

USER CALLABLE KERNAL ROUTINES				STACK	ERROR
ACPTR	FFA5	65445	Input byte from serial port	13	READST
CHKIN*	FFC6	65478	Open channel for input	-	3,5,6
CHKOUT*	FFC9	65481	Open channel for output	-	3,5,7
CHRIN*	FFCF	65487	Input character from channel	-	READST
CHROUT*	FFD2	65490	Output character to channel	-	READST
CIOUT	FFA8	65448	Output byte to serial port	-	READST
CLALL*	FFE7	65511	Close all channels and files	11	none
CLOSE*	FFC3	65475	Close a specified logical file	-	none
CLRCHN*	FFCC	65484	Restore default I/O devices	9	none
GETIN*	FFE4	65512	Get char from current input dev	-	none
IOBASE	FFF3	65523	Return base address of I/O device	2	none
LISTEN	FFB1	65457	Command devices on serial to listen	-	READST
LOAD*	FFD5	65493	Load (A=0) Verify (A=1) ram	-	0,4,5,8,9
MEMBOT	FF9C	65436	Read (C=1) Set(C=0) bottom of memory	-	none
MEMTOP	FF99	65433	Read (C=1) Set(C=0) top of memory	2	none
OPEN*	FFC0	65472	Open a logical file	-	1,2,4,5,6
PLOT	FFF0	65520	Read (C=1) Set(C=0) x y cursor pos	2	none
RAMTAS	FF87	65415	Init RAM, allocate tape buff, set screen	2	none
RDTIM	FFDE	65502	Read real time clock	2	none
READST	FFB7	65463	Read I/O status word (see page 118)	2	none
RESTOR	FF8A	65415	Restore default I/O vectors	2	none
SAVE*	FFD8	65496	Save RAM to device from \$2B to .X, .Y	-	5,8,9
SCNKEY	FF9F	65439	Scan keyboard	-	none
SCREEN	FFED	65517	Return screen size in rows columns	2	none
SECOND	FF93	65427	Send secondary address after LISTEN	-	READST
SETLFS	FFBA	65466	Set logical, first, and second address	2	none
SETMSG	FF90	65424	Enable/disable KERNAL messages	2	none
SETNAM	FFBD	65469	Set file name	-	none
SETTIM	FFD8	65499	Set real time clock	2	none
SETTMO	FFA2	65442	Set (A<#128) Reset (A>#127) timeout	2	none
STOP*	FFE1	65505	Scan STOP key	-	none
TALK	FFB4	65460	Command serial to TALK	-	READST
TKSA	FF96	65430	Send secondary address after TALK	-	READST
UDTIM	FFEA	65514	Update (increment) real time clock	2	none
UNLSN	FFAE	65454	Command serial to UNLISTEN	-	READST
UNTLK	FFAB	65451	Command serial to UNTALK	-	READST
VECTOR	FF8D	65412	Store (C=1) Restore (C=0) vectors	2	none

*through RAM vector

	PREP. ROUTINES	IN			OUT		
		.A	.X	.Y	.A	.X	.Y
ACPTR	TALK TSKA				data	alt	
CHKIN*	(OPEN)		LF#		alt		
CHKOUT*	(OPEN)		LF#		alt		
CHRIN*	(OPEN CHKIN)				data	alt	
CHROUT*	(CHKOUT OPEN)	data					
CIOUT	LISTEN [SECOND]	data					
CLALL*					alt	alt	
CLOSE*		LF#			alt	alt	alt
CLRCHN*					alt	alt	
GETIN*					data	alt	alt
IOBASE						addr lo	addr hi
LISTEN		DEV#					
LOAD*	SETLFS SETNAM	load/ver	start lo	start hi		end lo +1	end hi
MEMBOT		C=0	bot lo	bot hi	C=1	bot lo	bot hi
MEMTOP		C=0	top lo	top hi	C=1	top lo	top hi
OPEN*	SETLFS SETNAM				alt	alt	alt
PLOT		C=0	row	col	C=1	row	col
RAMTAS					alt	alt	alt
RDTIM					MSB	MSB2	LSB
READST					ST		
RESTOR					alt	alt	alt
SAVE*	SETLFS SETNAM	#<txttab (=#\$28)	end lo	end hi		end lo +1	end hi
SCNKEY					alt	alt	alt
SCREEN						#rows	#cols
SECOND	LISTEN	SA/\$60					
SETLFS		LF#	DEV#	SA			
SETMSG		.A val \$40 control msgs on, \$80 error msgs on.					
SETNAM		len	addr lo	addr hi			
SETTIM		MSB	MSB2	LSB			
SETTMO							
STOP*		yes: .Z = 1 no: .A = last row keyboard scan					
TALK		DEV#					
TKSA	TALK	SA/\$60					
UDTIM					alt	alt	
UNLSN					alt		
UNTLK					alt		
VECTOR		C=1	table lo	table hi	C=0	table lo	table hi

* through RAM vector

USER CALLABLE ROM SUBROUTINES

			IN			OUT		
			.A	.X	.Y	.A	.X	.Y
MAKSPC	C3BB	50107	Ary Top Hi		Ary Top Lo	unaltered		
RAMSPC	C408	50184	same as above, start address of move in \$5F, 60 (5C, 5D)					
MEMERR	C435	50229	direct call					
ERROR	C437	50231	error #					
READY	C474	50292	direct call					
<i>Label?</i>	C48A	50314	(direct call) \$7A=FF,\$7B=01(\$77,78) :01FF=basic Inbuf-1					
NEWLIN	C49C	50220		Lnbf len				
<i>Label?</i>	C52A	50474	direct call					
LNKPRG	C533	50483	direct call					
GETLIN	C560	50528	(direct call) \$7A=FF,\$7B=01(\$77,78) :01FF=basic Inbuf-1					
CRNCH	C579	50553	.X=Inbuf len (\$0200,X)=#\$00					
FINLIN	C613	50707	StrBasLo	StrBasHi				
NEW	C642	50754	direct call					
<i>Label?</i>	C659	50777	direct call					
CLR	C65E	50782	direct call					
STXTPT	C68E	50830	direct call			StrBasHi		
CONT	C857	51287	Cur Lin Lo		Cur Lin Hi			
DECBIN	C96B	49771?	address of text in Chrget ptr: \$7A,7B (\$77,78)					
<i>Label?</i>	CAD3	51923	direct call			LF(\$0A)		
<i>Label?</i>	CAD7	51927	direct call			LF(\$0A)		
PRTSTR	CB1E	51998	addr Lo		addr Hi			
<i>Label?</i>	CB24	52004	length	addr in \$22,23 (1F,20)				
<i>Label?</i>	CB45	52037	direct call					
<i>Label?</i>	CB47	52039	char			char		
FRMEVL \$	CD9E	52638	address of expression			addr Lo		addr Hi
FRMEVL #			in Chrget pointer			result in Acc #1		
SYNCHR	CEFF	52991	direct call			char		
LPACK	CEFA	52986	direct call			char		
RPACK	CEF7	52983	direct call			char		
SYNERR	CF08	53000	direct call					
FNDVAR	D0E7	53479				VarAddrLo		VarAddrHi
RETV	D185	53637	name in \$45,46 (42,43)			VarAddrLo		VarAddrHi
MAKINT	D1BF	53637	direct call					
MAKFP	D391	54049	direct call					
<i>Label?</i>	D79E	55086					data	
<i>Label?</i>	D7B5	55221	address= (Chrget ptr)			float pt result in Acc#1		
<i>Label?</i>	D7B9	55225		addr Lo	addr Hi	float pt result in Acc#1		

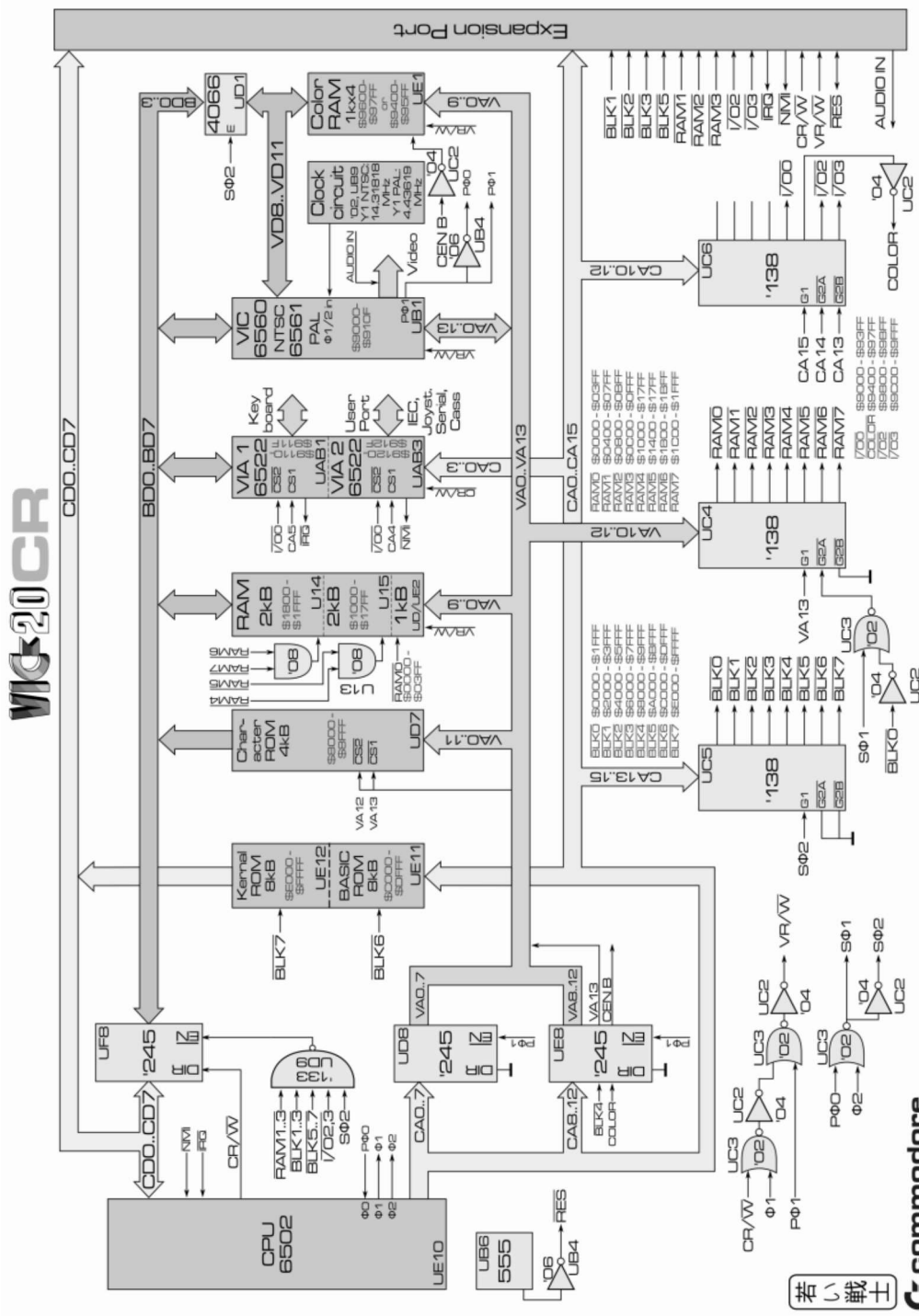
			IN			OUT		
			.A	.X	.Y	.A	.X	.Y
GETAD	D7EB	55275	address= (Chrget ptr)			.X=pram2,pram1 in Acc#1 (fxd)		
LAPLUS	D867	55399	addr Lo		addr Hi	float pt result in Acc#1		
TIMES	DA28	55848	addr Lo		addr Hi	float pt result in Acc#1		
MULTEN	DAE2	56034				result in Acc#1		
LODFAC	DBA2	56226	addr Lo		addr Hi			
<i>Label?</i>	DBD7	56279	addr Lo	addr Hi				
ATOF	DBFC	56316	direct call					
RFTOA	DC0C	56332	direct call					
FTOA	DC0F	56335	direct call					
ROUND	DC1B	56347	direct call					
PRTFIX	DDCD	56781	Value Hi	Value Lo				
<i>Label?</i>	DDD7	56791	direct call					
FLTASC	DDDD	56797	#\$00		#\$01			
SCRNOUT	E742	59202	char					
KMSGSH	F1E6	61926			offset			
TALK	EE14	60948	dev#					
LISTEN	EE17	60951	dev#					
CSECOND	FF93	65427	SA OR \$60					
CIOUT	EEE4	61156	char					
UNTLK	EEF6	61174	direct call					
UNLSN	EF04	61188	direct call					

USER CALLABLE ROM OPERATIONS

MAKSPC	Open up space in BASIC text
RAMSPC	Check available memory (called by 1)
MEMERR	?OUT OF MEMORY
ERROR	Send BASIC error message
READY	Warm start BASIC
C48A	Main CHRGET entry
NEWLIN	Crunch tokens, insert line
C52A	Fix chaining, CLR & READY
LNKPRG	Fix chaining
GETLIN	Receive line from keyboard
CRNCH	Crunch tokens (called by 7)
FINLIN	Find line in BASIC
CLR	Reset BASIC and do CLR
STXTPT	Purge stack of all returns and nexts (POP)
CONT	Continue BASIC execution (CONT)

DECBIN	Get fixed point number from BASIC text
CAD3	Send RETURN, LF if in screen mode
CAD7	Send RETURN, LINEFEED
PRTSTR	Print string from A, Y
CB24	Print precomputed string
CB45	Print “?”
CB47	Print char (output .A to device)
FRMEVL	Evaluate string \$OD =#\$FF (\$07) Evaluate number \$OD =#\$00 (\$07)
SYNCHR	Check for comma
LPACHK	Check for (
RPACHK	Check for)
SYNERR	Send SYNTAX ERROR
FNDVAR	Find floating point variable, given name
RETVP	Bump variable address by 2 (called by 31)
MAKINT	Float to fixed conversion in Acc#1
MAKFP	Fixed to float conversion in Acc#1
D79E	Get Acc#1 LSB to X register
D7B5	Evaluate string (VAL)
D7B9	Evaluate string from X,Y (above +4)
GETAD	Get two parameters for POKE, WAIT
LAPLUS	Add (from memory)
TIMES	Multiply by memory location
MULTEN	Multiply Acc#1 by 10
LODFAC	Unpack memory variable to Acc#1
DBD7	Copy Acc#1 to (X,Y) location
ATOF	Move Acc#2 to Acc#1
RFTOA	Move rounded Acc#1 to A#2
FTOA	Move unrounded Acc#1 to Acc#2
ROUND	Round Acc#1
PRTFIX	Print fixed point value
DDD7	Print floating point value in Acc#1
FLTASC	Convert number to string at \$0100 (called by 48)
SCRNOUT	Print a character
KMSGSH	Print system message
TALK	Send talk to IEEE/Serial
LISTEN	Send listen to IEEE/Serial
CSECOND	Send secondary address
CIOUT	Send char to IEEE/Serial
UNTLK	Send untalk
UNLSN	Send unlisten

Block Diagram



courtesy of Sven Petersen
http://tech.guitarsite.de/vic-20_block.html