NEWSLETTER '85
Vol. 1 No. 4

Greetings,

First of all, please allow the editor to say that he is sorry for the delay in getting this issue to you. You see, in between our last meeting and Rainbow Fest alot has happened in this household...It's amazing just how a little baby can keep you from your duties. Again, sorry guys!

LAST MEETING IN REVIEW

There were three product presentations during our last meeting. If you missed them...Too bad, they were great! John Keller reviewed CoCo Paint using the X-Pad. CoCo Paint is yet another graphic software program. Where it is not CoCoMax, it is a close second and at half the price. Ed Hathaway gave a brief showing of a new video digitizer called VIDX by Grafx. Once again a close second to Micro Works DS-69, but at a far less cost. Closing out the meeting, Tony Podraza tryed to show us how to burn a ADOS chip. Due to a bad tape copy of the customized ADOS, the presentation was cut short...Sorry Tony!

JULY MEETING INFORMATION

For the July meeting, I'm sure there will be a product presentation, but at this mailing, one has not been arranged. We will however, be hearing about a very important protection program for all of your software/hardware. If for nothing else, don't miss this information.

TIB-BITS

Attached with this newsletter is a complete listing of the CoCo's memory locations. I believe you will find this information handy in allot of ways. It was published in the December '83 issue of MICRO and is reprinted with-out permission, so mom's the word!!!

OUTSIDE WORLD

Coming soon will be the Commodore 128. Having read a great deal about this new system, this is the low-down. This system will use an upgraded basic called BASIC 7.0 and in 80 columns. It's disk operating system will be CP/M Plus (3.0) with a beige screen (not blue). There will be a throw switch to boot their 64 mode, thus, staying fully compatible with existing software. Commodore claims that this systems separate 64, 128 and CP/M operating modes offer features that would be considered standard on most machines but were lacking on the 64 - including a faster drive (the 1517), 80-column display (on the new 1902 monitor) and memory expansion up to a 512K total. No prices have been set for the 128 of peripherals, yet, according to experts, expect to see it at around $1000 (this price will include all of the above). Commodore feels that using CP/M, it will bring this system into the business world of computers.
Footnote: At the price of this NEW system, check out Tandy's 1000. MS-DOS, disk drive, 128K, etc., etc...with a sale price of $999.95. NOW, how has the edge!!!

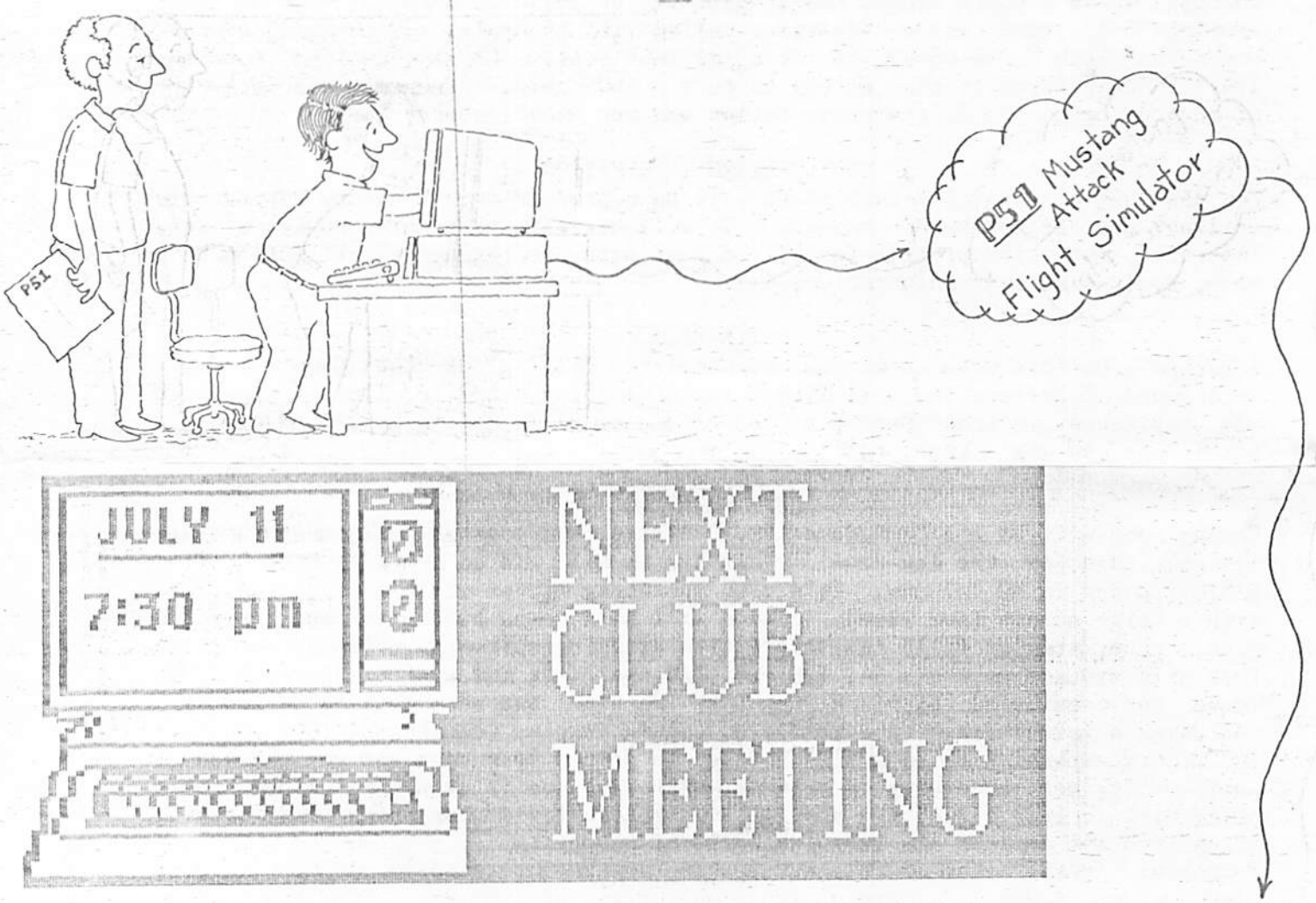
CLOSING NOTES

Remember, July 11th is our next meeting. For the new guys and guests, we meet at the Glenside Public Library on Fullerton Road in Glendale Heights. Our meetings start around 7:30pm and run no latter then 10:00pm. Hope to see you all there in cocoheaven! Oh, the Glenside Color Computer Club is doing a road show at the Glendale Heights Founders Day party. Details at our July meeting.

EDITOR: Ed Hathaway 8 West Stevenson Drive Glendale Hts., IL 60139 (312) 462-0694

G.C.C.C.

Glenside Color Computer Club



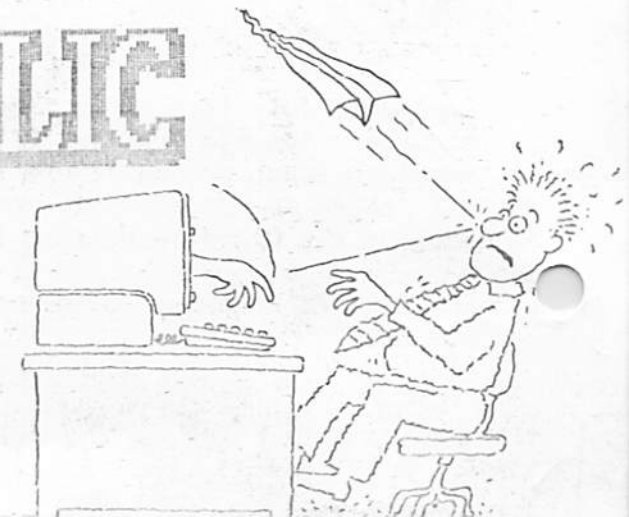
JULY 11
7:30 pm

NEXT CLUB MEETING

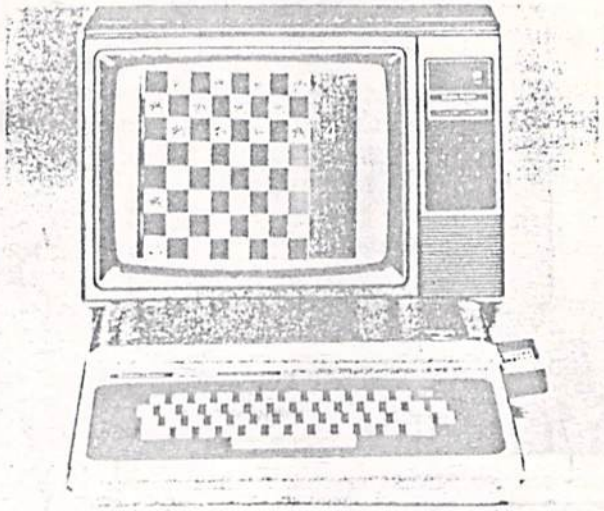
GLENSIDE PUBLIC LIBRARY

FULLERTON AVE.

GLENDALE HEIGHTS



Radio Shack Color Computer Memory Map



(All Numbers in Hex)

Overview

0000-03FF	Ram used by BASIC Interpreter
0400-05FF	Video Display (May be moved)
0600-0FFF	RAM for user program
1000-3FFF	Additional RAM in 16K system
4000-7FFF	Additional RAM in 32K system
8000-9FFF	Extended BASIC ROM
A000-BFFF	Basic Interpreter ROM
C000-FEFF	Cartridge ROM
FF00-FFFF	I/O and Control

Extended

0003	General Counter
0006	String Flag
0007	Flag if Garbage Collected
0019	Start of User RAM
0019-001A	BASIC Program Begin
001B-001C	Pointer to Top of Program/Begin Variables
001D-001E	Pointer to Top of Variables/Start of Arrays
001F-0020	Pointer to End of Arrays/Start of Available Memory
0021-0022	Top of Stack/Start of String Pool
0023-0024	Start of Used Area of String Pool
0025-0026	Pointer to BASIC Memory Limit
0027-0028	End of String Pool/Start of User Space
0033-0034	Pointer to Current Data Read Position
0037-0038	Current Variable Name
0041	4 Bytes Used by Tokenize
0041-0048	Start and End Address of Block Move
0041	Highest Address to Move to
0043	Highest Address to Move
0045	Lowest Address Moved to
0047	Lowest Address to Move
0047	Highest String Found
004B	Address of Descriptor of Highest String Found
004F-0054	Floating Point Accumulator #1 (6 bytes)
0056	String Length
005C-0061	Floating Point Accumulator #2 (6 bytes)
0062	Sign Comparison
0063	Extended Precision Byte
0068-0069	Current Program Line
006C	Current Column Position
006F	Device Number for Output Character (0 = Screen, $FE = Printer, $FF = Tape, 1-16 = Disk BASIC File#)
0070	EOF on Tape File Flag
0071	Reset Flag = $55 for Warmstart
0072-0073	Restart Pointer (contains $80C0-BASIC Warmstart)
0074-0075	Pointer to End of Memory
0078	File Mode (0 = None, 1 = Input, 2 = Output)
0079	Tape Working Buffer Length
007A-007B	Tape Working Buffer Pointer
007C	Tape File Block Type (0 = Header, 1 = Data, $FF = EOF)
007D	Number of Data Bytes in Cassette I/O Block
007E-007F	Program End Address 1 after a CLOADM
0080	Checksum
0081	Cassette Error #
0082	General Counter
0083	Pulse Width Count
0084	Rise/Fall Flag
0085	Last Sine Value
0087	Last Key Entered
0088-0089	Pointer to Current Cursor Position
008A-008B	Serial Read # of Tries
008C	Sound Frequency
008D-008E	Duration of Sound
008F	Start of Area Downloaded from ROM
0092	Controls Length of Unmodulated Carrier Preceeding Casette I/O
0094	Cursor Color
0095-0096	High and Low bytes of Baud Rate Code (Normally $0057)
0097-0098	Carriage Return Delay (Normally $0001)
0099	Comma Field Width (Normally $10)
009A	Last Comma Field (Normally $70)
009B	Printer Line Width (Normally $84)
009C	Affects positions of Vars. Line-printed in Comma Fields ($00)

009D-009E	Transfer Address after CLOADM
009F	Start of get next character subroutine
00A5	Start of get same character subroutine
00A6	Next Character Pointer
00A8-00AA	Jump Vector to Print OK
00AB-00AE	Extended Product Area
00AF	Trace Flag
00B5	Current Color
00B6	Current PMODE
00B7-00B8	End of Screen1
00B9	Number of Bytes per Line
00BA-00BB	Address of Graphics Page
00BC	$E = Disk system, $6 = No disk
00BD	X1
00BF	Y1
00C1	Color Set 1 = 8
00C3	X2
00C5	Y2
00D7	Temp
00DB	Change Flag
00E6	DLOAD Baud Rate
00E7	Input Timeout Constant
00EA	Operation Code
00EB	Drive Number
00EC	Track
00ED	Sector
00EE	Buffer Address
00F0	Status Returned
0100-0102	Software Interrupt 3 Called by Vector at $FFF2
0103-0104	Software Interrupt 2 Called by Vector at $FFF4
0105-0108	Software Interrupt 1 Called by Vector at $FFFA
0109-010B	Non-Maskable Interrupt Called by Vector at $FFFC Set to $D7AE by Disk
010C-010E	Interrupt Request Called by Vector at $FFF8 Set to $A9B3/Set to $894C by Extended/Set to $D7BC by Disk
010F-0111	Fast Interrupt Vector Called by Vector at $FFF6/Set to $A0F6
0112-0113	High and low bytes of TIMER
0116-0117	Seed for RND Function
011A	Shift Lock Flag
011C	Keyboard Delay Constant
011D-011F	Jump vector to $8489-Print OK
0120-013C	Token Table Directory(Byte 1 = # of Keywords, Byte 2,3 = Address of Table, Byte 4,5 = Address of Subroutines)
0120-0124	BASIC Commands
0125-0129	BASIC Functions
012A-012E	Extended BASIC Commands
012F-0133	Extended BASIC Functions
0134-0138	Disk BASIC Commands
0139-013C	Disk BASIC Functions
013E-013F	Address for USR0
0140-0141	Address for USR1
0142-0143	Address for USR2
0144-0145	Address for USR3
0146-0147	Address for USR4
0148-0149	Address for USR5
014A-014B	Address for USR6
014C-014D	Address for USR7
014E-014F	Address for USR8
0150-0151	Address for USR9
0152-0159	Keyboard Rollover Table
015A-015D	Joystick Readings
015A	Left Joystick Up/Down
015B	Left Joystick Left/Right
015C	Right Joystick Up/Down
015D	Right Joystick Right/Left
015E-0160	Open Device Hook Called at $A5F6/Set to $C426 by Disk
0161-0163	Device Number Check Called at $A5B9/Set to $C838 by Disk
0164-0166	Return Device Parameters Called at $A35F/Set to $C843 by Disk
0167-0169	Character Output Called at $A282/Set to $8273 by Extended/Set to $CB4A by Disk
016A-016C	Character Input Called at $A176/Set to $BCF1 by Extended/Set to $C58F by Disk
016D-016F	Check File OPEN for Input Called at $A3ED/Set to $C818 by Disk
0173-0175	Close All Open Files Called at $A426/Set to $CA3B by Disk
0176-0178	Close One File Called at $A42D/Set to $8286 by Extended/Set to $CA4B by Disk
0179-017B	Print Using Called at $B918/Set to $8E90 by Extended
017C-017E	File Item Scanner Called at $B061/Set to $CC5B by Disk
017F-0181	Break Key Check Called at $A549/Set to C859 by Disk
0182-0184	Get Line From Keyboard Called at $A390/Set to JMP RTS by Disk
0185-0187	Finish Loading ASCII File Called at $A4BF/Set to $CA36 by Disk
0188-018A	Check End Of File Called at $A5CE/Set to $C860 by Disk
018B-018D	Evaluate Operand Called at $B223/Set to $8846 by Extended/Set to $CDF6 by Disk
018E-0190	User Error Called at $AC46/Set to JMP RTS by Disk
0191-0193	Error Called at $AC49/Set to $88F0 by Extended/Set to $C24D by Disk
0194-0196	Run Called at $AE75/Set to $829C by Extended/Set to $C990 by Disk
0197-0199	Hex & Octal Called at $BD22/Set to $87E5 by Extended
019A-019C	Execute Line Called at $AD9E/Set to $82B9 by Extended
019D-019F	Graphics Address Called at $A8C4
01A0-01A2	CLS,GET,PUT etc. Called at $A910,$975C,$8AFA,$8162 Set to $C29A by Disk
01A3-01A5	Tokenize Called at $B821/Set to $8304 by Extended
8000-9FFF	Extended BASIC ROM
01D1	Tape File Length
01D2-01D9	Tape File Name
01DA-02D8	Cassette Buffer
01DA-01E1	CLOADM File Name
01EA-01E6	EXEC Address from Tape

01E7-01E8	Load Address from Tape
02DC	Contains token for first keyword in BASIC Statement
02DD-03DC	Console I/O Buffer
0400-05FF	Lo-res screen
0600-35FF	Posible Graphic Screens
0600	Bottom of program area/No Disk
0600-06FF	Disk Buffer
0700-07FF	Disk Buffer
0800-0927	Drive Table
097E	Table of Current Tracks
0982	NMI in use flag
0983	NMI JMP
0985	Motor shutoff counter
0986	Current latch data
0C00	Program Start/Disk System
0FFF	Top of memory (4K)
3FFF	Top of memory (16K)
7FFF	Top of memory (32K)
8000-9FFF	Extended BASIC ROM
807F	Cold Start to BASIC without size Search and Workspace init: Resets pointers to Start of BASIC Program
80C0	Warmstart to BASIC. Does not Reset Pointers to Start of BASIC Prog
8183-81EF	Extended Command Token Table
81F0-821D	Subroutine Entry Addresses
821E-8256	Extended Function Token Table
8257-8272	Subroutine Entry Addresses
82B9	Break or Stop Routine
82BB	Extended interpret loop
8378	COSine
8381	TANgent
83B0	ArcTaNgent
8446	LOG
8480	SQuare Root
84F2	EXPonential
8524	FIX
8533	EDIT
86A7	TRace ON
86A8	TRace OFF
86AC	POSition
86BE	VARiable PoinTeR
874E	STRING$
877E	INSTRing
8871	DEFine
8968	TIMER
8970	DELete
8A09	RENUMber
8BDD	HEX$
8C18	DownLOAD
8DBC	Input Serial Character
8E06	Output Serial Character
928F	Find Byte/Bit Routine
92A6	Byte/Bit; PMODES 0,2,4
92C2	Byte/Bit; PMODES 1,3
92DD	Bit Tables
9339	PPOINT
9361	PSET
9365	PRESET
93BB	LINE
9444	Draw Horizontal Line
946C	Draw Vertical Line
94A1	Draw Line
94E2	The Draw Line Loop
9506	Move Up, Down, Left, Right Routines
9532	PCLS
9546	COLOR
9621	PMODE
9670	SCREEN
968B	PCLEAR
9710	Compare Two Points
9723	PCOPY
9755	GET
9758	PUT
98EC	PAINT
9A22	PLAY
9CB6	DRAW
9E9D	CIRCLE
A000-BFFF	BASIC ROM
A000-A001	Address of Check Keyboard
A002-A003	Address of Character Out
A004-A005	Address of Cassette Read On
A006-A007	Address of Block In
A008-A009	Address of Block Out
A00A-A00B	Address of Joystick In
A00C-A00D	Address of Header Out
A00E	Secondary Reset
A027	Primary Reset
A06E	Hardstart (After Reset)
A0A6	Check for Disk ROM
A0CB	Check for Extended ROM
A0D7	Print Version
A0E8	Softstart (After Reset)
A0F6	FIRQ Entry (ROM Pack Check)
A10D	Start of Area Downloaded to RAM at $8F
A129	Start of Area Downloaded to RAM at $10C
A171	Input Character, Bit 7 Clear
A176	Input Character
A199	Blink Cursor Color
A1B1	Wait for Keypress and Read Kybd; Char Returned in A Register
A1C1	Check Keyboard and Get Key if pressed; Z = 1,A = 0 if no key Z = 0,A = key, B and X Preserved
A26E	Table of Codes for non-alpha keys
A282	Output Character to Device Specified by $6F, All But CC Preserved
A2BF	Output Character in A to Printer (RS232)
A30A	Output Character in A to Screen
A390	Input Line from Keyboard into Buffer at $02DD; Return X$02DC; Zero byte at End of Buffer
A416	CLOSE
A44C	CSAVE
A46C	Perform CSAVEM Function; Requires Start of Memory Block in $19-A0 and in $01E7-8, Transfer Address in $01E5-6, and File Name in $01D2-9. Enter with A = 2 and X = 0.
A498	CLOAD
A4FE	CLOADM
A53E	EXEC

A564	INKEY$
A59A	Transfer Block
A5CE	EOF
A5EC	SKIPF
A5F6	OPEN
A629	Open Tape File
A681	Find Filename
A6FE	Blink Screen Corner
A701	READ Block from Tape
A70B	Read a Block from Cassette; Must be On and In Bit Sync. $7C Contains File Block Type:0 = File Header, 1 = data, $FF = EOF. $7D Contains Number of Data Bytes in File (0-$FF). Z = 1, A = 0 if no Errors, Z = 0, A = 1 if Checksum Error, Z = 0, A = 2 if Memory Error. X = Buffer Start Block Length if no Error, X Points to Beyond Bad Address if Error. U and Y Preserved
A77C	Start Cassette and Get Into Bit Sync for Reading. U and Y Preserved, FIRQ and IRQ Masked.
A7BD	MOTOR
A7D8	Turn Cassette On and Write Leader
A7E5	Write Tape File
A7E9	Turn Off Motor
A7F4	Write Block to Cassette; Tape to Speed and Leader Written, $7E = Buffer Address, $7C = Block Type, $7D = Number of Data Bytes, X = Buffer Address Data Bytes, All Registers Modified
A85C	Sine Table for Cassette Out
A880	SET
A8B1	RESET
A8F5	POINT
A910	CLS
A928	Clear Screen and Home Cursor
A937	Print Copyright (CLS 9)
A94B	SOUND
A956	Generate Sound
A992	AUDIO
A9B3	Interrupt Processor (60 Hz Counter)
A9C6	JOYSTICK
A9DE	Read and Store Joystick Values; Left:Up/Down is $15A,Rt/Lft is $15B; Right:Up/Down is $15C,Rt/Lft is $15D. Y is Preserved
AA29	Function Address Table
AA51	Operation Table for +, −, *, /, ^, AND,OR (3 bytes each-Addresses and Precedence Values)
AA66	Command Name Table
AB1A	Function Name Table
AB67	Command Address Table
ABAF	Error Code Table
ABE1	Text Strings
ABF9	Search Stack for GOSUB or FOR
AC1E	Open up space in memory
AC20	Move Block of Memory Starting at Top; $41-2 is Destination Top Address, $43-4 is Source Top Address, $45-6 is Destination Bottom Address after Move, $47-8 is Source Bottom Address
AC46	Error Handler
AC73	Idle Loop
AD17	NEW (Clear Memory)
AD19	Execute NEW
AD47	FOR
AD9E	Interpret Loop
ADC6	Execute line
ADE4	RESTORE
ADEB	Check for Break or Pause
AE02	END
AE09	STOP
AE30	CONTinue
AE41	CLEAR
AE75	RUN
AE86	GO
AE92	GOSUB
AEA4	GOTO
AEC0	RETURN
AEE0	DATA
AEE3	REM or '
AEE8	ELSE
AF14	IF
AF42	ON
AF67	Get Unsigned Integer
AF89	LET
AFF5	INPUT
B046	READ
B0F8	NEXT
B156	Get Expression
B1CB	Another Entry in Operation Table
B223	Get Operand
B290	Execute Functions
B2D4	AND/OR Operations
B2F4	Relational Operations
B34E	DIMension
B38F	Variable Creation
B3E4	Evaluate Integer Expression
B3ED	Convert Number in FPAC into 16-bit Two's Complement Integer Left in D Register; Overflow, return to BASIC if > +32767 or < -32768
B4EE	MEM
B4FD	STR$
B518	Get String
B56D	Allocate string routine
B591	Garbage Collect
B5D8	Process one descriptor
B5EF	Compact one string
B6F1	LEN
B68C	CHR$
B6A0	ASC
B6AB	LEFT$
B6C8	RIGHT$
B6CF	MID$
B716	VAL
B750	PEEK
B757	POKE
B75E	LLIST Command
B764	LIST Command
B7C2	Untokenize

B7E6	Untokenize one token
B821	Tokenize
B892	Tokenize one word
B8F7	PRINT
B97E	TAB
B99C	Print Text String
B9AC	Print a Space
B9B4	Start of Floating Point Routines-Rounding
B9B9	Subtract from FPAC1
B9C2	Add to FPAC1
BA79	Two's Complement FPAC1
BAC5	Constant 1.0
BACA	Multiply
BB2F	Move [X] to FPAC2
BB7D	Constant 10.0
BB91	Divide
BC4A	Move FPAC2 to FPAC1
BC5F	Move FPAC1 to FPAC2
BC6D	Test FPAC1 for Zero and Sign
BC7A	SiGN
BC93	ABSolute value
BCEE	INTeger
BD12	Convert String to Floating Point
BDB6	Constants 99999999.9, 999999999, 1E09
BDCC	Display the Decimal Value in D Register
BDD9	Convert FPAC1 to ASCII
BEC0	Constant 0.5
BEC5	Series of 4 Byte Constants
BF1F	RaNDom
BF78	SINe
BFBD	Constants 2 pi, 0.25
BFC8	Series of 5 Byte Constants
BFF2	Interrupt and Reset Vectors
BBF2-BBF3	SWI3
BBF4-BBF5	SWI2
BBF6-BBF7	FIRQ
BBF8-BBF9	IRQ
BBFA-BBFB	SWI1
BBFC-BBFD	NMI
BBFE-BBFF	RESET
C000-D7FF	Disk BASIC ROM
C004	Address of DSKCON
C0D4	Warm Start to Disk BASIC
C17F-C1DA	Disk Command Token Table
C1DB-C200	Disk Subroutine Addresses
C6C2	KILL
C932	SAVE
C98B	MERGE
C99A	LOAD
CBCF	DIRectory
CD1A	CVN
CD28	MKN$
CD36	LOC
CD5B	LOF
CDC0	FREE
CDE9	DRIVE
CF3F	RENAME
CF8A	WRITE
CFE0	FIELD
D025	RSET
D026	LSET
D080	FILES
D146	UNLOAD
D175	BACKUP
D2CC	COPY
D3FF	DSKI$
D474	DSKO$
D4AB	DSKINI
D65B	VERIFY
D66C	DSKCON
D6C5	Restore
D6DE	Get Status
D6FD	Delay 78 msec
D705	Read/Write sector
D7A2	Command Address Table
D7AA	Bit Table for Drives
D7AE	NMI Handler
D7BC	IRQ Handler
FF00-FFFF	I/O and Control
FF00-FF03	PIA U8

FF00 Bit 0-Keyboard Row 1 and Right joystick switch
Bit 1-Keyboard Row 2 and Left joystick switch
Bit 2-Keyboard Row 3
Bit 3-keyboard Row 4
Bit 4-Keybbard Row 5
Bit 5-Keyboard Row 6
Bit 6-Keyboard Row 7
Bit 7-Joystick comparison input

FF01 Bit 0-Control of the Horizontal sync clock(63.5 microsec)
Bit 1-;interrupt input
;
Bit 2-Normally 1 0 = Changes FF00 to data direction register
Bit 3-SEL 1: LSB of the two analog MUX select lines
Bit 4-1 Always
Bit 5-1 Always
Bit 6-Not used
Bit 7-Horizontal sync interrupt flag

FF02 Bit 0-Keyboard Column 1
Bit 1-Keyboard Column 2
Bit 2-Keyboard Column 3
Bit 3-Keyboard Column 4
Bit 4-Keyboard Column 5
Bit 5-Keyboard Column 6
Bit 6-Keyboard Column 7
Bit 7-Keyboard Column 8

FF03 Bit 0-Control of the field ;sync clock 16.667 MS
Bit 1-;interrupt input
;
Bit 2-Normally 1 0 = changes FF02 to data direction register
Bit 3-SEL 2 MSB of the two analog MUX select lines
Bit 4-1 Always
Bit 5-1 Always
Bit 6-Not used
Bit 7-Feld sync interrupt flag

FF20-FF23 PIA U4

FF20 Bit 0-Cassette data input
Bit 1-RS-232 data output
Bit 2-6 bit D/A LSB
Bit 3-6 bit D/A
Bit 4-6 bit D/A
Bit 5-6 bit D/A
Bit 6-6 bit D/A
Bit 7-6 bit D/A MSB
FF21 Bit 0-Control of the CD;
;RS-232 status input
Bit 1-;
;
Bit 2-Normally 1
Bit 3-Cassette motor control 0 = Off 1 = On
Bit 4-1 Always
Bit 5-1 Always
Bit 6-Not used
Bit 7-CD interrupt flag
FF22 Bit 0-RS-232 Data input
Bit 1-Single bit sound output
Bit 2-RAM size input
Bit 3-VDG Control output
Bit 4-VDG Control output
Bit 5-VDG Control output
Bit 6-VDG Control output
Bit 7-VDG Control putput
FF23 Bit 0-;Control of the
;Cartridge interrupt
Bit 1-;input
;
Bit 2-Normally 1 — 0 changes FF22 to data direction register
Bit 3-Six bit sound enable
Bit 4-Always 1
Bit 5-Always 1
Bit 6-Not used
Bit 7-Cartridge interrupt flag
FF40 Output latch
Bit 0-Drive Select 0
Bit 1-Drive Select 1
Bit 2-Drive Select 2
Bit 3-Motor On
Bit 4-Precomp
Bit 5-Double Density
Bit 6-Drive Select 3
Bit 7-Halt Enable
FF48 Disk Status
FF49 Disk Track Number
FF4A Disk Sector Number
FF4B Disk Data

FFC0 FFC1	CS	V0	Display	1	0	1	0	1	0	1	0
FFC2 FFC3	CS	V1	Mode Control	1	1	0	0	1	1	0	0
FFC4 FFC5	CS	V2		1	1	1	1	0	0	0	0
FFC6 FFC7	CS	F0	Display Offset (Binary)	AI,AE,S4,S6							
FFC8 FFC9	CS	F1		-G1C,G1R							
FFCA FFCB	CS	F2		-G2C							
FFCC FFCD	CS	F3		-G2R							
FFCE FFCF	CS	F4		-G3C							
FFD0 FFD1	CS	F5		-G3R							
FFD2 FFD3	CS	F6		-G6R,G6C							
FFD4 FFD5	CS	P1	Page #1 (=0)	-N.U.							
FFD6 FFD7	CS	R0	CPU Rate	(Transparent refresh)				1	0	1	0
FFD8 FFD9	CS	R1						1	1	0	0
FFDA FFDB	CS	M0	Memory Size	1	0	1	0	0.9 MHz only			
FFDC FFDD	CS	M1		1	1	0	0	-0.9/1.8 MHz			
								-N.U.			
								-N.U.			
FFDE FFDF	CS	TY	Map Type (=0)	-4k				)			
				-16k				) Dynamic			
				-32/64k				) RAMs			
				-N.U.				)			

FFE0-FFF1 Not used
FFF2-FFF3 SWI3 Vector
FFF4-FFF5 SWI2 Vector
FFF6-FFF6 FIRQ Vector
FFF8-FFF8 IRQ Vector
FFFA-FFFB SWI1 Vector
FFFC-FFFD NMI Vector
FFFE-FFFF Reset Vector

Tokens for CoCo, Dragon 32 and MC – 10

HEX	DEC	COCO	DBL	DRAG	DBL	MC – 10
80	128	FOR	SGN	FOR	SGN	FOR
81	129	GO	INT	GO	INT	GOTO
82	130	REM	ABS	REM	ABS	GOSUB
83	131	'	USR	'	POS	REM
84	132	ELSE	RND	ELSE	RND	IF
85	133	IF	SIN	IF	SQR	DATA
86	134	DATA	PEEK	DATA	LOG	PRINT
87	135	PRINT	LEN	PRINT	EXP	ON
88	136	ON	STR$	ON	SIN	INPUT
89	137	INPUT	VAL	INPUT	COS	END
8A	138	END	ASC	END	TAN	NEXT
8B	139	NEXT	CHR$	NEXT	ATN	DIM
8C	140	DIM	EOF	DIM	PEEK	READ
8D	141	READ	JOYSTK	READ	LEN	LET
8E	142	RUN	LEFT$	LET	STR$	RUN
8F	143	RESTORE	RIGHT$	RUN	VAL	RESTORE
90	144	RETURN	MID$	RESTORE	ASC	RETURN
91	145	STOP	POINT	RETURN	CHR$	STOP
92	146	POKE	INKEY$	STOP	EOF	POKE
93	147	CONT	MEM	POKE	JOYSTK	CONT
94	148	LIST	ATN	CONT	FIX	LIST
95	149	CLEAR	COS	LIST	HEX	CLEAR
96	150	NEW	TAN	CLEAR	LEFT$	NEW
97	151	CLOAD	EXP	NEW	RIGHT$	CLOAD
98	152	CSAVE	FIX	DEF	MID$	CSAVE
99	153	OPEN	LOG	CLOAD	POINT	LLIST
9A	154	CLOSE	POS	CSAVE	INKEY$	LPRINT
9B	155	LLIST	SQR	OPEN	MEM	SET
9C	156	SET	HEX$	CLOSE	VARPTR	RESET
9D	157	RESET	VARPTR	LLIST	INSTR	CLS
9E	158	CLS	INSTR	SET	TIMER	SOUND
9F	159	MOTOR	TIMER	RESET	PPOINT	EXEC
A0	160	SOUND	PPOINT	CLS	STRING$	SKIPF
A1	161	AUDIO	STRING$	MOTOR	USR	TAB(
A2	162	EXEC	CVN	SOUND		TO
A3	163	SKIPF	FREE	AUDIO		THEN
A4	164	TAB(	LOC	EXEC		NOT
A5	165	TO	LOF	SKIPF		STEP
A6	166	SUB	MKN$	DEL		OFF
A7	167	THEN	AS	EDIT		+
A8	168	NOT		TRON		–
A9	169	STEP		TROFF		*
AA	170	OFF		LINE		/
AB	171	+		PCLS		^
AC	172	–		PSET		AND
AD	173	*		PRESET		OR
AE	174	/		SCREEN		>
AF	175	^		PCLEAR		=
B0	176	AND		COLOR		<
B1	177	OR		CIRCLE		SGN
B2	178	>		PAINT		INT
B3	179	=		GET		ABS
B4	180	<		PUT		USR
B5	181	DEL		DRAW		RND
B6	182	EDIT		PCOPY		SQR
B7	183	TRON		PMODE		LOG
B8	184	TROFF		PLAY		EXP
B9	185	DEF		DLOAD		SIN
BA	186	LET		RENUM		COS

(Continued on next page)

BB	187	LINE
BC	188	PCLS
BD	189	PSET
BE	190	PRESET
BF	191	SCREEN
C0	192	PCLEAR
C1	193	COLOR
C2	194	CIRCLE
C3	195	PAINT
C4	196	GET
C5	197	PUT
C6	198	DRAW
C7	199	PCOPY
C8	200	PMODE
C9	201	PLAY
CA	202	DLOAD
CB	203	RENUM
CC	204	FN
CD	205	USING
CE	206	DIR
CF	207	DRIVE
D0	208	FIELD
D1	209	FILES
D2	210	KILL
D3	211	LOAD
D4	212	LSET
D5	213	MERGE
D6	214	RENAME
D7	215	RSET
D8	216	SAVE
D9	217	WRITE
DA	218	VERIFY
DB	219	UNLOAD
DC	220	DSKINI
DD	221	BACKUP
DE	222	COPY
DF	223	DSKI$
E0	224	DSKO$

TAB(	TAN
TO	PEEK
SUB	LEN
FN	STR$
THEN	VAL
NOT	ASC
STEP	CHR$
OFF	LEFT$
+	RIGHT$
−	MID$
*	POINT
/	VARPTR
^	INKEY$
AND	MEM
OR	
>	
=	
<	
USING	

Information was gleaned from the following sources in addition to personal observation:

Color Computer News
The Rainbow
80 Micro
John Beckett
John Steiner
Ralph Tenny

John (Tony) Podraza
119 Adobe Circle
Carpentersville IL 60110