT 0 = 1	1 XOR 1 = 0
1 AND 0 = 0	1 OR 0 = 1	NOT 1 = 0	1 XOR 0 = 1
0 AND 1 = 0	0 OR 1 = 1		0 XOR 1 = 1
0 AND 0 = 0	0 OR 0 = 0		0 XOR 0 = 0
Result is 1 if both bits are 1	Result is 1 if either bit is 1	Each bit is complemented	Result is 1 if one or the other but not both

### Constant Values

Constant	Symbol	Value
Absolute Zero.		-273.15°C or -459.7°F
Ampere's Circuital Law Constant	K	$2 \times 10^7$ Newtons/Amp <sup>2</sup>
Avogadro's Number	$N_0$	$6.022169 \times 10^{23}$
Bohr Magneton	$\mu_B$	$9.274096 \times 10^{-24}$ Joules/Second
Boltzmann's Constant	k	$1.380622 \times 10^{-23}$ Joules/Degrees Kelvin
Coulomb's Law Constant	k	$8.988 \times m10^9$ Newton Meters Squared/Coulomb <sup>2</sup>
Electron Charge	e	$1.6021917 \times 10^{-19}$ C
Electron Charge To Mass Ratio	e/ $m_e$	$1.7588028 \times 10^{11}$ C/Kilogram
Faraday Constant	F	$9.648670 \times 10^7$ C k mole <sup>-1</sup>
Gas Constant	$R_0$	$8.31434 \times 10^3$ J-k mole <sup>-1</sup> K <sup>-1</sup>
Gravitational Constant	G	$6.6732 \times 10^{-11}$ Cubic Meters/Kilogram Seconds <sup>2</sup>
Planck's Constant	h	$6.626196 \times 10^{-34}$ Joule-Seconds
Rydberg Constant	$R_{\infty}$	$1.09737312 \times 10^7$ m <sup>-1</sup>
Speed of Light	C	$2.9979250 \times 10^8$ Meters/Second
Speed of Sound (in air at 28° C)		746 Miles/Hour
Speed of Sound (in air at 28° C)		348 Meters/Second
Earth Orbiting Satellite		7.5 Kilometers/Second (approx.)
Earth Orbiting Satellite		17000 Miles/Hour (approx.)
Compton Electron Wavelength	$\lambda_c$	$2.4263096 \times 10^{-12}$ Meters
Compton Proton Wavelength	$\lambda_{c,p}$	$1.3214409 \times 10^{-15}$ Meters
Compton Neutron Wavelength	$\lambda_{c,n}$	$1.3196217 \times 10^{-15}$ Meters
Electron Magnetic Moment	$\mu_e$	$9.284851 \times 10^{-24}$ Joules/Second
Proton Magnetic Moment	$\mu_p$	$1.4106203 \times 10^{-26}$ Joules/Second
Electron Rest Mass	$m_e$	$9.109558 \times 10^{-31}$ Kilograms
Proton Rest Mass	$m_p$	$1.672614 \times 10^{-27}$ Kilograms
Neutron Rest Mass	$M_n$	$1.00727661$ Atomic Mass Units
	$M_p$	$1.674920 \times 10^{-27}$ Kilograms
	$M_n$	$1.00866520$ Atomic Mass Units

### Force Formulae

**Force = Mass × Acceleration**  
**Horsepower**  
 1 HP = 33000 Foot-Pounds of Work per Minute

**Torque**  
 Torque = Force × Radius  
 Torque = 63025 × Horsepower / RPM

**Centrifugal Force**  
 Centrifugal Force (outward) = Centripetal Force (inward)  
 Centrifugal Force = Weight × Linear Velocity<sup>2</sup> / (32.16 × Radius)  
 Centrifugal Force = Weight × Radius × RPM<sup>2</sup> / 2932.55  
 Centrifugal Force = 1.22760 × Weight × Radius × RPS<sup>2</sup>  
 Weight is in pounds                          RPM is in revolutions/minute  
 Linear Velocity is in feet/second                  RPS is in revolutions/second  
 Radius is in feet

**Propeller Thrust**  
**Typical Thrust for a power boat:**  
 Prop Thrust = 33000 × Motor Horsepower × Prop Efficiency / Speed  
 Prop Thrust = 33000 × Motor HP × Prop Effic / (Prop Pitch × RPMs)  
 Where Prop Efficiency in water ranges from 60% to 70% (65% practically)  
 Speed is in feet/minute  
 Prop Pitch is in feet  
 RPMs is RPMs @ n Motor Horsepower

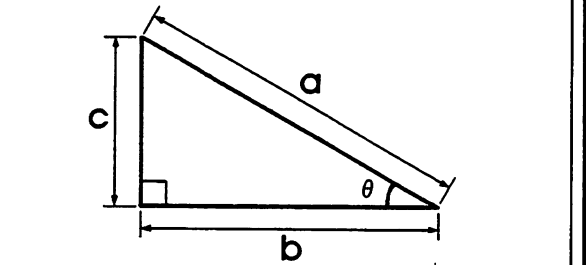
**Typical Thrust for an airplane in level flight:**  
 Prop Thrust = 375 × Motor Horsepower × Prop Efficiency / MPH  
 Where Prop Efficiency in air ranges from 70% to 87% (80% practically)

**Gravity**  
 X = Forward Velocity × Time  
 Y = Upward Velocity × Time - 1/2 Gravity × Time<sup>2</sup>  
 Where Gravity on Earth at Sea Level is 32.2 Feet/Second<sup>2</sup>

### Mathematical Functions

Function	BASIC Equivalent
Secant	SEC(X) = 1 / COS(X)
Cosecant	CSC(X) = 1 / SIN(X)
Cotangent	COT(X) = 1 / TAN(X)
Inverse Sine	ARCSIN(X) = ATN( X / SQR(-X*X + 1) )
Inverse Cosine	ARCCOS(X) = ATN( X / SQR(-X*X + 1) ) + pi/2
Inverse Secant	ARCSEC(X) = ATN( X / SQR(X*X - 1) )
Inverse Cosecant	ARCCSC(X) = ATN( X / SQR(X*X - 1) ) + (SGN(X) - 1)*pi/2
Inverse Cotangent	ARCCOT(X) = ATN(X) + pi/2
Hyperbolic Sine	SINH(X) = (EXP(X) - EXP(-X)) / 2
Hyperbolic Cosine	COSH(X) = (EXP(X) + EXP(-X)) / 2
Hyperbolic Tangent	TANH(X) = EXP(-X) / (EXP(X) + EXP(-X)) * 2 + 1
Hyperbolic Secant	SECH(X) = 2 / (EXP(X) + EXP(-X))
Hyperbolic Cosecant	CSCH(X) = 2 / (EXP(X) - EXP(-X))
Hyperbolic Cotangent	COTH(X) = EXP(-X) / (EXP(X) - EXP(-X)) * 2 + 1
Inverse Hyperbolic Sine	ARCSINH(X) = LOG( X + SQR(X*X + 1) )
Inverse Hyperbolic Cosine	ARCCOSH(X) = LOG( X / SQR(X*X - 1) )
Inverse Hyperbolic Tangent	ARCTANH(X) = LOG( (1 + X) / (1 - X) ) / 2
Inverse Hyperbolic Secant	ARCSECH(X) = LOG( SQR(-X*X + 1) + 1/X )
Inverse Hyperbolic Cosecant	ARCCSCH(X) = LOG( X / SQR(X*X - 1) ) + (SGN(X) - 1)*pi/2
Inverse Hyperbolic Cotangent	ARCCOTH(X) = LOG(X) + pi/2

### Trigonometry Rules



SIN θ	c / a	Opposite / Hypotenuse
COS θ	b / a	Adjacent / Hypotenuse
TAN θ	c / b	Opposite / Adjacent
CSC θ	a / c	Hypotenuse / Opposite
SEC θ	a / b	Hypotenuse / Adjacent
COT θ	b / c	Adjacent / Hypotenuse

# Unit Conversion Table

Avoidupois: indicates regular English measure – based on 16 ounces to the pound.

To Convert:	Multiply by:	To Get:
<b>A</b>		
Abcoulombs	2.998 x 10 <sup>10</sup>	Statcoulombs
Acres	160	Rods
Acres	10	Square Chains (Gunters)
Acres	43560	Square Feet
Acres	0.4047	Hectares
Acres	100000	Square Links (Gunters)
Acres	4047	Square Meters
Acres	0.0016	Square Miles
Acres	4840	Square Yards
Acre Feet	43560	Cubic Feet
Acre Feet	1233.48	Cubic Meters
Acre Feet	3.259 x 10 <sup>5</sup>	Gallons
Amperes/Square Centimeters	6.452	Amps/Square Inch
Amperes/Square Inch	0.1550	Amps/Square Centimeter
Ampere-Hours	3600	Coulombs
Ampere-Hours	0.03731	Faradays
Ampere-Turns	1.257	Gilberts
Ampere-Turns/Inch	0.4950	Gilberts/Centimeter
Ampere-Turns/Meter	0.01257	Gilberts/Centimeter
Angstroms	3937 x 10 <sup>9</sup>	Inches
Angstroms	10 <sup>10</sup>	Meters
Angstroms	10 <sup>4</sup>	Microns
Ares	0.02471	Acres (US.)
Ares	119.60	Square Yards
Ares	100	Square Meters
Arpens (French measure)	58.47131	Meters
Arpens (French area measure)	0.3418894	Hectares
Astronomical Units	1.49597870 x 10 <sup>8</sup>	Kilometers
Atmospheres (atm.)	76.0	Centimeters-Mercury
Atmospheres	33.90	Feet of Water (at 4° C)
Atmospheres	29.92	Inches-Mercury (at 0° C)
Atmospheres	1.0333	Kilogram/Square Centimeters
Atmospheres	14.70	Pounds/Square Inch
Atmospheres	1.058	Tons/Square Foot
Atmospheres	0.007348	Tons/Square Inch
Atomic Mass Units (amu)	1.660531 x 10 <sup>27</sup>	Kilograms
<b>B</b>		
Barrels (US.) (dry)	7056	Cubic Inches
Barrels (US.) (dry)	105	Quarts (dry)
Barrels (US.) (liquid)	31.5	Gallons (US.)
Barrels (oil)	42	Gallons (oil)
Bars	0.9869	Atmospheres
Bars	10 <sup>6</sup>	Dynes/Square Centimeter
Bars	1.020 x 10 <sup>4</sup>	Kilograms/Square Meter
Bars	2089	Pounds/Square Foot
Bars	14.50	Pounds/Square Inch
Baryls	1.0	Dynes/Square Centimeter
Bolts (US.) (cloth)	36.576	Meters
Board Feet	2359.7	Cubic Centimeters
Board Feet	144	Cubic Inches
British Thermal Units (BTU)	1.0550 x 10 <sup>10</sup>	Ergs
BTU	778.3	Foot-Pounds
BTU	252.0	Gram-Calories
BTU	3.931 x 10 <sup>4</sup>	Horsepower-Hours
BTU	1054.8	Joules
BTU	2.928 x 10 <sup>4</sup>	Kilowatt-Hours
BTU	107.5	Kilowatt-Meters
BTU	10.409	Liter-Atmospheres
BTU/Hour	0.2162	Foot-Pounds/Second
BTU/Hour	0.0700	Gram-Calories/Second
BTU/Hour	3.929 x 10 <sup>4</sup>	Horsepower-Hours
BTU/Hour	0.2931	Watts
BTU/Minute	12.96	Foot-Pounds/Second
BTU/Minute	0.02356	Horsepower
BTU (thermochemical)/Minute	17.57250	Watts
BTU (International)/Minute	17.58426	Watts
BTU/Square Foot/Minute	0.1221	Watts/Square Inch
Bucket (British) (dry)	1.818 x 10 <sup>4</sup>	Cubic Centimeters
Bushel (struck measure)	4	Pecks
Bushel (struck measure)	32	Dry Quarts
Bushel (struck measure)	1.2445	Cubic Feet
Bushel (struck measure)	2150.42	Cubic Inches
Bushel (struck measure)	35.238	Liters
Bushel (struck measure)	64.0	Pints (dry)
Bushel (struck measure)	32.0	Quarts (dry)
Bushel (heaped)	1.278	Bushels (struck measure)
Bushel (heaped)	2747.715	Cubic Inches
<b>C</b>		
Calory-grams	3.96832 x 10 <sup>3</sup>	British Thermal Units
Candle/Square Centimeter	3.142	Lamberts
Candle/Square Inch	0.4870	Lamberts
Carat (c.)	3.086	Grains
Carat	200	Milligrams
Celsius	(C x 9/5) + 32	Fahrenheit
Centares	1.0	Square Meters
Centigrams (cgm.)	0.01	Grams
Centiliters (cl.)	0.3882	Ounces (US. liquid)

To Convert:	Multiply by:	To Get:
Centiliters	0.6103	Cubic Inches
Centiliters	2.705	Drams
Centimeters (cm.)	0.3937	Inches
Centimeters	10	Millimeters
Centimeters	393.7	Mils
Centimeters	0.01094	Yards
Centimeters/Second	1.1969	Feet/Minute
Centimeters/Second	0.03281	Feet/Second
Centimeters/Second	0.036	Kilometers/Hour
Centimeters/Second	0.1943	Knots
Centimeters/Second	0.6	Meters/Minute
Centimeters/Second	0.02237	Miles/Hour
Centimeters/Second	3.728 x 10 <sup>4</sup>	Miles/Minute
Centimeter-Dynes	1.020 x 10 <sup>3</sup>	Centimeter-Grams
Centimeter-Dynes	1.020 x 10 <sup>8</sup>	Meter-Kilograms
Centimeter-Dynes	7.376 x 10 <sup>4</sup>	Pound-Feet
Centimeter-Grams	980.7	Centimeter-Dynes
Centimeter-Grams	10 <sup>5</sup>	Meter-Kilograms
Centimeter-Grams	7.233 x 10 <sup>5</sup>	Pound-Feet
Centimeters of Mercury	0.01316	Atmospheres
Centimeters of Mercury	0.4461	Feet of Water
Centimeters of Mercury	136.0	Kilograms/Square Meter
Centimeters of Mercury	27.85	Pounds/Square Foot
Centimeters of Mercury	0.1934	Pounds/Square Inch
Central	100	Pounds
Central	45.359	Kilograms
Chains	66.0	Feet
Chains	792.0	Inches
Chains	20.1168	Meters
Chains	22.00	Yards
Circular Mils	5.067 x 10 <sup>6</sup>	Square Centimeters
Circular Mils	7.854 x 10 <sup>7</sup>	Square Inches
Circular Mils	0.7854	Square Mils
Circumference	6.283	Radians
Coal Tubs (NFLD.)	100.0	Pounds
Cord (stacked wood)	3.6246	Cubic Meters
Cord (stacked wood)	128	Cubic Feet
Coulombs	2.998 x 10 <sup>9</sup>	Statcoulombs
Coulombs	6.242 x 10 <sup>18</sup>	Elem. Ch.
Coulombs	1.036 x 10 <sup>5</sup>	Faradays
Coulombs/Square Centimeter	64.52	Coulombs/Square Inch
Cubic Centimeters (cc.)	3.531 x 10 <sup>5</sup>	Cubic Feet
Cubic Centimeters	0.061023	Cubic Inches
Cubic Centimeters	1 x 10 <sup>6</sup>	Cubic Meters
Cubic Centimeters	1.3079 x 10 <sup>6</sup>	Cubic Yards
Cubic Centimeters	2.642 x 10 <sup>4</sup>	Gallons (US.)
Cubic Centimeters	2.199 x 10 <sup>4</sup>	Gallons (Imp.)
Cubic Centimeters	0.0010	Liters
Cubic Centimeters	1.0	Milliliters
Cubic Centimeters	0.0021	Pints (liquid)
Cubic Centimeters	0.0011	Quarts (liquid)
Cubic Feet	1728	Cubic Inches
Cubic Feet	0.02831685	Cubic Meters
Cubic Feet	7.48052	Gallons (US. liquid)
Cubic Feet	28.317	Liters
Cubic Feet	59.84	Pints (US. liquid)
Cubic Feet	29.92	Quarts (US. liquid)
Cubic Feet/Minute	472.0	Cubic Centimeters/Second
Cubic Feet/Minute	0.1247	Gallons/Second
Cubic Feet/Minute	0.4719	Liters/Second
Cubic Feet/Minute	0.0011	Quarts (liquid)
Cubic Feet/Minute	0.0011	Quarts (liquid)
Cubic Feet/Second	448.831	Gallons/Minute
Cubic Feet/Second	0.646317	Million Gallons/Day
Cubic Feet Aluminum	169	Pounds of Aluminum
Cubic Feet Brass	520	Pounds of Brass
Cubic Feet Brick	125 (approx.)	Pounds of Brick
Cubic Feet Cast Iron	450	Pounds of Cast Iron
Cubic Feet Concrete	145	Pounds of Concrete
Cubic Feet Copper	555	Pounds of Copper
Cubic Feet Cork	15	Pounds of Cork
Cubic Feet Glass	160-180	Pounds of Glass
Cubic Feet Gold	1204	Pounds of Gold
Cubic Feet Hardwood	45 (approx.)	Pounds of Hardwood
Cubic Feet Ice	57	Pounds of Ice
Cubic Feet Lead	708	Pounds of Lead
Cubic Feet Silver	655	Pounds of Silver
Cubic Feet Softwood	30 (approx.)	Pounds of Softwood
Cubic Feet Steel	490	Pounds of Steel
Cubic Feet Water	62.43	Pounds of Water
Cubic Inches	16.387	Cubic Centimeters
Cubic Inches	0.0005787	Cubic Feet
Cubic Inches	1.6387 x 10 <sup>5</sup>	Cubic meters
Cubic Inches	2.1433 x 10 <sup>5</sup>	Cubic Yards
Cubic Inches	0.004329	Gallons (US.)
Cubic Inches	0.003605	Gallons (Imp.)
Cubic Inches	0.016387	Liters
Cubic Inches	1.061 x 10 <sup>5</sup>	Mil-Feet
Cubic Inches	4.433	Drams (liquid)

To Convert:	Multiply by:	To Get:
Cubic Inches	0.554	Ounces (liquid)
Cubic Inches	0.03463	Pints (US. liquid)
Cubic Inches	0.01732	Quarts (US. liquid)
Cubic Meters	$1 \times 10^6$	Cubic Centimeters
Cubic Meters	35.31	Cubic Feet
Cubic Meters	61023	Cubic Inches
Cubic Meters	1.308	Cubic Yards
Cubic Meters	264.2	Gallons (US.)
Cubic Meters	220.0	Gallons (Imp.)
Cubic Meters	1000	Liters
Cubic Meters	2113	Pints (US. liquid)
Cubic Meters	1759.4	Pints (Imp. liquid)
Cubic Meters	1057	Quarts (US. liquid)
Cubic Meters	880.1	Quarts (Imp. liquid)
Cubic Tons	40	Cubic Feet
Cubic Tons	1.1327	Cubic Meters
Cubic Yards	27	Cubic Feet
Cubic Yards	46.656	Cubic Inches
Cubic Yards	0.76456	Cubic Meters
Cubic Yards	202.0	Gallons (US.)
Cubic Yards	168.2	Gallons (Imp.)
Cubic Yards	764.5	Liters
Cubic Yards	1615.9	Pints (US. liquid)
Cubic Yards	807.9	Quarts (US. liquid)
Cubic Yards	1345.5	Pints (Imp. liquid)
Cubic Yards	672.7	Quarts (Imp. liquid)
Cubic Yards/Minute	0.45	Cubic Feet/Second
Cubic Yards/Minute	3.367	Gallons/Second
Cubic Yards/Minute	12.74	Liters/Second
Cunits (timber)	100.0	Cubic Feet
Cunits (timber)	2.83168	Cubic Meters
Cups (Cdn.)	227.0	Milliliters
Cups (US.)	236.0	Milliliters
Cups (measuring)	8	Ounces (liquid)
Cups (measuring)	0.5	Pints (liquid)
Cups (measuring)	16	Tablespoons

**D**

Dalton	$1.650 \times 10^{24}$	Grams
Days	86400	Seconds
Degrees (angle)	1.1111	Grads
Degrees (angle)	60	Minutes
Degrees (angle)	0.01111	Quadrants
Degrees (angle)	0.01745 (or $\pi/180$ )	Radians
Degrees (angle)	3600	Seconds
Degrees/Second	0.01745	Radians/Second
Degrees/Second	0.1667	Revolutions/Minute
Degrees/Second	0.002778	Revolutions/Second
Dekaliter (dkl.)	2.642	Gallons (US.)
Dekaliter (dkl.)	3.1729	Gallons (Imp.)
Dekaliter (dkl.)	1.135	Pecks
Drams (dr.) (avoirdupois)	27.3437	Grains
Drams (dr. ap.) (apothecaries')	60	Grains
Drams (apothecaries')	3.888	Grams
Drams (apothecaries')	0.1371429	Ounces (avoirdupois)
Drams (apothecaries')	0.125	Ounces (apothecaries')
Drams (fl. dr.) (liquid) (avoirdupois)	0.0625	Ounces
Drams (liquid) (avoirdupois)	0.2256	Cubic Inches
Drams (liquid) (avoirdupois)	3.6967	Milliliters
Drams (avoirdupois)	1.7718	Grams
Drams (liquid) (British)	0.217	Cubic Inches
Drams (liquid) (British)	0.961	Drams (US. liquid)
Drams (liquid) (British)	3.552	Milliliters
Drops (Cdn. Hospital)	0.01	Teaspoons
Drops (Cdn. Hospital)	0.05	Milliliters
Dynes	$1.020 \times 10^3$	Grams
Dynes	$10^7$	Joules/Centimeter
Dynes	$10^5$	Joules/Meter (Newtons)
Dynes	$7.233 \times 10^5$	Pounds
Dynes	$2.248 \times 10^6$	Pounds
Dynes/Centimeter	0.01	Ergs/Square Millimeter
Dynes/Square Centimeter	$10^6$	Bars
Dynes/Square Centimeter	$9.869 \times 10^7$	Atmospheres
Dynes/Square Centimeter	$2.953 \times 10^5$	Inches of Mercury (at 0° C)
Dynes/Square Centimeter	$4.015 \times 10^4$	Inches of Water (at 4° C)

**E**

Ells	114.30	Centimeters
Ells	45.0	Inches
Ergs	$9.480 \times 10^{11}$	BTU
Ergs	1.0	Dyne-Centimeters
Ergs	$7.3756103 \times 10^8$	Foot-Pounds
Ergs	$0.2389 \times 10^7$	Gram-Calories
Ergs	$1.020 \times 10^3$	Gram-Centimeters
Ergs	$3.7250 \times 10^{14}$	Horsepower-Hours
Ergs	$10^7$	Joules
Ergs	$0.2778 \times 10^{13}$	Kilowatt-Hours
Ergs/Second	$5.688 \times 10^6$	BTU/Minute
Ergs/Second	$4.427 \times 10^6$	Foot-Pounds/Minute
Ergs/Second	$7.3756 \times 10^8$	Foot-Pounds/Second
Ergs/Second	$1.341 \times 10^{10}$	Horsepower
Ergs/Second	$1.433 \times 10^9$	Kilogram-Calories/Minute
Ergs/Second	$10^{10}$	Kilowatts

**F**

Farads	$10^6$	Microfarads
--------	--------	-------------

To Convert:	Multiply by:	To Get:
Faradays	26.80	Ampere-Hours
Faradays	$9.649 \times 10^4$	Coulombs
Faradays/Second	$9.649 \times 10^4$	Amperes (absolute)
Fahrenheit	$(F - 32) \times 5/9$	Celsius
Fathoms	6	Feet
Fathoms	1.828804	Meters
Feet	0.3048	Meters
Feet (French measure)	0.324841	Meters
Feet (US. survey, limited use)	0.3048006	Meters
Feet	$1.2 \times 10^{-4}$	Mils
Feet	$1.645 \times 10^{-4}$	Nautical Miles
Feet	$1.894 \times 10^{-4}$	Statute Miles
Feet of Water	0.02950	Atmospheres
Feet of Water	0.8826	Inches of Mercury
Feet of Water	0.03048	Kilograms/Square Centimeter
Feet of Water	62.43	Pounds/Square Feet
Feet of Water	0.4335	Pounds/Square Inch
Feet/Minute	0.5080	Centimeters/Second
Feet/Minute	0.01829	Kilometers/Hour
Feet/Minute	0.3048	Meters/Minute
Feet/Minute	0.01136	Miles/Hour
Feet/Second	30.48	Centimeters/Second
Feet/Second	1.097	Kilometers/Hour
Feet/Second	0.5921	Knots
Feet/Second	18.29	Meters/Minute
Feet/Second	0.6818	Miles/Hour
Feet/Second	0.01136	Miles/Minute
Firkins	9.0	Gallons
Firkins	40.91	Liters
Foot-Pounds	$1.286 \times 10^3$	British Thermal Units (BTU)
Foot-Pounds	$1.356 \times 10^7$	Ergs
Foot-Pounds	0.3238	Gram-Calories
Foot-Pounds	$5.0505 \times 10^{-7}$	Horsepower-Hours
Foot-Pounds	1.356	Joules
Foot-Pounds	0.1383	Kilogram-Meters
Foot-Pounds	$3.766 \times 10^{-7}$	Kilowatt-Hours
Foot-Pounds/Minute	0.01667	Foot-Pounds/Second
Foot-Pounds/Minute	$3.030 \times 10^5$	Horsepower
Foot-Pounds/Minute	$2.2597 \times 10^5$	Kilowatts
Foot-Pounds/Second	4.6263	BTU/Hour
Foot-Pounds/Second	0.07717	BTU/Minute
Foot-Pounds/Second	$1.818 \times 10^3$	Horsepower
Foot-Pounds/Second	0.01945	Kilogram-Calories/Minute
Foot-Pounds/Second	$1.356 \times 10^3$	Kilowatts
Furlongs	660	Feet
Furlongs	201.168	Meters
Furlongs	0.125	Miles
Furlongs	40	Rods
Furlongs	220	Yards

**G**

Gallons (gal.)	8	Pints (liquid)
Gallons	4	Quarts (liquid)
Gallons Imperial	1.2009	U.S. Gallons
Gallons U.S.	0.8327	Imperial Gallons
Gallons (US.)	3785	Cubic Centimeters
Gallons (US.)	0.1337	Cubic Feet
Gallons (US.)	231	Cubic Inches
Gallons (US.)	0.0038	Cubic Meters
Gallons (US.)	1024	Drams (liquid)
Gallons (US.)	3.785	Liters
Gallons (US.)	32	Gills (liquid)
Gallons (US.)	128	Ounces (US. liquid)
Gallons (Imp.)	4545.6	Cubic Centimeters
Gallons (Imp.)	0.1606	Cubic Feet
Gallons (Imp.)	277.42	Cubic Inches
Gallons (Imp.)	0.00456	Cubic Meters
Gallons (Imp.)	1229.77	Drams (liquid)
Gallons (Imp.)	4.5456	Liters
Gallons (Imp.)	38.43	Gills (liquid)
Gallons (Imp.)	160	Ounces (Imp. liquid)
Gallons (US.) of Water	6.9489	Pounds of Water
Gallons (Imp.) of Water	8.3453	Pounds of Water
Gausses	6.452	Lines/Square Inch
Gausses	$10^8$	Webers/Square Centimeter
Gausses	$6.452 \times 10^8$	Webers/Square Inch
Gilberts	0.7958	Ampere-Turns
Gilberts/Centimeter	2.021	Ampere-Turns/Inch
Gilberts/Centimeter	79.58	Ampere-Turns/Meter
Gill (gi.)	142.07	Cubic Centimeters
Gill	7.219	Cubic Inches
Gill	4	Ounces (US. liquid)
Gill	0.118	Liters
Grade	0.01571	Radians
Grads	0.90	Degrees (angle)
Grains (troy or apothecaries')	1.0	Grains (avoirdupois)
Grains	64.799	Milligrams
Grains	$2.286 \times 10^3$	Ounces (avoirdupois)
Grains	0.04167	Pennyweight (troy)
Grains/US. Gallon	17.118	Parts/Million
Grains/Imp. Gallon	14.286	Parts/Million
Grains/US. Gallon	142.86	Pounds/Million Gallons
Grams (g.)	980.7	Dynes
Grams	15.432	Grains
Grams	$9.807 \times 10^5$	Joules/Centimeter

To Convert:	Multiply by:	To Get:
Grams	9.807 x 10 <sup>3</sup>	Newtons
Grams	0.03527	Ounces (avoirdupois)
Grams	0.03215	Ounces (troy)
Grams	0.07093	Poundals
Grams	0.002205	Pounds
Gram-Calories	3.9683 x 10 <sup>3</sup>	BTU
Gram-Calories	4.1868 x 10 <sup>2</sup>	Ergs
Gram-Calories	3.0880	Foot-Pounds
Gram-Calories	1.5596 x 10 <sup>6</sup>	Horsepower-Hours
Gram-Calories	1.1630 x 10 <sup>6</sup>	Kilowatt-Hours
Gram-Calories/Second	14.286	BTU/Hour
Gram-Centimeters	9.297 x 10 <sup>8</sup>	BTU
Gram-Centimeters	980.7	Ergs
Gram-Centimeters	9.807 x 10 <sup>5</sup>	Joules
Gram-Centimeters	2.343 x 10 <sup>4</sup>	Kilogram-Calories
Gram-Centimeters	10 <sup>5</sup>	Kilogram-Meters
Grams/Centimeter	5.6 x 10 <sup>3</sup>	Pounds/Inch
Grams/Cubic Centimeter	62.43	Pounds/Cubic Foot
Grams/Cubic Centimeter	0.03613	Pounds/Cubic Inch
Grams/Cubic Centimeter	3.405 x 10 <sup>7</sup>	Pounds/Mil-Foot
Grams/Liter	58.417	Grains/Gallon (US.)
Grams/Liter	1000.0	Parts/Million
Grams/Liter	8.345	Pounds/1000 Gallons
Grams/Liter	0.062427	Pounds/Cubic Foot
Grams/Square Centimeter	2.0481	Pounds/Square Feet
<b>H</b>		
Hand	10.16	Centimeters
Hectares	2.471	Acres
Hectares	1.076 x 10 <sup>5</sup>	Square Feet
Hectoliter (hl.)	26.418	Gallons
Hectoliter	2.838	Bushels
Hogsheads (British)	10.114	Cubic Feet
Hogsheads (US.)	8.42184	Cubic Feet
Hogsheads (US.)	63.0	Gallons (US.)
Hogsheads (US.)	52.4	Gallons (Imp.)
Hogsheads (US.)	236.4	Liters
Horsepower	1.014	Horsepower metric
Horsepower (metric)	0.9863	Horsepower
Horsepower	42.44	BTU/Minute
Horsepower	33000	Foot-Pounds/Minute
Horsepower	550	Foot-Pounds/Second
Horsepower (metric)	542.5	Foot-Pounds/Second
Horsepower	10.68	Kilogram-Calories/Minute
Horsepower	0.7457	Kilowatts
Horsepower (boiler)	33479	BTU/Hour
Horsepower (boiler)	9.803	Kilowatts
Horsepower Hours	2547	BTU
Horsepower Hours	2.6845 x 10 <sup>13</sup>	Ergs
Horsepower Hours	1.98 x 10 <sup>6</sup>	Foot-Pounds
Horsepower Hours	641190	Gram-Calories
Horsepower Hours	2.6845 x 10 <sup>6</sup>	Joules
Horsepower Hours	2.737 x 10 <sup>5</sup>	Kilogram-Meters
Hours	0.04167	Days
Hours	0.005952	Weeks
Hundredweights (cwt.) (gross or long)	112	Pounds
Hundredweights (gross or long)	50.802	Kilograms
Hundredweights (gross or long)	0.05	Tons (long)
Hundredweights (net cwt)(net or short)	1600	Ounces (avoirdupois)
Hundredweights (net or short)	100	Pounds
Hundredweights (net or short)	45.359	Kilograms
Hundredweights (net or short)	0.0453592	Tons (metric)
Hundredweights (net or short)	0.0446429	Tons (long or gross)
<b>I</b>		
Inches	2.540	Centimeters
Inches	1.578 x 10 <sup>5</sup>	Miles
Inches	1000	Mils
Inches	6	Picas (typography)
Inches	72	Points (typography)
Inches	2.778 x 10 <sup>2</sup>	Yards
Inches of Mercury	0.03342	Atmospheres
Inches of Mercury	1.133	Feet of Water
Inches of Mercury	0.03453	Kilograms/Square Centimeter
Inches of Mercury	70.73	Pounds/Square Foot
Inches of Mercury	0.4912	Pounds/Square Inch
Inches of Water (at 4° C)	2.458 x 10 <sup>3</sup>	Atmospheres
Inches of Water (at 4° C)	0.07355	Inches of Mercury
Inches of Water (at 4° C)	2.540 x 10 <sup>3</sup>	Kilograms/Square Centimeter
Inches of Water (at 4° C)	0.5781	Ounces/Square Inch
Inches of Water (at 4° C)	5.204	Pounds/Square Foot
Inches of Water (at 4° C)	0.03613	Pounds/Square Inch
International Amperes	0.9998	Amperes (absolute)
International Volts	1.0003	Volts (absolute)
International Volts	1.593 x 10 <sup>19</sup>	Joules (absolute)
International Volts	9.654 x 10 <sup>4</sup>	Joules
<b>J</b>		
Joules	9.478 x 10 <sup>4</sup>	BTU
Joules	10 <sup>7</sup>	Ergs
Joules	0.7376	Foot-Pounds
Joules	2.389 x 10 <sup>4</sup>	Kilogram-Calories
Joules	0.1020	Kilogram-Meters
Joules	2.778 x 10 <sup>7</sup>	Kilowatt-Hours
Joules/Centimeter	1.020 x 10 <sup>4</sup>	Grams
Joules/Centimeter	10 <sup>7</sup>	Dynes

To Convert:	Multiply by:	To Get:
Joules/Centimeter	100.0	Newtons
Joules/Centimeter	723.3	Poundals
Joules/Centimeter	22.48	Pounds
<b>K</b>		
Kilderkins	17	Gallons
Kilderkins	77.28	Liters
Kilogram-Calories	3.968	BTU
Kilogram-Calories	3088	Foot-Pounds
Kilogram-Calories	1.560 x 10 <sup>3</sup>	Horsepower-Hours
Kilogram-Calories	4186	Joules
Kilogram-Calories	4.186	Kilojoules
Kilogram-Calories	426.9	Kilogram-Meters
Kilogram-Calories	1.163 x 10 <sup>3</sup>	Kilowatt-Hours
Kilogram-Meters	9.294 x 10 <sup>3</sup>	BTU
Kilogram-Meters	9.804 x 10 <sup>7</sup>	Ergs
Kilogram-Meters	7.233	Foot-Pounds
Kilogram-Meters	9.804	Joules
Kilogram-Meters	2.342 x 10 <sup>3</sup>	Kilogram-Calories
Kilogram-Meters	2.723 x 10 <sup>6</sup>	Kilowatt-Hours
Kilograms	980665	Dynes
Kilograms	0.09807	Joules/Centimeter
Kilograms	9.807	Newtons
Kilograms	70.93	Poundals
Kilograms	2.2046226	Pounds
Kilograms	0.0685	Slugs
Kilograms	9.842 x 10 <sup>4</sup>	Tons (long)
Kilograms	1.102 x 10 <sup>3</sup>	Tons (short)
Kilograms/Cubic Meter	0.06243	Pounds/Cubic Feet
Kilograms/Cubic Meter	3.613 x 10 <sup>5</sup>	Pounds/Cubic Inch
Kilograms/Cubic Meter	3.405 x 10 <sup>10</sup>	Pounds/Mil Foot
Kilograms/Meter	0.6720	Pounds/Feet
Kilograms/Square Centimeter	980665	Dynes
Kilograms/Square Centimeter	0.9678	Atmospheres
Kilograms/Square Centimeter	32.81	Feet of Water
Kilograms/Square Centimeter	28.96	Inches of Mercury
Kilograms/Square Centimeter	2048	Pounds/Square Foot
Kilograms/Square Centimeter	14.22	Pounds/Square Inch
Kilograms/Square Meter	9.678 x 10 <sup>5</sup>	Atmospheres
Kilograms/Square Meter	98.07 x 10 <sup>6</sup>	Bars
Kilograms/Square Meter	3.281 x 10 <sup>3</sup>	Feet of Water
Kilograms/Square Meter	2.896 x 10 <sup>3</sup>	Inches of Mercury
Kilograms/Square Meter	9.806650	Pascals
Kilograms/Square Meter	0.2048	Pounds/Square Foot
Kilograms/Square Meter	1.422 x 10 <sup>3</sup>	Pounds/Square Inch
Kilograms/Square Millimeter	10 <sup>6</sup>	Kilograms/Square Meter
Kilolines	1000.0	Maxwells
Kilometers	3281	Feet
Kilometers	3.937 x 10 <sup>4</sup>	Inches
Kilometers	0.621371	Miles
Kilometers	1094	Yards
Kilometers/Hour	27.78	Centimeters/Second
Kilometers/Hour	54.68	Feet/Minute
Kilometers/Hour	0.9113	Feet/Second
Kilometers/Hour	0.5396	Knots
Kilometers/Hour	16.67	Meters/Minute
Kilometers/Liter	2.3521458	Miles/Gallon (US.)
Kilometers/Liter	2.8248094	Miles/Gallon (Imp.)
Kilowatts	56.92	BTU/Minute
Kilowatts	44253.7	Foot-Pounds/Minute
Kilowatts	736.7	Foot-Pounds/Second
Kilowatts	1.341003	Horsepower
Kilowatts	14.34	Kilogram-Calories/Minute
Kilowatt-Hours	3413.10	BTU
Kilowatt-Hours	3.60 x 10 <sup>11</sup>	Ergs
Kilowatt-Hours	2.656 x 10 <sup>6</sup>	Foot-Pounds
Kilowatt-Hours	859850	Gram-Calories
Kilowatt-Hours	1.341	Horsepower-Hours
Kilowatt-Hours	3.6 x 10 <sup>6</sup>	Joules
Kilowatt-Hours	3.671 x 10 <sup>5</sup>	Kilogram-Meters
Kilowatt-Hours	3.53	Lbs. of Water evap'd at 212F
Kilowatt-Hours	22.75	... raised from 62 to 212F
Knots	6080	Feet/Hour
Knots	1.689	Feet/Second
Knots	1.8532	Kilometers/Hour
Knots	1.151	Statute Miles/Hour
Knots	2027	Yards/Hour
<b>L</b>		
Leagues (International nautical)	5.556	Kilometers
Leagues (UK nautical)	5.559552	Kilometers
Leagues (US. nautical)	4.828032	Kilometers
Leagues	15,840	Feet
Leagues	3	Miles (approx.)
Leagues	5280	Yards
Legal Subdivisions (Cdn.)	40	Acres
Legal Subdivisions (Cdn.)	0.1618742	Square Kilometers
Light Years	9.46091 x 10 <sup>12</sup>	Kilometers
Light Years	5.9 x 10 <sup>12</sup>	Miles
Lines/Square Centimeter	1.0	Gausses
Lines/Square Inch	0.1550	Gausses
Lines/Square Inch	1.550 x 10 <sup>9</sup>	Webers/Square Centimeter
Lines/Square Inch	10 <sup>8</sup>	Webers/Square Inch
Lines/Square Inch	1.550 x 10 <sup>5</sup>	Webers/Square Meter
Links (Engineers's)	0.010	Chains
Links (Engineers's)	20.1168	Centimeters



To Convert:	Multiply by:	To Get:
Links (Engineers's)	12.0	Inches
Links (Surveyors's)	7.92	Inches
Liters	0.02838	Bushels (US. dry)
Liters	1000	Cubic Centimeters (cc.)
Liters	0.03531	Cubic Feet
Liters	61.025	Cubic Inches
Liters	$1.308 \times 10^{-4}$	Cubic Yards
Liters	0.2642	Gallons (US. liquid)
Liters	0.21999	Gallons (Imp. liquid)
Liters	2.1133	Pints (US. liquid)
Liters	1.75969	Pints (Imp. liquid)
Liters	1.0567	Quarts (US. liquid)
Liters	0.87988	Quarts (Imp. liquid)
Liters	0.908	Quarts (dry)
Liters/Minute	$5.885 \times 10^{-4}$	Cubic Feet/Second
Liters/Minute	$4.4033 \times 10^{-4}$	Gallons (US.)/Second
Liters/Minute	$3.6665 \times 10^{-4}$	Gallons (Imp.)/Second
Lumens	0.07958	Spherical Candle Power
Lumens	0.001496	Watts
Lumens/Square Foot	1.0	Foot Candles
Lumens/Square Foot	10.76	Lumens/Square Meter
Lux	0.0929	Foot Candles

M		
Maxwells	0.001	Kilolines
Maxwells	$10^{*}0.001$	Webers
Megalines	$10^{*}$	Maxwells
Megohms	$10^{**}$	Microhms
Meters	3.2808399	Feet
Meters	39.37	Inches
Meters	$5.396 \times 10^{-4}$	Nautical Miles
Meters	$6.214 \times 10^{-4}$	Statute Miles
Meters	1.0936133	Yards
Meters	1.179	Varas
Meters/Minute	0.05468	Feet/Second
Meters/Minute	0.06	Kilometers/Hour
Meters/Minute	0.03238	Knots
Meters/Minute	0.03728	Miles/Hour
Meters/Second	196.8	Feet/Minute
Meters/Second	3.6	Kilometers/Hour
Meters/Second	2.2369363	Miles/Hour
Meters/Second	0.03728	Miles/Minute
Meter-Kilograms	$9.807 \times 10^{-2}$	Centimeter-Dynes
Meter-Kilograms	$10^{*}$	Centimeter-Grams
Meter-Kilograms	7.233	Pound-Feet
Microns	$10^{**}$	Meters
Miles (UK. Nautical)	1.853184	Kilometers
Miles (US. Nautical)	1.1507794	Miles (Statute)
Miles (US. Nautical)	6.076.11549	Feet
Miles (Statute)	0.8689762	Miles (US. Nautical)
Miles (Statute)	5280	Feet
Miles (Statute)	8	Furlongs
Miles (Statute)	$6.336 \times 10^{4}$	Inches
Miles (Statute)	1.609344	Kilometers
Miles	1760	Yards
Miles/Hour	44.70	Centimeters/Second
Miles/Hour	88	Feet/Minute
Miles/Hour	1.467	Feet/Second
Miles/Hour	0.8684	Knots
Miles/Hour	26.82	Meters/Minute
Miles/Hour	0.4470	Meters/Second
Miles/Minute	2682	Centimeters/Second
Miles/Minute	88	Feet/Second
Miles/Minute	60	Miles/Hour
Mil-Feet	$9.425 \times 10^{-6}$	Cubic Inches
Milliers	1000.0	Kilograms
Milligram (mg.)	0.01543236	Grains
Milligrams/Liter	1.0	Parts/Million
Milliliters (ml.)	1.0	Cubic Centimeters
Milliliters	0.271	Drams (liquid)
Milliliters	16.231	Minims
Milliliters	0.061	Cubic Inches
Millimeters	0.0394	Inches
Million Gallons (US.)/Day	1.54723	Cubic Feet/Second
Million Gallons (Imp.)/Day	1.85815	Cubic Feet/Second
Mils	$2.540 \times 10^{-4}$	Centimeters
Mils	$8.333 \times 10^{-3}$	Feet
Mils	0.001	Inches
Mils	$2.778 \times 10^{-3}$	Yards
Miner's Inches	1.5	Cubic Feet/Minute
Minims (British)	0.059192	Cubic Centimeter
Minims (US. liquid)	1.0408	Minims (British)
Minims (US. liquid)	0.061612	Cubic Centimeter
Minutes (angle)	0.01667	Degrees
Minutes (angle)	$1.852 \times 10^{-1}$	Quadrants
Minutes (angle)	$2.909 \times 10^{-1}$	Radians
Minutes (angle)	60.0	Seconds
Myriagrams	10.0	Kilograms
Myriameters	10.0	Kilometers
Myriawatts	10.0	Kilowatts

N		
Nepers	8.686	Decibels
Newtons	0.2248	Pounds
Newtons	$10^{*}$	Dynes
Newtons/Square Meter	1.0	Pascals

To Convert:	Multiply by:	To Get:
Noggins	1.0	Gills
Noggins	142.1	Milliliters
O		
Ounces (oz.) (avoirdupois)	16	Drams
Ounces (oz.) (apothecaries')	8	Drams
Ounces (avoirdupois)	437.5	Grains
Ounces (oz. t.) (troy or apothecaries')	480	Grains
Ounces (avoirdupois)	28.350	Grams
Ounces (troy or apothecaries')	31.103	Grams
Ounces (troy or apothecaries')	20.0	Pennyweights
Ounces (avoirdupois)	0.0625	Pounds
Ounces (avoirdupois)	0.9115	Ounces (troy)
Ounces (troy)	1.09714	Ounces (troy)
Ounces (avoirdupois)	$2.8349 \times 10^{-5}$	Metric Tons
Ounces US. (liquid)	1.041	Ounces British (liquid)
Ounces British (liquid)	0.961	Ounces US. (liquid)
Ounces (fl. oz.) (US.) (liquid)	1.8047	Cubic Inches
Ounces (US.) (liquid)	29.573	Milliliters
Ounces (liquid)	0.125	Cups
Ounces (liquid)	0.0296	Liters
Ounces (British) (liquid)	1.734	Cubic Inches
Ounces (British) (liquid)	28.412	Milliliters
Ounces/Square Inch	4309	Dynes/Square Centimeter

P		
Pascals	1.0	Newtons/Square Meter
Pascals	0.10197	Kilograms/Square Meter
Pascals	0.020886	Pounds/Square Foot
Pascals	145.03774	Pounds/Square Inch (psi)
Parsecs	$19 \times 10^{13}$	Miles
Parsecs	$3.084 \times 10^{13}$	Kilometers
Parts/Million	0.0584	Grains/Gallon (US.)
Parts/Million	0.07016	Grains/Gallon (Imp.)
Parts/Million	8.345	Pounds/Million Gallons (US.)
Pascals (Newtons/Square Meter)	$1.45136 \times 10^{-4}$	Pounds/Square Inch
Pecks (pk.) (British)	554.6	Cubic Inches
Pecks (British)	9.091901	Liters
Pecks (US.)	0.25	Bushels
Pecks (US.)	537.605	Cubic Inches
Pecks (US.)	8.809582	Liters
Pecks	16	Pints
Pecks	8	Quarts
Pennyweights (dwt.) (troy)	24.0	Grains
Pennyweights (troy)	1.55517	Grams
Pennyweights (troy)	0.05	Ounces (troy)
Pennyweights (troy)	$4.1667 \times 10^{-3}$	Pounds (troy)
Perch (French area measure)	34.18894	Square Meters
Petrograds (sawn timber)	165.0	Cubic Feet
Petrograds (sawn timber)	4.67228	Cubic Meters
Picas (typography)	0.16667 (1/6)	Inches
Picas	0.4233	Centimeters
Pints (liquid)	473.2	Cubic Centimeters
Pints (liquid)	28.875	Cubic Inches
Pints (liquid)	2	Cups
Pints (liquid)	128	Fluid Drams
Pints (liquid)	16	Fluid Ounces
Pints (liquid)	4	Gills
Pints (liquid)	0.4732	Liters
Pints (dry)	33.600	Cubic Inches
Pints (dry)	0.5510	Liters
Planck's Quantum	$6.624 \times 10^{-27}$	Erg-Seconds
Points (typography)	0.08333 (1/12)	Picas
Poise	1.00	Grams/Centimeter-Second
Poundals	13826	Dynes
Poundals	14.10	Grams
Poundals	0.1383	Newtons (Joules/Meter)
Poundals	0.01410	Kilograms
Poundals	0.03108	Pounds
Pound-Feet	$1.356 \times 10^{7}$	Centimeter-Dynes
Pound-Feet	13825	Centimeter-Grams
Pound-Feet	0.13825	Meter-Kilograms
Pounds (lb.) (avoirdupois)	16	Ounces (oz.) (avoirdupois)
Pounds (avoirdupois)	14.5833	Ounces (troy)
Pounds (avoirdupois)	1.21528	Pounds (troy)
Pounds (lb. t.) (troy)	12	Ounces (oz. t.) (troy)
Pounds (troy)	13.1657	Ounces (avoirdupois)
Pounds (troy)	0.82286	Pounds (avoirdupois)
Pounds (avoirdupois)	256	Drams
Pounds (avoirdupois)	7000	Grains
Pounds (avoirdupois)	453.592370	Grams
Pounds (avoirdupois)	4.448	Newtons (Joules/Meter)
Pounds (avoirdupois)	32.17	Poundals
Pounds (avoirdupois)	0.0005	Short Tons
Pounds (troy)	5760	Grains
Pounds (troy)	373.24177	Grams
Pounds (troy)	240.0	Pennyweights (troy)
Pounds (troy)	$3.6735 \times 10^{-4}$	Tons (long)
Pounds (troy)	$3.7324 \times 10^{-4}$	Tons (metric)
Pounds (troy)	$4.1143 \times 10^{-4}$	Tons (short)
Pounds/Cubic Foot	0.01602	Grams/Cubic Centimeter
Pounds/Cubic Foot	$5.787 \times 10^{-4}$	Pounds/Cubic Inch
Pounds/Cubic Foot	$5.456 \times 10^{*}$	Pounds/Mil-Foot
Pounds/Cubic Inch	1728	Pound/Cubic Foot
Pounds/Foot	1.488	Kilograms/Meter
Pounds/Inch	178.6	Grams/Centimeter

To Convert:	Multiply by:	To Get:
Pounds/Mil-Foot	2.306 x 10 <sup>6</sup>	Grams/Cubic Centimeter
Pounds/Square Foot	4.725 x 10 <sup>4</sup>	Atmospheres
Pounds/Square Foot	0.01602	Feet of Water
Pounds/Square Foot	0.01414	Inches of Mercury
Pounds/Square Foot	4.882	Kilograms/Square Meter
Pounds/Square Foot	47.88026	Pascals
Pounds/Square Foot	6.944 x 10 <sup>-3</sup>	Pounds/Square Inch
Pounds/Square Inch	0.06804	Atmospheres
Pounds/Square Inch	2.307	Feet of Water
Pounds/Square Inch	2.036	Inches of Mercury
Pounds/Square Inch	703.1	Kilograms/Square Meter
Pounds/Square Inch	6894.757	Pascals
Pounds/Square Inch	144.0	Pounds/Square Foot
Pounds of Water	0.0160179	Cubic Feet
Pounds of Water	27.68	Cubic Inches
Pounds of Water	0.1198	Gallons (US.)
Pounds of Water	0.09975	Gallons (Imp.)
Pounds of Water/Minute	2.670 x 10 <sup>-4</sup>	Cubic Feet/Second
<b>Q</b>		
Quadrants (angle)	90.0	Degrees
Quadrants (angle)	5400.0	Minutes
Quadrants (angle)	1.571	Radians
Quadrants (angle)	3.24 x 10 <sup>5</sup>	Seconds
Quarters	12.701	Kilograms
Quarters	2.0	Stones
Quarts (qt.) (liquid)	32	Ounces
Quarts (liquid)	256	Drams
Quarts (liquid)	0.25	Gallons
Quarts US. (dry)	0.969	Quarts British
Quarts British (dry)	1.032	Quarts US.
Quarts US. (liquid)	0.833	Quarts British
Quarts British (liquid)	1.201	Quarts US.
Quarts British	69.354	Cubic Inches
Quarts (US.) (dry)	67.201	Cubic Inches
Quarts (US.) (dry)	1.101	Liters
Quarts (US.) (liquid)	0.03342	Cubic Feet
Quarts (US.) (liquid)	57.75	Cubic Inches
Quarts (US.) (liquid)	946.4	Cubic Centimeters
Quarts (US.) (liquid)	1.238 x 10 <sup>-3</sup>	Cubic Yards
Quarts (US.) (liquid)	0.9463	Liters
<b>R</b>		
Radians	57.2958 (or 180/π)	Degrees
Radians	3438	Minutes
Radians	0.6366	Quadrants
Radians	2.063 x 10 <sup>5</sup>	Seconds
Radians/Second	9.549	Revolutions/Minute
Radians/Second	0.1592	Revolutions/Second
Revolutions	4	Quadrants
Revolutions	6.283	Radians
Revolutions/Minute	6	Degrees/Second
Revolutions/Second	360	Degrees/Second
Revolutions/Second	6.283	Radians/Second
Rods (Pole or Perch)	0.25	Chains (Gunthers)
Rods (Pole or Perch)	16.5	Feet
Rods (Pole or Perch)	5.029	Meters
Rods (Pole or Perch)	5.5	Yards
Roods	0.1011714	Hectares
Roods	1210.0	Square Yards
<b>S</b>		
Scruples (s. ap.)	20	Grains
Scruples	1.296	Grams
Seconds (angle)	2.778 x 10 <sup>-4</sup>	Degrees
Seconds (angle)	0.01667	Minutes
Seconds (angle)	3.087 x 10 <sup>-6</sup>	Quadrants
Seconds (angle)	4.8481 x 10 <sup>-6</sup>	Radians
Sections	640	Acres
Sections	1.0	Square Miles
Sections	2.589988	Square Kilometers
Slugs	14.59	Kilograms
Slugs	32.17	Pounds
Slugs	12.57	Steradians
Square Centimeters	1.973 x 10 <sup>6</sup>	Circular Mills
Square Centimeters	0.001076	Square Feet
Square Centimeters	3.861 x 10 <sup>-11</sup>	Square Miles
Square Centimeters	0.1550	Square Inches
Square Centimeters	1.196 x 10 <sup>-4</sup>	Square Yards
Square Feet	2.2957 x 10 <sup>-5</sup>	Acres
Square Feet	1.833 x 10 <sup>6</sup>	Circular Mills
Square Feet	929.0304	Square Centimeters
Square Feet	144	Square Inches
Square Feet	3.5870 x 10 <sup>-6</sup>	Square Miles
Square Feet	9.290 x 10 <sup>4</sup>	Square Millimeters
Square Feet	0.1111	Square Yards
Square Feet (French measure)	105.521	Square Centimeters
Square Inches	1.273 x 10 <sup>6</sup>	Circular Mills
Square Inches	6.4516	Square Centimeters
Square Inches	0.0069	Square Feet
Square Inches	10 <sup>6</sup>	Square Mills
Square Inches	7.716 x 10 <sup>-4</sup>	Square Yards
Square Kilometers	247.1	Acres
Square Kilometers	10 <sup>10</sup>	Square Centimeters
Square Kilometers	1.0764 x 10 <sup>7</sup>	Square Feet
Square Kilometers	1.550 x 10 <sup>9</sup>	Square Inches

To Convert:	Multiply by:	To Get:
Square Kilometers	0.3861	Square Miles
Square Kilometers	1.1960 x 10 <sup>6</sup>	Square Yards
Square Meters	2.471 x 10 <sup>-4</sup>	Acres
Square Meters	10.764	Square Feet
Square Meters	1550.0	Square Inches
Square Meters	3.861 x 10 <sup>-7</sup>	Square Miles
Square Meters	1.1960	Square Yards
Square Miles	640	Acres
Square Miles	27.88 x 10 <sup>6</sup>	Square Feet
Square Miles	2.589998	Square Kilometers
Square Miles	3.0976 x 10 <sup>6</sup>	Square Yards
Square Millimeters	1973.0	Circular Mills
Square Millimeters	0.00153	Square Inches
Square Mills	1.273	Circular Mills
Square Mills	6.452 x 10 <sup>-6</sup>	Square Centimeters
Square Mills	10 <sup>-6</sup>	Square Inches
Square Yards	2.066 x 10 <sup>-4</sup>	Acres
Square Yards	8361.0	Square Centimeters
Square Yards	9	Square Feet
Square Yards	1296	Square Inches
Square Yards	0.8361274	Square Meters
Square Yards	3.2283 x 10 <sup>-7</sup>	Square Miles
Stones	6.3503	Kilograms
Stones	14.0	Pounds
<b>T</b>		
Tablespoons	4	Drams (liquid)
Tablespoons	0.5	Ounces (liquid)
Tablespoons	3	Teaspoons
Tablespoons	14.21	Milliliters
Tablespoons (Cdn. Hospital)	15.0	Milliliters
Tablespoons (UK)	17.8	Milliliters
Tablespoons (US.)	14.8	Milliliters
Teaspoons	4.74	Milliliters
Teaspoons	0.16667	Ounces (liquid avoirdupois)
Teaspoons (Cdn. Hospitals)	5.0	Milliliters
Teaspoons (UK.)	5.92	Milliliters
Teaspoons (US.)	4.93	Milliliters
Tons (gross tn.) (gross or long)	1016.0	Kilograms
Tons (gross or long)	2240	Pounds
Tons (gross or long)	1.120	Tons (net or short)
Tons (gross or long)	1.016	Tons (metric)
Tons (tonne or t.) (metric)	1000	Kilograms
Tons (metric)	0.984	Tons (gross or long)
Tons (metric)	1.1023113	Tons (net or short)
Tons (metric)	2204.623	Pounds
Tons (tn. or net tn.) (short or net)	2000	Pounds
Tons (short or net)	907.1848	Kilograms
Tons (short or net)	32000.0	Ounces (avoirdupois)
Tons (short or net)	29166.66	Ounces (troy)
Tons (short or net)	2430.56	Pounds (troy)
Tons (short or net)	0.89286	Tons (long or gross)
Tons (short or net)	0.90718	Tons (metric)
Tons (short or net)/ Square Foot	9765.0	Kilograms/Square Meter
Tons of Water/24 Hours	83.333	Pounds of Water/Hour
Tons of Water/24 Hours	0.16643	Gallons (US.)/Minute
Tons of Water/24 Hours	0.13858	Gallons (Imp.)/Minute
Tons of Water/24 Hours	1.3349	Cubic Feet/Hour
Townships	36.0	Sections
Townships	93.23957	Square Kilometers
<b>V</b>		
Volts (absolute)	0.003336	Statvolts
Volts (absolute)	1.602 x 10 <sup>-19</sup>	Joules
Volts/Inch	0.39370	Volts/Centimeter
<b>W</b>		
Watts	3.4129	BTU (mean)/Hour
Watts	0.056884	BTU (mean)/Minute
Watts	107.0	Ergs/Second
Watts	44.27	Foot-Pounds/Minute
Watts	0.7378	Foot-Pounds/Second
Watts	0.001341	Horsepower
Watts	0.001360	Horsepower (metric)
Watts	1.0	Joules/Second
Watts	0.01433	Kilogram Calories/Minute
Watts (International)	1.0002	Watts (absolute)
Watt-Hours	3.6 x 10 <sup>11</sup>	Ergs
Watt-Hours	2656	Foot-pounds
Watt-Hours	859.85	Gram-Calories
Watt-Hours	0.001341	Horsepower-Hours
Watt-Hours	367.2	Kilogram-Meters
Webers	10 <sup>6</sup>	Maxwells
Webers	10 <sup>6</sup>	Kilolines
Webers/Square Inch	1.550 x 10 <sup>-7</sup>	Gausses
Webers/Square Inch	10 <sup>6</sup>	Lines/Square Inch
Webers/Square Inch	0.1550	Webers/Square Centimeter
Weber/Square Meter	10 <sup>4</sup>	Gausses
Weber/Square Meter	6.452 x 10 <sup>4</sup>	Gausses
Webers/Square Meter	10 <sup>4</sup>	Webers/Square Centimeter
Webers/Square Meter	6.452 x 10 <sup>-4</sup>	Webers/Square Inch
<b>Y</b>		
Yards	91.44	Centimeters
Yards	4.934 x 10 <sup>-4</sup>	Miles (nautical)
Yards	5.682 x 10 <sup>-4</sup>	Miles (statute)

# Geometric Areas and Volumes

## Nomenclature:

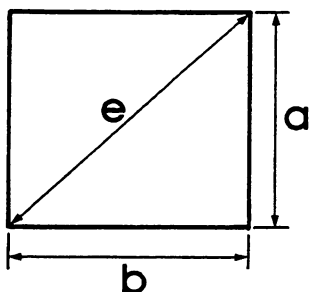
A - Total Area  
 $A_b$  - Area of Base  
 $A_L$  - Area of Lateral Surfaces  
 $A_T$  - Area of Top Section

a,b,c,d - Length of Sides  
 e,f - Angular Lengths  
 h,H - Vertical Height  
 l,L - Arc Length

p - Perimeter  
 $p_b$  - Perimeter of Base  
 $r_1, r_2$  - Radii  
 V - Volume

### Square

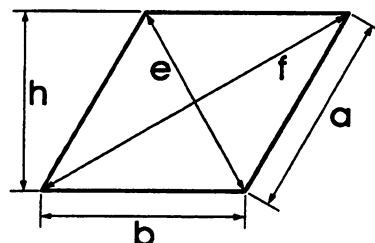
$$\begin{aligned} a &= b \\ p &= 4 \cdot a \\ A &= a \cdot a \\ &= .5 \cdot e \cdot e \\ e &= a \cdot \text{sq}(2) \\ &= a \cdot 1.414 \end{aligned}$$



### Rhombus

(Sides Equal and Parallel)

$$\begin{aligned} a &= b \\ p &= 4 \cdot a = 4 \cdot b \\ e \cdot e + f \cdot f &= 4 \cdot a \cdot a \\ A &= a \cdot h \\ &= e \cdot f / 2 \end{aligned}$$



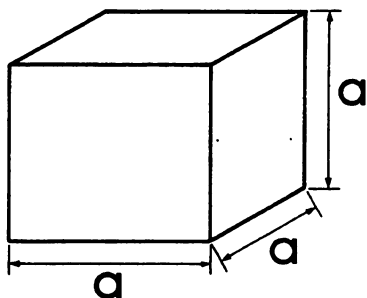
### Parallelogram or Rhomboid

(Sides Parallel but Not Equal)

$$\begin{aligned} p &= 2 \cdot (a + b) \\ e \cdot e + f \cdot f &= 2 \cdot (a \cdot a + b \cdot b) \\ A &= a \cdot h \end{aligned}$$

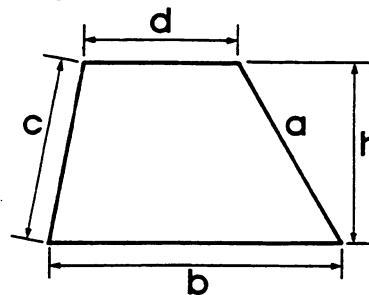
### Cube

$$\begin{aligned} A &= 6 \cdot a \cdot a \\ V &= a \cdot a \cdot a \end{aligned}$$



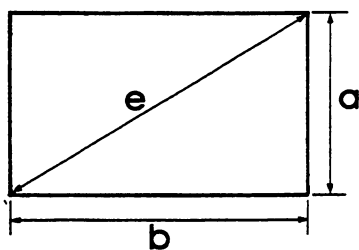
### Trapezoid

$$\begin{aligned} p &= a + b + c + d \\ A &= h \cdot (d + b) / 2 \end{aligned}$$



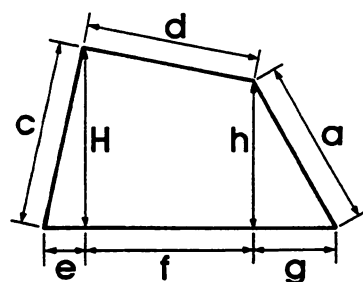
### Rectangle

$$\begin{aligned} p &= 2 \cdot (a + b) \\ e &= \text{sq}(a \cdot a + b \cdot b) \\ a &= \text{sq}(e \cdot e - b \cdot b) \\ A &= a \cdot b \end{aligned}$$



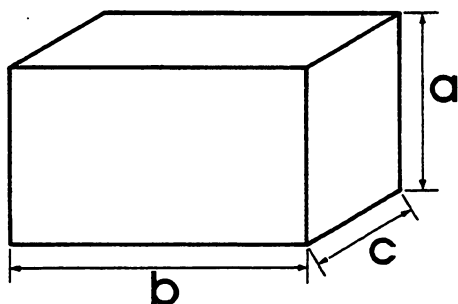
### Trapezium

$$\begin{aligned} p &= a + d + c + e + f + g \\ A &= ((H + h) \cdot f + e \cdot H + g \cdot h) / 2 \end{aligned}$$



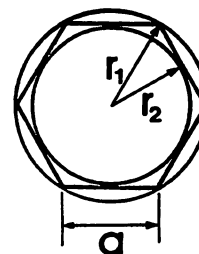
### Parallelopiped

$$\begin{aligned} A &= 2 \cdot (a \cdot b + a \cdot c + b \cdot c) \\ V &= a \cdot b \cdot c \end{aligned}$$



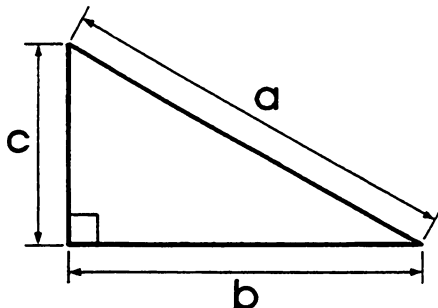
### n-Sided Regular Polygon

$$\begin{aligned} p &= n \cdot a \\ a &= 2 \cdot \text{sq}(r_1 \cdot r_1 - r_2 \cdot r_2) \\ A &= n \cdot a \cdot r_2 / 2 \\ &= n \cdot a / 2 \cdot \text{sq}(r_1 \cdot r_1 - a \cdot a / 4) \\ &= n \cdot \text{area of each triangle} \end{aligned}$$

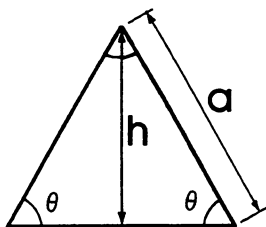


**Right Angled Triangle**

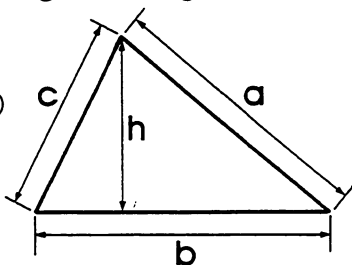
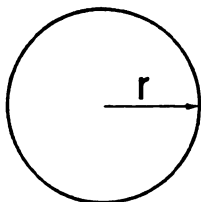
$$\begin{aligned}
 p &= a + b + c \\
 a &= \text{sqr}(b^2 + c^2) \\
 b &= \text{sqr}(a^2 - c^2) \\
 c &= \text{sqr}(a^2 - b^2) \\
 A &= b \cdot c / 2
 \end{aligned}$$

**Equilateral Triangle**

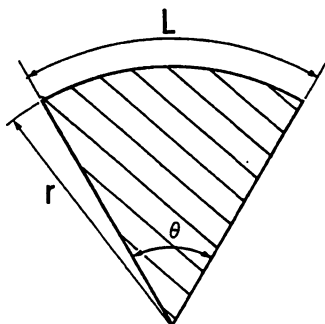
$$\begin{aligned}
 p &= 3 \cdot a \\
 h &= a / 2 \cdot \text{sqr}(3) \\
 &= a \cdot 0.8666 \\
 A &= a \cdot a \cdot \text{sqr}(3) / 4 \\
 &= a \cdot a \cdot 0.4333
 \end{aligned}$$

**General or Oblique Angled Triangle**

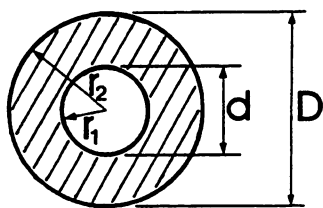
$$\begin{aligned}
 p &= a + b + c \\
 h &= 2 / b \cdot \text{sqr}((s \cdot (s - a)) \cdot (s - b)) \cdot (s - c)) \\
 \text{where } s &= (a + b + c) / 2 \\
 A &= b \cdot h / 2 \\
 \text{or } A &= \text{sqr}((s \cdot (s - a)) \cdot (s - b)) \cdot (s - c))
 \end{aligned}$$

**Circle**

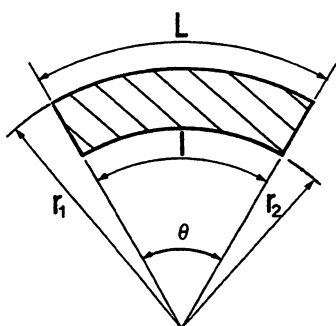
$$\begin{aligned}
 A &= \pi \cdot r \cdot r \\
 p &= 2 \cdot \pi \cdot r
 \end{aligned}$$

**Sector of a Circle**

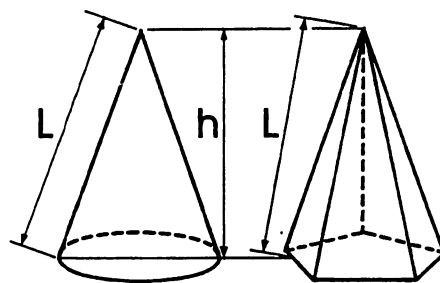
$$\begin{aligned}
 L &= \pi \cdot r \cdot \theta / 180 \\
 &= 2 \cdot A / r \\
 A &= \pi \cdot \theta \cdot r^2 / 360 \\
 &= L \cdot r / 2
 \end{aligned}$$

**Hollow Circle or Annulus**

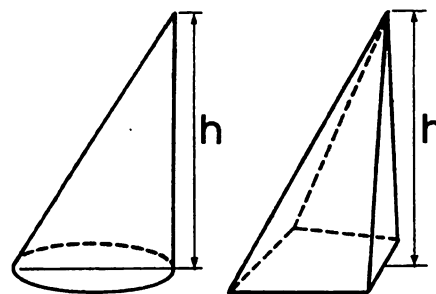
$$\begin{aligned}
 A &= \pi / 4 \cdot (D^2 - d^2) \\
 &= \pi \cdot (r_2^2 - r_1^2) \\
 &= \pi / 2 \cdot (d + D) \cdot (r_2 - r_1) \\
 &= \pi \cdot (r_1 + r_2) \cdot (r_2 - r_1)
 \end{aligned}$$

**Sector of a Hollow Circle**

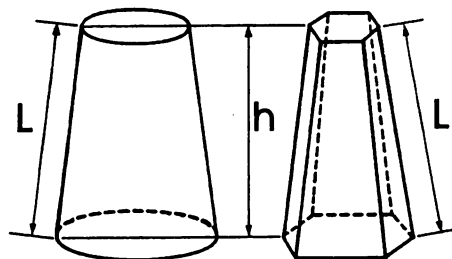
$$\begin{aligned}
 A &= \pi \cdot \theta \cdot (r_2^2 - r_1^2) / 360 \\
 A &= (r_1 - r_2) \cdot (l + L) / 2
 \end{aligned}$$

**Cone or Pyramid (Right Regular)**

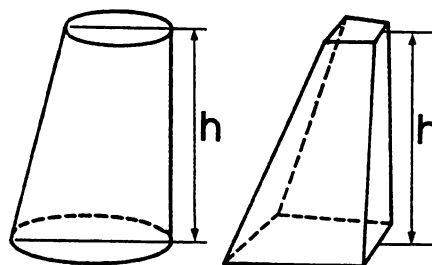
$$\begin{aligned}
 V &= A_B \cdot h / 3 \\
 \text{where } A_B &= \text{area of base} \\
 \text{Lateral surface} &= p_B \cdot L / 2 \\
 \text{where } p_B &= \text{perimeter of base} \\
 A &= \pi \cdot r \cdot \text{sqr}(r^2 + h^2) + \pi \cdot r \cdot r
 \end{aligned}$$

**Cone or Pyramid (General)**

$$\begin{aligned}
 V &= A_B \cdot h / 3 \\
 \text{where } A_B &= \text{area of base}
 \end{aligned}$$

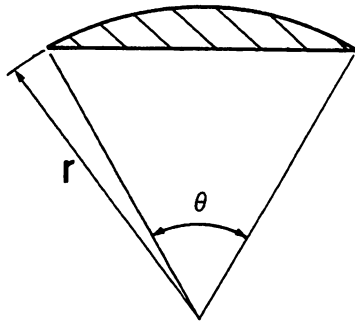
**Frustum of a Cone (Right Regular)**

$$\begin{aligned}
 V &= h \cdot (A_B + A_T + \text{sqr}(A_B \cdot A_T)) / 3 \\
 A_L &= L \cdot (p_B + p_T) / 2 \\
 A &= A_L + A_B + A_T \\
 A_B &= \text{area of base} \\
 A_T &= \text{area of top} \\
 p_B &= \text{perimeter of base} \\
 p_T &= \text{perimeter of top} \\
 A_L &= \text{Lateral surface area}
 \end{aligned}$$

**Frustum of a Cone (General)**

$$\begin{aligned}
 V &= (A_B + A_T + \text{sqr}(A_B \cdot A_T)) \cdot h / 3 \\
 \text{where } A_B &= \text{area of base} \\
 \text{and } A_T &= \text{area of top}
 \end{aligned}$$

### Segment of a Circle



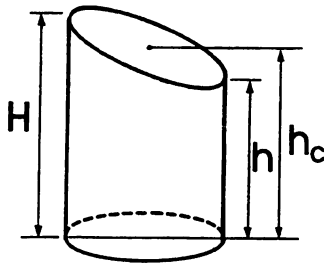
for  $\theta < 90^\circ$ :

$$A = r * r * (\pi * \theta / 180 - \sin(\theta)) / 2$$

for  $\theta > 90^\circ$ :

$$A = r * r * (\pi * \theta / 180 - \sin(180 - \theta)) / 2$$

### Frustum of a Cylinder (Right Circular)



$$A_L = \pi * r * (h + H)$$

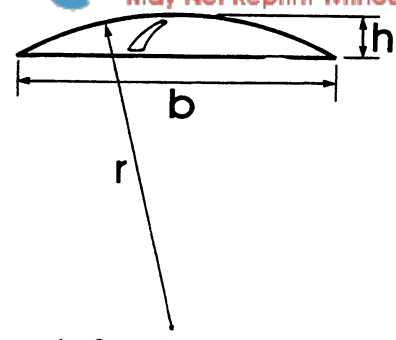
$$A_T = \pi * r * \text{sqrt}(r * r + ((h - H) / 2)^2)$$

$$A_B = \pi * r * r$$

$$A = A_L + A_T + A_B$$

$$V = \pi * r * r * (h + H) / 2$$

### Segment of a Sphere



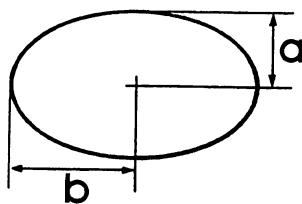
$$A = 2 * \pi * r * h$$

$$\text{or } A = \pi / 4 * (4 * h * h + b * b)$$

$$V = \pi * h * h * (r - h / 3)$$

$$\text{or } V = \pi * h * (b * b / 8 + h * h / 6)$$

### Ellipse



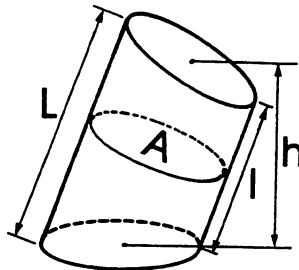
$$p \cong \pi * (a + b)$$

$$p = \pi * (1.5 * (a + b) - \text{sqrt}(a * b))$$

(more accurately)

$$A = \pi * a * b$$

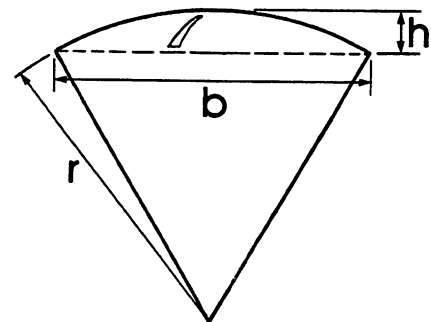
### Frustum of a Cylinder (General)



$$V = A * (L + l) / 2$$

$$V = A_B * h$$

### Sector of a Sphere

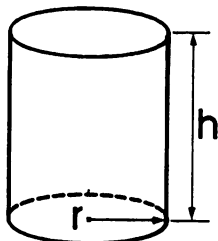


$$A = \pi * r * (2 * h + b / 2)$$

$$b = 2 * \text{sqrt}(h * (2 * r - h))$$

$$V = 2 / 3 * \pi * r * h$$

### Cylinder (Right Circular)

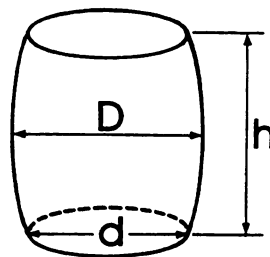


$$A_L = 2 * \pi * r * h$$

$$A = 2 * \pi * r * (r + h)$$

$$V = \pi * r * r * h$$

### Barrel



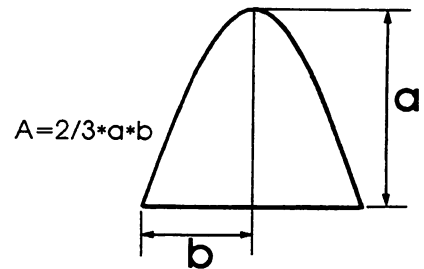
with sides bent to arc of a circle:

$$V = \pi * h * (2 * D * D + d * d) / 12$$

with sides bent to arc of a parabola:

$$V = .209 * h * (2 * D * D + D * d + .75 * d * d)$$

### Parabola



$$A = 2 / 3 * a * b$$

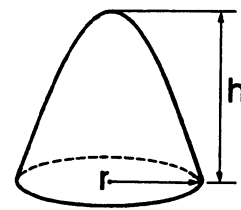
### Paraboloid

$$A = 2 * \pi * (\text{sqrt}((r * r + p * p)^3) - p^3) / (3 * p)$$

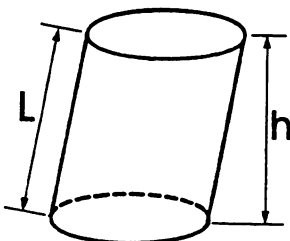
where:

$$p = r * r / (2 * h)$$

$$V = \pi * r * r * h / 2$$



### Cylinder (General)



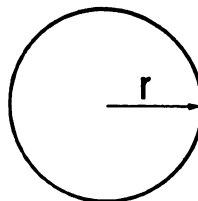
$$A_L = p_B * h$$

$$A = A_L + 2 * A_B$$

$$V = A_B * h$$

where  $A_B$  = area of base ( $\pi * r * r$ )

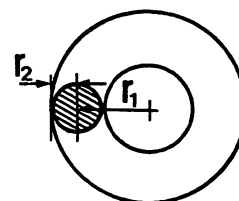
### Sphere



$$A = 4 * \pi * r * r$$

$$V = 4 / 3 * \pi * r^3$$

### Torus (doughnut)



$$A = 4 * \pi * r_1 * r_2$$

$$V = 2 * \pi * r_1 * r_2 * r_1$$

PERIODIC TABLE OF THE ELEMENTS

Table of Selected Radioactive Isotopes

Selected Radioactive Isotopes
Naturally occurring radioactive isotopes are designated by a mass number in blue (although some are also manufactured). Letter m indicates half-life in minutes, h in hours, d in days, and y in years.

Periodic table grid showing elements from Hydrogen (1) to Oganesson (118). Includes atomic numbers, symbols, names, and various isotopic data for radioactive elements.

The A & B subgroup designations, applicable to elements in rows 4, 5, 6, and 7, are those recommended by the International Union of Pure and Applied Chemistry. It should be noted that some authors and organizations use the opposite convention in distinguishing these subgroups.

1 The names and symbols of elements 104-106 are those recommended by IUPAC as systematic alternatives to those suggested by the proposed discoverers. Berkeley (USA) researchers have proposed Subhfermium, Rf, for element 104 and Dubnium, Db, for element 105. The names were derived from the names of the cities in which they were discovered.
2 If there are elements have proposed different names (and symbols).

Continuation of the periodic table from element 119 to 118, including elements like Ununennium, Unbinilium, and others, with their respective atomic numbers and symbols.

Notes and key information:
NOTES: (1) Black - solid, Red - gas, Blue - liquid, Outline - synthetically prepared.
KEY: ATOMIC NUMBER, BOILING POINT, K, MELTING POINT, K, DENSITY at 300K (g/cm³), OXIDATION STATES, ELECTRON CONFIGURATION, NAME.

ARGENT-WELCH SCIENTIFIC COMPANY
7300 NORTH LINDER AVENUE, SKOKIE, ILLINOIS 60077



# **The Complete Commodore Inner Space Anthology**

has been brought to you by the makers of

## **The Transactor** The Tech/News Journal For Commodore Computers

Published once every two months,  
The Transactor brings you detailed and accurate information  
about the Commodore world from the inside out!

Each issue is packed to the limit with concepts, programming techniques,  
hardware projects, events and product news, plus lots more!  
If keeping one step ahead of your computer is the scenario you demand, then  
The Transactor is the most cost effective accessory you can add to your system! And, we're

# **95% Advertising Free!**

Every article is printed back-to-back without interruption by advertisements.

## **The Transactor Disk**

Is also published along with every issue.  
Each disk contains every program from the corresponding magazine in order as they appear.  
There is also a standard set of utility programs included to complement the programs.

# **Subscribe to Both Today!**

**Your Commodore System Will Love You For It!**



# Jim Butterfield's Complete C128 Memory Map

A few issues back we published an abridged C128 RAM/ROM map as prepared by Jim Butterfield. At the time we were quite pleased to have the privilege of publication. Although the maps were not in any way complete, they were good enough to start many hungry programmers on their way with the C128.

After many months of careful and very well calculated pestering on our part, Jim has finally consented to allow us to publish his yet unreleased C128 Map. This opportunity comes as a form of prelude to Jim's yet unreleased new version of, "Machine Language For The Commodore 64 And Other Commodore Computers". Jim has carefully re-written it to include the C128, and as is usual with Jim's books, articles, videos, TV shows, etc., etc., etc., his Machine Language book takes the reader by the hand and gently force feeds knowledge without any painful infliction.

Jim's new book is expected to be released in April of 1986, published by Bradey, a division of Simon and Shuster. As with his last Machine Language book, this version will be available most everywhere through many of the major book stores. If after this incredible bit of JB propaganda you remain unmoved, let me assure you that I am not being paid for this, except for a bottle of Steam beer he bought me in San Francisco (for which I

paid him back promptly). If ever you get the chance, have a read... you will not be disappointed. - RTE

## COMMODORE 128 Memory Maps

Jim Butterfield

These maps apply to the machine when used in the 128K mode. When used in the 64 mode, the machine's map is identical to that of the Commodore 64.

Architecture: "Bank numbers" as used in Basic BANK and the MLM addressing scheme are misleading; in fact, they are more correctly "configuration numbers". Bank 0 shows RAM level 0, which contains work areas and the user's Basic program. Bank 1 also shows RAM, this time (for addresses above hexadecimal 0400) level 1 which contains variables, arrays, and strings. Other "banks" are really configurations, with various types of ROM or I/O overlaying RAM. Thus, bank 15 (the most popular) is ROM and I/O covering RAM bank 0. Bank 14, however, is ROM and the character generator overlaying RAM bank 0. Architecture is set so that addresses below \$0400 reference bank 0 only. Other bank switching (more complex than the simplified 16-bank concept) is accomplished via storing a mask to address \$FF00, or calling up pre-stored masks by writing to \$FF01-FF04.

## The Commodore C128 Memory Map as of February 1986

### All Banks:

Hex	Decimal	Description	0076	118	Graphics flag	00D7	215	40/80 columns: 0 = 40 columns
0000	0	I/O directional register	0077	119	Color source number	00D8	216	Graphics mode code
0001	1	I/O port, similar to C64	0078 -0079	120-121	Temporary counters	00D9	217	Character base: 0 = ROM, 4 = RAM
0002 -0004	2-4	SYS address, MLM registers (SR, PC)	007A -007C	122-124	DSS descriptor	00DA-00DF	218-223	Misc work area
0005 -0009	5-9	SYS, MLM register save (A, X, Y, SR/SP)	007D -007E	125-126	BASIC pseudo-stack pointer	00E0 -00E1	224-225	Pointer to screen line/cursor
000A	10	Scan-quotes flag	007F	127	Flag: 0 = direct mode	00E2 -00E3	226-227	Color line pointer
000B	11	TAB column save	0080 -0081	128-129	DOS, USING work flags	00E4	228	Current screen bottom margin
000C	12	0 = LOAD, 1 = VERIFY	0082	130	Stack pointer save for errors	00E5	229	Current screen top margin
000D	13	Input buffer pointer/number of subscripts	0083	131	Graphic color source	00E6	230	Current screen left margin
000E	14	Default DIM flag	0084	132	Multicolor 1 (1)	00E7	231	Current screen right margin
000F	15	Type: FF = string; 00 = numeric	0085	133	Multicolor 2 (2)	00E8 -00E9	232-233	Input cursor log (row, column)
0010	16	Type: 80 = integer; 00 = floating point	0086	134	Graphic foreground color (13)	00EA	234	End-of-line for input pointer
0011	17	DATA scan/LIST quote/memory flag	0087 -008A	135-138	Graphic scale factors, X & Y	00EB	235	Position of cursor on screen line
0012	18	Subscript/FNx flag	008B -008F	139-143	Graphic work values	00EC	236	Row where cursor lives
0013	19	0 = INPUT; \$40 = GET; \$98 = READ	0090	144	Status word ST	00ED-00EE	237-238	Maximum screen lines, columns
0014	20	ATN sign/Comparison evaluation flag	0091	145	Keyswitch IA: STOP and RVS flags	00EF	239	Current I/O character
0015	21	Current I/O prompt flag	0092	146	Timing constant for tape	00F0	240	Previous character printed
0016 -0017	22-23	Integer value	0093	147	Work value, monitor, LOAD/SAVE	00F1	241	Character color
0018	24	Pointer: temporary string stack	0094	148	Serial output: deferred character flag	00F2	242	Temporary color save
0019 -0023	25-35	Stack for temporary strings	0095	149	Serial deferred character	00F3	243	Screen reverse flag
0024 -0027	36-39	Utility pointer area	0096	150	Cassette work value	00F4	244	0 = direct cursor; else programmed
0028 -002C	40-44	Product area for multiplication	0097	151	Register save	00F5	245	Number of INSERTs outstanding
002D -002E	45-46	Pointer: start-of-BASIC (for bank 0)	0098	152	How many open files	00F6	246	255 = Auto Insert enabled
002F -0030	47-48	Pointer: start-of-variables (bank 1)	0099	153	Input device, normally 0	00F7	247	Text mode lockout
0031 -0032	49-50	Pointer: start-of-arrays	009A	154	Output CMD device, normally 3	00F8	248	0 = Scrolling enabled
0033 -0034	51-52	Pointer: end-of-arrays	009B -009C	155-156	Tape parity, output-received flag	00F9	249	Bell disable
0035 -0036	53-54	Pointer: string-storage (moving down)	009D	157	I/O messages: 192 = all, 64 = errors, 0 = nil	00FA -00FF	250-255	Not used
0037 -0038	55-56	Utility string pointer	009E -009F	158-159	Tape error pointers	0100 -01FF	256-511	Processor stack area
0039 -003A	57-58	Pointer: limit-of-memory (bank 1)	00A0 -00A2	160-162	Jiffy Clock HML	0100 -013E	256-318	Tape error log
003B -003C	59-60	Current BASIC line number	00A3 -00AB	163-171	I/O work bytes	0100 -0124	256-292	DOS work area
003D -003E	61-62	Textpointer: BASIC work point	00AC -00AD	172-173	Pointer: tape buffer, scrolling	0125 -0138	293-312	PRINT/USING work area
003F -0040	63-64	Utility Pointer	00AE -00AF	174-175	Tape end adds/End of program	0200 -02A0	512-672	BASIC input buffer
0041 -0042	65-66	Current DATA line number	00B0 -00B1	176-177	Tape timing constants	02A2 -02AE	674-686	Bank peek subroutine
0043 -0044	67-68	Current DATA address	00B2 -00B3	178-179	Pointer: start of tape buffer	02AF -02BD	687-701	Bank poke subroutine
0045 -0046	69-70	Input vector	00B4 -00B6	180-182	RS-232, Misc work values	02BE -02CC	702-716	Bank compare subroutine
0047 -0048	71-72	Current variable name	00B7	183	Number of characters in file name	02CD -02E2	717-738	JSR to another bank
0049 -004A	73-74	Current variable address	00B8	184	Current logical file	02E3 -02FB	739-763	JMP to another bank
004B -004C	75-76	Variable pointer for FOR/NEXT	00B9	185	Current secondary address	02FC -02FD	764-765	Function execute hook [4C78]
004D -004E	77-78	Y-save; op-save; BASIC pointer save	00BA	186	Current device	0300 -0301	768-769	Error message link
004F	79	Comparison symbol accumulator	00BB -00BC	187-188	Pointer to file name	0302 -0303	770-771	BASIC warm start link
0050 -0055	80-85	Miscellaneous work area, pointers, and so on	00BD -00C5	189-197	I/O work pointers	0304 -0305	772-773	Crunch BASIC tokens link
0056 -0058	86-88	Jump vector for functions	00C6 -00C7	198-199	Banks: I/O data, filename	0306 -0307	774-775	Print tokens link
0059 -0062	89-98	Miscellaneous numeric work area	00C8 -00CB	200-203	RS-232 input/output buffer addresses	0308 -0309	776-777	Start new BASIC code link
0063	99	Accum#1: exponent	00CC -00CD	204-205	Keyboard decode pointer (bank 15)	030A -030B	778-779	Get arithmetic element link
0064 -0067	100-103	Accum#1: mantissa	00CE -00CF	206-207	Print string work pointer	030C -030D	780-781	Crunch FE hook
0068	104	Accum#1: sign	00D0	208	Number of characters in keyboard buffer	030E -030F	782-783	List FE hook
0069	105	Series evaluation constant pointer	00D1	209	Number of programmed chars waiting	0310 -0311	784-785	Execute FE hook
006A -006F	106-111	Accum#2: exponent, and so on	00D2	210	Programmed key character index	0312 -0313	786-787	Unused
0070	112	Sign comparison, Acc#1 versus #2	00D3	211	Key shift flag: 0 = no shift	0314 -0315	788-789	IRQ vector [FA65]
0071	113	Accum#1 lo-order (rounding)	00D4	212	Key code: 88 if no key	0316 -0317	790-791	Break interrupt vector [B003]
0072 -0073	114-115	Cassette buffer len/Series pointer	00D5	213	Key code: 88 if no key	0318 -0319	792-793	NMI interrupt vector [FA40]
0074 -0075	116-117	Auto line number increment	00D6	214	Input from screen/from keyboard	031A -031B	794-795	OPEN vector [EFBD]





E105	RAM Bank Masks	E68E	Set RS-232 Bit Count	EEA8	IRQ Vectors	F53E	--save--	F7AE	Get Char From Memory
E109	-ioint-	E69D	(NMI) RS-232 Receive	EEB0	Kill Tape Motor	F5B5	Terminate Serial Input	F7BC	Store Loaded Byte
E1DC	Set Up CRTC Registers	E75F	Send to RS-232	EEB7	Check End Address	F5BC	Print 'saving'	F7C9	Read Byte to be Saved
E1F0	Check Special Reset	E795	Connect RS-232 Input	EEC1	Bump Address	F5C8	Save to Tape	F7D0	Get Char From Memory Bank
E242	Reset to 64/128	E7CE	Get From RS-232	EEC8	(IRQ) Clear Break	F5FD	--udtim--	F7DA	Store Char to Memory Bank
E24B	Switch to 64 Mode	E7EC	Interlock RS-232/Serial	EED0	Control Tape Motor	F63D	Watch For RUN or Shift	F7E3	Compare Char With Memory Bank
E263	Code to \$02	E805	(NMI) RS-232 Control I/O	EEEB	--getin--	F65E	--rdtim--	F7EC	Load Mem Control Mask
E26B	Scan All ROMs	E850	RS-232 Timings	EF06	--chrin--	F665	--setim--	F7F0	Bank Masks
E2BC	ROM Addresses Hi	E878	(NMI) RS-232 Receive Timing	EF48	Get Char From Tape	F66E	--stop--	F800	Subrtns to \$02A2-\$02FB
E2C0	ROM Banks	E8A9	(NMI) RS-232 Transmit Timing	EF79	--chrout--	F67C	Print 'too many files'	F85A	DMA Code to \$03F0
E2C4	Print 'cbm' Mask	E8D0	Find Any Tape Header	EFBD	--open--	F67F	Print 'file open'	F867	Check Auto Start ROM
E2C7	VIC 8564 Set Up	E919	Write Tape Header	F0B0	Set CIA to RS-232	F682	Print 'file not open'	F890	Check For Boot Disk
E2F8	CRTC 8563 Set Up Pairs	E980	Get Buffer Address	F0CB	Check Serial Open	F685	Print 'file not found'	F90B	Print 'booting'
E33B	--talk--	E987	Get Tape Buffer Start & End Addr	F106	--chkin--	F688	Print 'device not present'	F92F	Print '...'
E33E	--listen--	E99A	Find Specific Header	F14C	--chkout--	F68B	Print 'not input file'	F98B	Wind Up Disk Boot
E43E	--acptr--	E9BE	Bump Tape Pointer	F188	--close--	F68E	Print 'not output file'	F9B3	Read Next Boot Block
E4D2	--second--	E9C8	Print 'press play ...'	F1E4	Delete File	F691	Print 'missing file name'	F9FB	To 2-Digit Decimal
E4E0	--tksa--	E9DF	Check Tape status	F202	Search For File	F694	Print 'illegal device no'	FA08	Block Read
E503	--ciout-- Print Serial	E9E9	Print 'press record.'	F212	Set File Parameters	F697	Error #0	FA15	Print '*i'
E515	--untlk--	E9F2	Initiate Tape Read	F222	--clall--	F6B0	Messages	FA17	Print a Message
E526	--unlsl--	EA15	Initiate Tape Write	F226	--clrchn--	F71E	Print If Direct	FA40	NMI Sequence
E535	Reset ATN	EA26	Common Tape Code	F23D	Clear I/O Path	F722	Print I/O Message	FA65	(IRQ) Normal Entry
E545	Set Clock Low	EA7D	Wait For Tape	F265	--load--	F731	--setnam--	FA80	Keyboard Matrix Un-Shifted
E54E	Set Clock High	EA8F	Check Tape Stop	F27B	Serial Load	F738	--setfls--	FAD9	Keyboard Matrix Shifted
E557	Set Data High	EAA1	Set Read Timing	F32A	Tape Load	F73F	Set Load/Save Bank	FB32	Keyboard Matrix C-Key
E560	Set Data Low	EAEB	(IRQ) Read Tape Bits	F3A1	Disk Load	F744	--rdst--	FB8B	Keyboard Matrix Control
E569	Read Serial Lines	EC1F	Store Tape Chars	F3EA	Burst Load	F757	Set Status Bit	FBE4	Keyboard Matrix Caps Lock
E573	Stabilize Timing	ED51	Reset Pointer	F48C	Close Off Serial	F75C	--setmsg--	FF00	MMU Controls
E59F	Restore Timing	ED5A	New Char Set Up	F4BA	Get Serial Byte	F75F	Set Serial Timeout	FF05	NMI Transfer Entry
E5BC	Prepare For Response	ED69	Send Transit to Tape	F4C5	Receive Serial Byte	F763	--memtop--	FF17	IRQ Transfer Entry
E5C3	Fast Disk Off	ED8B	Write Data to Tape	F503	Toggle Clock Line	F772	--membot--	FF33	Return From Interrupt
E5D6	Fast Disk On	ED90	(IRQ) Tape Write	F50C	Print 'u0' Disk Reset	F781	--iobase--	FF3D	Reset Transfer Entry
E5FB	Fast Disk On/Off	EE2E	(IRQ) Tape Leader	F50F	Print 'searching'	F786	Search For SA	FF47	Jumbo Jump Table
E5FF	(NMI) Transmit RS-232	EE57	Wind Up Tape I/O	F521	Send File Name	F79D	Search & Set Up File	FFFA	Transfer Vectors
E64A	RS-232 Handshake	EE9B	Switch IRQ Vector	F533	Print 'loading'	F7A5	Trigger DMA		

**8502 Processor I/O Registers**

0000	X	0=in	1=out	0=in	1=out	1=out	1=out	00000
0001	X	Caps Key	Tape Motor	Tape Sense	Tape Output	HiRes	LoRes	Color Access

**8722 Memory Management Unit**

D500	RAM select 0-3	HIGH RAM /ROM	MID RAM /ROM	LO RAM	C GEN	54528
D501-D504	Preconfiguration registers: Similar to D500, above					54529-54532
D505	40/80 Key Mode	C64 Mode	Carr-Sense Color-Bank	Fast Disk	X X Z80	54533
D506	Video-Bank	X X	Shared RAM hi	Shared RAM low	0=1K	54534
D507	Zero Page Pointer (\$0000)					L 54535
D508	Stack Page Pointer (\$0000)					H 54536
D509	Zero Page Pointer (\$0000)					L 54537
D50A	Stack Page Pointer (\$0000)					H 54538

**6526 CIA 1 (IRQ)**

(Same as CIA 1 for C64, until DC0C)

DC00	Paddle Select A	Fire	Right	Joystick 0 Left	Down	Up	PRA 56320		
DC01	Keyboard Row Select (inverted)						PRB 56321		
DC02	\$FF - All Output						DDRA 56322		
DC03	\$00 - All Input						DDRB 56323		
DC04	Timer A					L	TAL 56324		
DC05	Timer B					H	TAH 56325		
DC06	Timer A					L	TBL 56326		
DC07	Timer B					H	TBH 56327		
DC0C	Serial (shift) Register						56332		
DC0D	IRQ	X	X	Flag	S.Reg	X	Tim.B	Tim.A	56333
DC0E	S Reg I/O			Load	O/S	Timer A Toggle	Start		56334
DC0F	S Reg I/O			Load	O/S	Timer B	Start		56335

**DMA Controller**

DF00	Busy	Fault	X	X	X	X	X	X	57088
DF01	Exec	Sum	X	X	IRQ	Inc	Mode		57089
DF02	Host Address								L 57090
DF03	Expansion Address								H 57091
DF04	Expansion Bank								L 57092
DF05	Expansion Bank								H 57093
DF06	X	X	X	X	X	Expansion Bank			57094
DF07	Transfer Length								L 57095
DF08	Checksum								H 57096
DF09	Version, Maximum-Memory								57097
DF0A	Version, Maximum-Memory								57098

**6526 CIA 2 (NMI)**

(Same as CIA 2 for C64)

DD00	Serial IN	Clock IN	Serial OUT	Clock OUT	ATN OUT	RS232 IN	Video	Block	PRA 56576
DD01	DSR IN	CYS IN	DCD* IN	RI* IN	DTR OUT	RTS OUT	RS232 IN		PRB** 56577
DD02	IN	IN	OUT	OUT	OUT	OUT	OUT		DDRA 56578
DD03	\$06 for RS232								DDRB 56579
DD04	Timer A							L	TAL 56580
DD05	Timer B							H	TAH 56581
DD06	Timer A							L	TBL 56582
DD07	Timer B							H	TBH 56583
DD0D	RS232 IN			Timer B	Timer A	ICR 56589			
DD0E	Timer A Start			Timer B Start	CRA 56590				
DD0F	Timer B Start			CRB 56591					

\* Connected but not used by O.S.  
\*\* PRB is the Parallel User Port  
DDRA = \$3F at reset

### 8564 Video Chip Control & Miscellaneous Registers

D011	Extended Clr. Mode	Bit Map	Display Enable	Row Select	Y-Scroll	53265			
D012	Raster Register					53266			
D013						X 53267			
D014	Light Pen Input					Y 53268			
D016	x	x	Reset	Multi Colour	Column Select	X-Scroll 53270			
D018	VM13	VM12	Screen VM11	VM10	Character Base CB13	CB12	CB11	x	53272
D019	IRQ	Interrupt Sense:			Light Pen	Spr-Spr Collision	Spr-Back Collision	Raster	53273
D01A	Interrupt Enable:			Light Pen	Spr-Spr Collisions	Spr-Back Collisions	Raster	53274	
<b>Colour Registers</b>									
D020	X	Exterior Colour (Border)				53280			
D021	X	Background Colour #0				53281			
D022	X	Background Colour #1				53282			
D023	X	Background Colour #2				53283			
D024	X	Background Colour #3				53284			
D025	X	Sprite MultiColour #0				53285			
D026	X	Sprite MultiColour #1				53286			
D02F	x	x	x	x	x	[Keyboard Rows] 53295			
D030	X	X	X	X	X	X Test Fast Clock 53296			

### 6581 SID Sound Chip (Identical to 6581 on C64)

Voice 1	Voice 2	Voice 3		Voice 1	Voice 2	Voice 3
D400	D407	D40E	Frequency	L	54272	54279 54286
D401	D408	D40F		L	54273	54280 54287
D402	D409	D410	Pulse Width	L	54274	54281 54288
D403	D40A	D411	0 0 0 0	H	54275	54282 54289
D404	D40B	D412	Voice Type: NSE PUL SAW TRI	Key	54276	54283 54290
D405	D40C	D413	Attack Time: 2ms-8sec	Decay Time: 6ms-24sec	54277	54284 54291
D406	D40D	D414	Sustain Level:	Release Time: 6ms-24sec	54278	54285 54292
Voices are "write-only"						
D415	0	0	0	0	0	L 54293
D416	Filter Frequency					H 54292
D417	Resonance	Ext	Filter Voices V3	V2	V1	54295
D418	Passband V3 off	Hi	BP	LO	Master Volume	54296
Filter and Volume (write only)						
D419	Paddle X (A/D #1)					54297
D41A	Paddle Y (A/D #2)					54298
D41B	Noise 3 (random)					54299
D41C	Envelope 3					54300
Sense (read only)						

Note: Special Voice Features (TEST, RING, MOD, SYNC) are omitted from the above diagram

### 8564 Video Chip Sprite Registers

Sprite 0	Sprite 7		Sprite 0	Sprite 7					
D000	D00E	X Position	53248	53262					
D001	D00F	Y Position	53249	53263					
D027	D02E	Sprite Colour	53287	53294					
Bit For Sprite*:									
	7	6	5	4	3	2	1	0	
D010	X-Position High								53264
D015	Sprite Enable Flags								53269
D017	Y-Expand								53271
D01B	Background Priority								53275
D01C	Sprite MultiColour Mode								53276
D01D	X-Expand								53277
D01E	Interrupt: Sprite Collision								53278
D01F	Interrupt: Background Collision								53279

### 8563 80-Column CRT Controller

D600 read (status):

D600	Status	Light Pen	Vert Blank	X	X	X	X	X	54784
------	--------	-----------	------------	---	---	---	---	---	-------

D600	54784	D601	54785	Typical Value			
0 \$00	Horizontal Total			126			
1 \$01	Horizontal Characters Displayed (R0)			80			
2 \$02	Horizontal Sync position			102			
3 \$03	Vertical Sync Width	Horizontal Sync Width		1 and 3			
4 \$04	X	Vertical Total		32 or 39			
5 \$05	X	X	X	Vertical Total Adjust	0		
6 \$06	X	Vertical Displayed (25)		25			
7 \$07	X	Vertical Sync Position		29 or 32			
8 \$08	X	X	X	X	X	Interlace	0
9 \$09	X	X	X	Scan Lines per Character		7	
10 \$0A	X	Cursor Mode	Cursor Start		32		
11 \$0B	X	X	X	Cursor End		7	
12 \$0C	X	X	Display Address		H 0		
13 \$0D					L 0		
14 \$0E	Cursor Address				H 0		
15 \$0F					L 0		
16 \$10	Light Pen Input				H varies		
17 \$11					L varies		
18 \$12	Video RAM Address (See register 31)				H varies		
19 \$13					L varies		
20 \$14	Colour Address				H 8		
21 \$15					L 0		
22 \$16	Character Total		Character Display Horizontal		120		
23 \$17	X	X	X	Character Display Vertical		8	
24 \$18	Block Copy	Scrn RVS	Blink Rate	V Scroll		32	
25 \$19	Bit Map	Colour Enable	Semi Graph	Wide Pixel	H Scroll		64 or 71
26 \$1A	Foreground Colour		Background Colour		240		
27 \$1B	Scroll Control Horizontal					0	
28 \$1C	Char Set Address	RAM	X	X	X	32	
29 \$1D	X	X	X	Underline Scan Line Count		7	
30 \$1E	Character Count					varies	
31 \$1F	Video RAM data (see registers 18,19)					varies	
32 \$20					H	varies	
33 \$21	Block Copy Start Address				L	varies	
34 \$22					begin	125	
35 \$23	Display Enable				end	100	
36 \$24	X	X	X	X	DRAM Refresh Rate		5



