

Notice

Apple Computer, Inc. reserves the right to make improvements in the product described in this manual at any time and without notice.

Disclaimer of All Warranties and Liabilities

Apple Computer, Inc. makes no warranties, either express or implied, with respect to this manual or with respect to the software described in this manual, its quality, performance, merchantability, or fitness for any particular purpose. Apple Computer, Inc. software is sold or licensed "as is." The entire risk as to its quality and performance is with the buyer. Should the programs prove defective following their purchase, the buyer (and not Apple Computer, Inc., its distributor, or its retailer) assumes the entire cost of all necessary servicing, repair, or correction and any incidental or consequential damages. In no event will Apple Computer, Inc. be liable for direct, incidental, or consequential damages resulting from any defect in the software, even if Apple Computer, Inc. has been advised of the possibility of such damages. Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you.

This manual is copyrighted. All rights are reserved. This document may not, in whole or part, be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine readable form without prior consent, in writing, from Apple Computer, Inc.

© 1982 by Apple Computer, Inc. 20525 Mariani Avenue Cupertino, California 95014 (408) 996-1010

The word Apple and the Apple logo are registered trademarks of Apple Computer, Inc.

Simultaneously published in the U.S.A and Canada.



Warning

This equipment has been certified to comply with the limits for a Class B computing device, pursuant to Subpart J of Part 15 of FCC Rules. Only peripherals (computer input/output devices, terminals, printers, etc.) certified to comply with the Class B limits may be attached to this computer. Operation with non-certified peripherals is likely to result in interference to radio and TV reception.

Apple *II*e

Reference Manual Addendum: Monitor ROM Listings

Radio and Television Interference

The equipment described in this manual generates and uses radiofrequency energy. If it is not installed and used properly, that is, in strict accordance with our instructions, it may cause interference with radio and television reception.

This equipment has been tested and complies with the limits for a Class B computing device in accordance with the specifications in Subpart J, Part 15, of FCC rules. These rules are designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that the interference will not occur in a particular installation, especially if you use a "rabbit ear" television antenna. (A "rabbit ear" antenna is the telescoping-rod type usually contained on TV receivers.)

You can determine whether your computer is causing interference by turning it off. If the interference stops, it was probably caused by the computer or its peripheral devices. To further isolate the problem:

 Disconnect the peripheral devices and their input/output cables one at a time. If the interference stops, it is caused by either the peripheral device or its I/O cable. These devices usually require shielded I/O cables. For Apple peripheral devices, you can obtain the proper shielded cable from your dealer. For non-Apple peripheral devices, contact the manufacturer or dealer for assistance.

If your computer does cause interference to radio or television reception, you can try to correct the interference by using one or more of the following measures:

- Turn the TV or radio antenna until the interference stops.
- Move the computer to one side or the other of the TV or radio.
- Move the computer farther away from the TV or radio.
- Plug the computer into an outlet that is on a different circuit than the TV or radio. (That is, make certain the computer and the radio or television set are on circuits controlled by different circuit breakers or fuses.)
- Consider installing a rooftop television antenna with coaxial cable lead-in between the antenna and TV.

If necessary, you should consult your dealer or an experienced radio/television technician for additional suggestions. You may find helpful the following booklet, prepared by the Federal Communications Commission:

"How to Identify and Resolve Radio-TV Interference Problems"

This booklet is available from the U.S. Government Printing Office, Washington, DC 20402, stock number 004-000-00345-4.

<u>_</u>1

Table of Contents

Monitor ROM Listings

3 Monitor Firmware Listing

3

- 19 Monitor Symbol Table, Sorted by Symbol
- 21 Monitor Symbol Table, Sorted by Address
- 23 80-Column Firmware Listing
- 51 80-Column Symbol Table, Sorted by Symbol
- 53 80-Column Symbol Table, Sorted by Address

0000:			2	******	********	***	*****	4.4	
0000:			3		*******	****	*********		
0000:				* APPLE	11				
0000:				* MONITO					
0000:			6						
0000:			7	* COPYRI	GHT 1978	BY			
0000:				* APPLE	COMPUTER	, IN	4C.		
0000:				*					
0000:				* ALL RI	GHTS RESI	ERVE	D		
0000:				*					
0000:				* STEVE	WUZNIAK				
0000:			13	*	********		********	**	
0000:			15	*******					
0000:			16	* MODIFI	ED NOV 1	978			
0000				* BY JOH					
0000:			18						
0000:				* MODIFI					
0000:			20	* BY RIC	K AURICCI	HIO			
0000:			21		RYAN STE				
0000:			22		APPLE2E	BOC	COLS		
0000:			23	*	-	-	(0040001 (
0000:					S MARKED	BA	'RRA0981 '		
0000:		0001	25	* APPLE2E	EQU	1		COND	ASSM/RRA0981
0000:		0001	26		EGO	1		COND	MODIT/ KKHO/DI
0000					*****	****	*********	**	
	NEXT	DB. FCT		NAME IS					
F800;	inc A i	F300			ORG		800		
F800:		0000	30		OBJ	\$2	2000		
F800:			31	******	****	***	*****	¥	
F800:		0000		LOCO	EQU	\$0	00		
F800:		0001		LOC1	EQU	\$(
F800:		0020		WNDLFT	EQU	\$2			
F800:		0021		WNDWDTH	EQU	\$2			
F800:		0022		WNDTOP	EQU	\$2			
F800:		0023		WNDBTM	EQU	\$6			
F800: F800:		0024			EQU	\$2			
F800:		0025		GBASL	EQU	\$2			
F800		0025		GBASH	EQU	\$2			
F800:		0028		BASL	EQU	\$2			
F800		0029		BASH	EQU	\$2	29		
F800:		002A		BAS2L	EQU	\$2	2A		
F800:		002B	45	BAS2H	EQU	\$2			
F800:		0020	46	H2	EQU	\$2			
F800:		0020		LMNEM	EQU	\$2			
F800:		002D		V2	EQU	\$2			
F800:		0020		RMNEM	EQU		2D		
F800:		002E		MASK	EQU	\$2	26		
F800:		002E	51	CHKSUM FORMAT	EQU		2E		
F800:		002E		LASTIN	EQU	\$2			
F800		002F	54	LENGTH	EQU	\$2			
F800:		002F	55	SIGN	EQU	\$2	2F		
F800:		0030		COLOR	EQU	\$3	30		
F800:		0031		MODE	EQU	\$	31		
F800:		0032		INVFLG	EQU		32		
F800:		0033		PROMPT	EQU		33		
F800:		0034		YSAV	EQU		34		
F800:		0035		YSAV1	EQU		35		
F800:		0035		CSWL	EQU		36		
F800: F800:		0037		KSWL	EQU		38		
F800:		0038		KSWH	EQU		39		
F800:		0034		PCL	EQU		3A A		
F800:		003B		PCH	EQU		3B		
F800:		0030		AIL	EQU		30		
F800:		OOBD		AIH	EQU	\$:	3D		
F800:		003E.		A2L	EQU		3E.		
F800:		003F	71	A2H	EQU		3F		
F800:		0040		AGL	EQU		40		
F800:		0041		A3H	EQU		41 42		
F800:		0042		A4L A4H	EQU		42 43		
F800:		0043		A4H A5L	EGU		43		
1 000:		0044	/6		EGU	-0-			

F800: 0045	77 A5H	5011		
F800: 0045		EQU	\$45 \$45	/ NOTE OVERLAP WITH A5H!
F800: 0046	79 XREG	EQU	\$46	
F800: 0047		EQU	\$47	
F800: 0048 F800: 0049		EQU	\$48 \$49	
F800: 0047		EQU	\$49 \$4E	
F800: 004F	84 RNDH	EQU	\$4F	
FB00: 0095		EQU	\$95	
F800: 0200 F800: 03F0		EQU	\$0200	
F800: 03F2		EQU	\$3F0 \$3F2	; NEW VECTOR FOR BRK ; VECTOR FOR WARM START
F800: 03F4		EQU	\$3F4	THIS MUST = EDR #\$A5 OF SOFTEV+1
F800: 03F5		EQU	\$3F5	; APPLESOFT & EXIT VECTOR
F800: 03F8 F800: 03F8		EQU	\$03F8 \$03FB	
FB00: 03FF		EQU	\$3FE	
F800: 0400	94 LINE1	EQU	\$400	
F800: 07F8	95 MSLOT	EQU	\$07F8	
F800: C000 F800: C000	96 IDADR 97 KBD	EQU	\$C000	
F800: C010		EQU	\$C000 \$C010	
F800: C020		EQU	\$C020	
F800: C030		EQU	\$C030	
F800: C050 F800: C051	101 TXTCLR 102 TXTSET	EQU	\$C050 \$C051	
F800: C052		EQU	\$C052	
F800: C053	104 MIXSET	EQU	\$C053	
F800: C054	105 LOWSCR	EQU	\$C054	
F800: C055 F800: C056	106 HISCR 107 LORES	EQU	\$C055 \$C056	
FB00: C057	108 HIRES	EQU	\$C056 \$C057	
F800: C058		EQU	\$C058	
F800: C059	110 CLRANO	EQU	\$C059	
F800: C05A F800: C05B	111 SETAN1 112 CLRAN1	EQU	\$C05A \$C05B	
F800: C05C	113 SETAN2	EQU	\$C056 \$C05C	
F800: C05D	114 CLRAN2	EQU	\$C05D	
F800: C05E	115 SETAN3	EQU	\$C05E	
F800: C05F F800: C060	116 CLRAN3 117 TAPEIN	EQU	\$C05F \$C050	
F800: C060	118 PADDLO	EGU	\$C050 \$C064	
F800: C070	119 PTRIG	EQU	\$C070	
F800: CFFF	120 CLRROM	EQU	\$CFFF	
F800: E000 F800: E003	121 BASIC 122 BASIC2	EQU	\$E000	
F800: 4A	123 PLOT	LSR	\$E003 A	Y-CODRD/2
F801:08	124	PHP		SAVE LSB IN CARRY
F802:20 47 F8	125	JSR	GBASCALC	; CALC BASE ADR IN GHASL, H
F805:28 F806:A9 OF	126 127		#\$0F	RESTORE LSB FROM CARRY
F808: 90 02 F80C	128	BCC	RTMASK	; MASK \$OF IF EVEN
F80A: 69 E0	129	ADC	#\$E0	/MASK \$FO IF ODD
F80C:85 2E	130 RTMASK	STA	MASK	
F80E: B1 26 F810: 45 30	131 PLOT1 132	LDA EOR	(GBASL), Y COLOR	; DATA ; XOR COLOR
F812:25 2E	133	AND	MASK	AND MASK
F814:51 26	134	EOR	(GBASL), Y	XOR DATA
F816: 91 26	135	STA	(GBASL), Y	J TO DATA
F818:60 F819:20 00 F8	136 137 HLINE	RTS JSR	PLOT	PLOT SQUARE
F81C:C4 2C	138 HLINE1	CPY	H2	DONE?
F81E: B0 11 F831	139	BCS	RTS1	; YES, RETURN
F820:C8 F821:20 0E F8	140	INY JSR	PLOT 1	; YES, RETURN ; NO, INCR INDEX (XCOORD) ;PLOT NEXT SQUARE
F824: 90 F6 F81C	142	BCC	HLINE1	
F826:69 01	143 VLINEZ	ADC	#\$01	NEXT Y-COORD
F828: 48	144 VLINE	PHA		; SAVE ON STACK ; PLOT SQUARE
F829:20 00 F8 F82C:68	145 146	JSR PLA	PLOT) PLOT SQUARE
F82D:C5 2D	147	CMP	V2	DONE?
F82F: 90 F5 F826	148	BCC	VLINEZ	, NO, LOOP.
F831:60	149 RTS1 150 CLRSCR	RTS		
FB32: A0 2F FB34: D0 02 FB38	150 CLRSCR 151	LDY BNE	#\$2F CLRSC2	MAX Y, FULL SCRN CLR ;ALWAYS TAKEN
F836: A0 27	152 CLRTOP	LDY	#\$27	; MAX Y, TOP SCRN CLR
F838:84 2D	153 CLRSC2	STY	V2	STORE AS BOTTOM COORD
F83A F83A: A0 27	154 155			FOR VLINE CALLS
F83C: A9 00	155 CLRSC3	LDY	#\$27 #\$00	FIGHTMOST X-COORD (COLUMN) TOP COORD FOR VLINE CALLS
F83E:85 30	157	STA	COLOR	CLEAR COLOR (BLACK)
F840:20 28 F8	158	JSR	VLINE	DRAW VLINE
F843:88 F844:10 F6 F830	159 160	DEY BPL	CLRSC3	NEXT LEFTMOST X-COORD
F'846: 60	161	RTS	00000	FLOOP UNLIE DUNG.
F847:48	162 GBASCALC	PHA		FOR INPUT OODEFGH
F848: 4A	163	LSR	A	
F849:29 03 F848:09 04	164 165	AND ORA	#\$03 #\$04	GENERATE GBASH=000001FG
F84D:85 27	166	STA	GBASH	SERVICE OF CONTROLS
F84F: 68	167	PLA		AND GBASL=HDEDECOO
F850:29 18	168	AND	#\$18	

F852:90 02 F856	169	BCC GBCALC	
F854: 69 7F	170	ADC #\$7F	
F856:85 26	171 GBCALC	STA GBASL	
F858: 0A F859: 0A	172 173	ASL A	
F854:05 26	173	ORA GBASL	
F85C: 85 26	175	STA GBASL	
F85E: 60	176	RTS	
F85F: A5 30	177 NXTCOL	LDA COLOR	INCREMENT COLOR BY 3
F861:18	178	CLC	
F862:69 03	179	ADC #\$03	
F864: 29 OF	180 SETCOL	AND #\$OF	SETS COLOR=17*A MOD 16
F866:85 30	181	STA COLOR	
F868: 0A	182	ASL A	; BOTH HALF BYTES OF COLOR EQUAL
F869: 0A	183	ASL A	
FB6A: OA	184	ASL A	
F868: 0A	185	ASL A	
F86C:05 30 F86E:85 30	186 187	ORA COLOR STA COLOR	
F870: 60	188	RTS	
F871: 4A	189 SCRN	LSR A	READ SCREEN Y-COORD/2
F872:08	190	PHP	SAVE LSB (CARRY)
F873:20 47 F8	191	JSR GBASCALC	CALC BASE ADDRESS
F876: B1 26	192	LDA (GBASL), Y	GET BYTE
F878: 28	193	PLP	RESTORE LSB FROM CARRY
F879:90 04 F87F	194 SCRN2	BCC RTMSKZ	; IF EVEN, USE LO H
F878:4A	195	LSR A	
F87C: 4A	196	LSR A	
F87D: 4A	197	LSR A	SHIFT HIGH HALF BYTE DOWN
F87E: 4A	198	LSR A	
F87F: 29 OF	199 RTMSKZ	AND #\$OF	; MASK 4-BITS
F881:60 F882:66 36	200	RTS LDX PCL	BRINT POL H
F882: A6 3A F884: A4 3B	201 INSDS1 202	LDX PCL LDY PCH	PRINT PCL, H
F884:A4 38 F886:20 96 FD	202	JSR PRYX2	
F889:20 48 F9	204	JSR PRBLNK	FOLLOWED BY A BLANK
F88C: A1 3A	205 INSDS2	LDA (PGL, X)	GET OPCODE
F88E: A8	206	TAY	, oel 61 000E
F88F: 4A	207	LSR A	EVEN/ODD TEST
F890:90 09 F89B	208	BCC IEVEN	
F872: 6A	209	RDR A	BIT 1 TEST
F893: B0 10 F8A5	210	BCS ERR	XXXXXX11 INVALID OP
F895:C9 A2	211	CMP #\$A2	
F897: F0 0C F8A5	212	BEQ ERR	; OPCODE \$89 INVALID
F899:29 87	213	AND #\$87	MASK BITS
F89B: 4A	214 IEVEN	LSR A	LSB INTO CARRY FOR L/R TEST
F89C: AA	215	TAX	
F89D: BD 62 F9 F8A0: 20 79 F8	216	LDA FMT1, X	GET FORMAT INDEX BYTE
	217	JSR SCRN2	RIL H-BYTE ON CARRY
F8A3: D0 04 F8A9	218	BNE GETFMT	
F8A3: D0 04 F8A9 F8A5: A0 80	218 219 ERR	BNE GETFMT LDY #\$80	SUBSTITUTE \$80 FOR INVALID OPS
F8A3: D0 04 F8A9 F8A5: A0 80 F8A7: A9 00	218 219 ERR 220	BNE GETFMT LDY #\$80 LDA #\$00	SUBSTITUTE \$80 FOR INVALID OPS SET PRINT FORMAT INDEX TO O
F8A3: D0 04 F8A9 F8A5: A0 80 F8A7: A9 00 F8A9: AA	218 219 ERR 220 221 GETFMT	BNE GETFMT LDY #\$80 LDA #\$00 TAX	SET PRINT FORMAT INDEX TO O
F8A3:D0 04 F8A9 F8A5:A0 80 F8A7:A9 00 F8A9:AA F8AA:BD A6 F9	218 219 ERR 220 221 GETFMT 222	BNE GETFMT LDY #\$80 LDA #\$00 TAX LDA FMT2,X	SET PRINT FORMAT INDEX TO O
F8A3: D0 04 F8A9 F8A5: A0 80 F8A7: A9 00 F8A9: AA F8AA: BD A6 F9 F8AD: 85 2E	218 219 ERR 220 221 GETFMT	BNE GETFMT LDY #\$80 LDA #\$00 TAX LDA FMT2,X STA FDRMAT	SET PRINT FORMAT INDEX TO O INDEX INTO PRINT FORMAT TABLE SAVE FOR ADR FIELD FORMATTING
F8A3:D0 04 F8A9 F8A5:A0 80 F8A7:A9 00 F8A9:AA F8AA:BD A6 F9	218 219 ERR 220 221 GETFMT 222 223 223	BNE GETFMT LDY #\$80 LDA #\$00 TAX LDA FMT2,X STA FORMAT AND #\$03	SET PRINT FORMAT INDEX TO O INDEX INTO PRINT FORMAT TABLE AVE FOR ADR FIELD FORMATTING MASK FOR 2-BIT LENGTH
FBA3: D0 04 FBA7 FBA5: A0 80 FBA7: A7 00 FBA9: AA FBA4: BD A6 F9 FBAD: 85 2E FBAF: 27 03	218 219 ERR 220 221 GETFMT 222 223 223	BNE GETFMT LDY #\$80 LDA #\$00 TAX LDA FMT2,X STA FORMAT AND #\$03	SET PRINT FORMAT INDEX TO O INDEX INTO PRINT FORMAT TABLE AVE FOR ADR FIELD FORMATTING MASK FOR 2-BIT LENGTH
F8A3:D0 04 F8A9 F8A5:A0 80 F8A7:A9 00 F8A7:A9 00 F8A4:BD A6 F9 F8A1:BD A6 F9 F8A1:B5 2E F8AF:29 03 F8B1: F8B1:B5 2F F8B3:98	218 219 ERR 220 ERR 221 GETFMT 222 223 224 225 ; (0=1 BYTE 226 227	BNE GETFMT LDY #\$80 LDA #\$00 TAX LDA LDA FMT2, X STA FORMAT AND #\$03 E, 1=2 BYTE, 2=3 STA LENGTH TYA TYA	;SET PRINT FORMAT INDEX TO O ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FIELD FORMATTING ;MASK FOR 2-BIT LENGTH TE) ;OPCODE
F8A3:D0 04 F8A9 F8A5:A0 80 F8A7:A9 00 F8A7:A9 00 F8A9:AA F8A1:BD A6 F9 F8A2:A0 F8B1:E5 2E F8B1:E5 2F F8B1:E5 2F F8B3:98 F8B4:29 8F	218 219 ERR 220 221 GETFMT 222 223 224 225 ; (0=1 BYTE 226 227 228	BNE GETFMT LDY #\$80 LDA #\$80 LDA #\$00 TAX	;SET PRINT FORMAT INDEX TO 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FIELD FORMATTING ;MASK FOR 2-BIT LENGTH TE) ;OPCODE ;MASK FOR 1XXX1010 TEST
F8A3:D0 04 F8A9 F8A5:A0 80 F8A7:A9 00 F8A7:A9 00 F8A2:BD A6 F9 F8A1:BD A6 F9 F8A1:B5 2E F8A1:29 03 F8B1: F8B1:B5 2F F8B3:78 F8B4:29 8F F8B4:29 8F	218 219 ERR 220 221 GETFMT 222 223 224 225 ; (O=1 BYTE 226 227 228 227 228	BNE GETFMT LDY #\$80 LDA #\$00 TAX LDA FMT2, X STA FORMAT AND #\$03 STA LENGTH TYA LA AND #\$8F TAX *	;SET PRINT FORMAT INDEX TO 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FIELD FORMATTING ;MASK FOR 2-BIT LENGTH TE) ;OPCODE ;MASK FOR 1XXX1010 TEST ; SAVE IT
FBA3: D0 04 FBA9 FBA5: A0 80 FBA7: A7 00 FBA7: A7 00 FBA7: A7 00 FBA7: A7 00 FBA7: A7 00 FBA7: A7 00 FBA7: A7 00 FBA7: B5 26 FBA7: B5 26 FBA1: B5 26 FBB1: FBB1: B5 27 FBB3: 78 FBB3: 78 FBB4: 27 87 FBB4: 27 87 FBB5: AA FB7: 78	218 219 ERR 220 221 GETFMT 222 223 224 225 ; (0=1 BYTE 226 227 228 229 230	BNE GETFMT LDY #800 LDA #800 TAX STA STA FORMAT AND #603 J.1 = BYTE 2=3 BYT STA LENOTH TYA AND TAX #58F TAX TYA	;SET PRINT FORMAT INDEX TO 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FIELD FORMATTING ;MASK FOR 2-BIT LENGTH TE) ;OPCODE ;MASK FOR 1XXX1010 TEST
F8A3:D0 04 F8A9 F8A5:A0 80 F8A7:A9 00 F8A7:A9 00 F8A2:BD A6 F9 F8A6:BD A6 F9 F8A6:B5 2E F8A7:29 03 F8B1: F8B1:B5 2F F8B3:98 F8B4:29 BF F8B6:AA F8B7:98 F8B8:A0 03	218 219 ERR 220 GETFMT 222 223 223 224 225 (0=1 BYTE 226 227 228 229 229 230 231	BNE GETFMT LDA #\$90 TAX LDA LDA #\$00 TAX STA FORMAT AND \$7A FORMAT STA LENGTH TYA AND TAX TAX TYA LANG TAX LOY TYA LOY	;SET PRINT FORMAT INDEX TO 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FIELD FORMATTING ;MASK FOR 2-BIT LENGTH TE) ;OPCODE ;MASK FOR 1XXX1010 TEST ; SAVE IT
FBA3:D0 04 FBA9 FBA5:A0 80 FBA7:A0 80 FBA7:A7 00 FBA7:A7 00 FBA7:A5 20 FBA9:B5 22 FBA1:B5 22 FBA1:B5 22 FBB1:B5 25 FBB3:78 FBA2:B5 27 FBB3:78 FB44:27 FB FB42:A7 FB FB42:B5 FB FB42:B5 FB FB45:AA FB FB54:E0 FB FB8:A0 03 FB8:A0 A3	218 219 ERR 220 221 GETFMT 222 223 224 225 ; (0=1 BYTE 226 227 228 228 229 229 229 229 230 231	DNE GETFMT LDY #\$80 LDA #\$00 TAX #\$03 LDA FORMAT AND #\$03 J.1 = BYTE 2=3 BYT STA EONTH TYA AND AND #\$BF TAX TYA LDY #\$03 CPX #\$8A	;SET PRINT FORMAT INDEX TO 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FIELD FORMATTING ;MASK FOR 2-BIT LENGTH TE) ;OPCODE ;MASK FOR 1XXX1010 TEST ; SAVE IT
F8A3:D0 04 F8A9 F8A3:A0 80 F8A7:A9 00 F8A7:A9 00 F8A9:A1 F8A2 F8A8:B0 A6 F9 F8A9 F8A8:B0 A6 F9 F8B1:B5 F8B1:B5 F8B1:B5 2F F8B3:98 F8B3:98 F8B4:27 F8 F8B4:27 F8 F8B4:27 F8 F8B4:20 F8 F8B4:20 F8 F8B5:A0 03 F888:A0 03 F888:A0 F88:C5 F8 F885:A0 03 F886:C5 08 F88C9 F8 F8	218 219 ERR 220 GETFMT 222 GETFMT 222 223 223 224 225 (0=1 BYTE 226 227 228 229 230 231 232 233	BNE CETFMT LDY #\$80 LDA #\$50 TAX STA FORMAT STA STA FORMAT STA FORMAT STA FORMAT STA LENGTH TYA AND TYA LOY CPX #\$63 CPX #\$80 EEG MNDDX3	;SET PRINT FORMAT INDEX TO 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FIELD FORMATTING ;MASK FOR 2-BIT LENGTH TE) ;OPCODE ;MASK FOR 1XXX1010 TEST ; SAVE IT
FBA3:D0 04 FBA9 FBA5:A0 80 FBA7:A9 00 FBA7:A7 00 FBA7:A9 00 FBA7:A5 D0 A6 F9 FBA5:B5 2E FBA7:B5 2E FBA7:39 FBB1:B5 2F FBB3:378 FBB4:A27 9F FBB4:27 9F FBB4:A0 03 FBB7:78 FBB2:A0 03 FB26:AA FBB2:C6 08 FBC9 FBB2:C7 08 FBC9 FBB2:A0 13 FBC9 FBB2:A4 FBC9	218 219 ERR 220 221 GETFMT 222 223 224 225; (0=1 BYTE 226 227 228 228 229 228 229 229 229 230 231 232 232 232 232 233	DNE GETFMT LDY #\$80 LDA #\$90 TAX STA LDA FMT2,X STA FCMAIT STA FCMAIT STA FCMAIT TYA A AND #\$9F TAX TYA LDY #\$03 CPX #\$8A BEG MNIDX3 LSR A	;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FILLD FORMATTING ;MASK FOR 2-BIT LENGTH TEP ;OPCODE ;MASK FOR 1XXX1010 TEST ; SAVE IT ;DPCODE TO A AGAIN
F8A3:D0 04 F8A9 F8A5:A0 80 F8A7:A9 00 F8A7:A9 00 F8A9:A4 F8A7:A9 00 F8A7:A9 00 F8A9:A4 F8A7:A9 00 F8A7:A9 00 F8A7:A9 00 F8A7:A9 F8A7:A9 00 F8A7:A9 00 F8A7:A9 F8A7:A9 00 F8A7:A9 00 F8B1: F8B1:F8A7:29 03 F8B3:78 F8B3:78 F8B4:A4 F8B3:78 F8B4:A0 03 F8B4:A9 F8B5:40 03 F8B5:40 03 F8B4:C0 BA F8B5:40 BA F8B5:10 F8C9 F8C9 F8B5:40 03 F8C5 68 F8C9 <	218 219 ERR 220 221 GETFMT 222 223 224 225 (0=1 BYTE 226 227 228 229 228 229 230 231 232 233 234 MNNDX1 235	BNE GETFMT LDY #\$00 LDA #\$00 TAX FDRMAT STA FDRMAT STA FDRMAT STA FDRMAT STA FDRMAT STA LENOTH TYA LDY LDY #\$03 CPX #\$8A BEG MNNDX3 LSR A	;SET PRINT FORMAT INDEX TO 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FIELD FORMATTING ;MASK FOR 2-BIT LENGTH TE) ;OPCODE ;MASK FOR 1XXX1010 TEST ; SAVE IT
FBA3: D0 04 FBA9 FBA5: A0 80 FBA7: A9 00 FBA7: A7 00 FBA7: A4 90 FBA7: B5 2E FBA7: B5 2E FBA1: B5 2E FBA1: B5 2F FBB1: B5 2F FBB3: 98 FBB5: A0 FBB2: A0 03 FBB2: A0 FBB2: A0 FBB2: CF0 0B FBC9 FBS2: FA FBB2: CF0 0B FBC9 FBS2: FA FBB2: CF0 0B FBC9 FBC9 FBB2: CF0 CB FBC9 FBC9 FBC1: CF0	218 219 ERR 220 2219 ERR 220 223 224 225 ; (0=1 BYTE 225 227 227 228 229 229 229 230 231 232 232 232 233 MNNDX1 232 234 MNNDX1 233 4	DNE GETFMT LDP #\$80 LDA #\$00 TAX STA STA FCMAIT STA FCMAIT STA FCMAIT STA FCMAIT TYA STA AND #\$BF TAX TYA LDY #\$03 CDY #\$04 BEG MNDX3 LSR A BCS A	;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FILLD FORMATTING ;MASK FOR 2-BIT LENGTH FE) ;OPCODE ;MASK FOR 1XXX1010 TEST ; SAVE IT ;DPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE
F8A3:D0 04 F8A9 F8A5:A0 80 F8A7:A9 00 F8A7:A9 00 F8A9:A4 F8A7:A9 00 F8A7:A9 00 F8A9:A4 F8A7:A9 00 F8A7:A9 00 F8A7:A9 00 F8A7:A9 F8A7:A9 00 F8A7:A9 00 F8A7:A9 F8A7:A9 00 F8A7:A9 00 F8B1: F8B1:F8A7:29 03 F8B3:78 F8B3:78 F8B4:A4 F8B3:78 F8B4:A0 03 F8B4:A9 F8B5:40 03 F8B5:40 03 F8B4:C0 BA F8B5:40 BA F8B5:10 F8C9 F8C9 F8B5:40 03 F8C5 68 F8C9 <	218 219 ERR 220 221 GETFMT 222 223 224 225 (0=1 BYTE 226 227 228 229 228 229 230 231 232 233 234 MNNDX1 235	BNE GETFMT LDY #\$00 LDA #\$00 TAX FDRMAT STA FDRMAT STA FDRMAT STA FDRMAT STA FDRMAT STA LENOTH TYA LDY LDY #\$03 CPX #\$8A BEG MNNDX3 LSR A	;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FILLD FORMATTING ;MASK FOR 2-BIT LENGTH [] ;OPCODE ;MASK FOR 1XXX1010 TEST ; SAVE IT ;DAVE IT ;DPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE
FBA3: D0 04 FBA7 FBA5: A0 80 FBA7: A9 00 FBA7: A7 00 FBA7: A4 80 FBA7: BD A6 F7 FBA7: B5 2E FBA7: B5 2E FBA7: B5 2E FBB1: B5 2F FBB1: B5 2F FBB2: B5 2F FBB2: B5 2F FBB2: B5 2F FBB2: A0 03 FBB2: B4 03 FBB2: B5 C0 03 FBB2: B4 PB2: B5 4A FBB5: CF 0B FBC9 FBC9 FB2: B4A FB3F: 90 08 FBC9 FBB5: CF 00 FBC9 FB2: C14A FB2: C14A<	218 219 ERR 220 221 GETFMT 222 223 224 225 (0=1 BYTE 226 227 228 229 228 229 230 231 232 233 234 MNNDX1 235 235 236 237 MNNDX2	BNE GETFMT LDY #\$00 LDA #\$00 TAX FORMAT STA FORMAT STA FORMAT STA FORMAT STA FORMAT STA LENOTH TVA AND LDY #\$03 CPX #\$8A BEG MNNDX3 LSR A LSR A	;SET PRINT FORMAT INDEX TO 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FIELD FORMATTING ;MASK FOR 2-BIT LENGTH TE> ;DPCODE ;MASK FOR 1XXX1010 TEST ;SAVE IT ;DPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) 1XXX1010 => 00101XXX
FBA3: D0 04 FBA9 FBA5: A0 80 FBA7: A9 00 FBA7: A9 00 FBA7: A9 52 FBA7: B5 22 FBA5: B5 22 FBB1: B5 27 FBB1: B5 27 FBB2: A0 03 FBB2: A0 03 FBB2: A0 03 FBB2: CFO 08 FBB2: CFO 08 FB2: CFO 9 FB2: A0 68 FB3: CFO 08 FB2: A0 68 FB3: CFO 9 FB3: CFO 9 FB5: CFO 90 FB5: CFO 70 FB5: CF	218 219 ERR 220 221 GEFMT 222 223 225 ; (0=1 BYTE 226 227 227 229 229 229 229 232 233 232 233 232 234 MNNDX1 235 234 237 236 237 MNNDX2 238 239 239 239 239 240	DNE GETFMT LDP #\$80 LDA #\$00 TAX STA LDA FMT2,X STA FORMAT AND #\$03 c, 1=2 BYTE, 2=3 BYTA LENGTH TYA AND LDY #\$03 CDY #\$603 LDY #\$03 CDY #\$03 LDY #\$03 LDY #\$03 LDY #\$03 LSR A BEG MNNDX3 LSR A LSR A DEX MNNDX2	;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;GAVE FOR ADR FILLD FORMATTING ;MASK FOR 2-BIT LENGTH ; ;MASK FOR 1XXX1010 TEST ; MASK FOR 1XXX1010 TEST ; SAVE IT ;DPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) 1XXX1010 => 00101XXX ; 2) XXXYYY01 => 00110XXX ; 4) XXXYY100 => 00100XXX
FBA3: D0 04 FBA7 FBA5: A0 80 FBA7: A7 00 FBA7: A4 90 FBA7: A5 90 FBA7: A4 90 FBA7: A5 90 FB1: 95 FB1: 81 FB32: 78 FB86: A0 FB8: A0 03 FB8: C6 84 FB8: C8 A FB8: C8 A FB8: C8 FB67: 78 FB8: C8 FB67: 98 FB8: A0 03 F88: F0 08 F88: F0 86 F88: F0 86 F88: F0 9 F88: F0 9 F86: S8 F8C9 F86: S8 F8C2 F86: S8 F8C2 F86: C8 F8C2	218 219 ERR 220 221 GETFMT 222 223 224 225 (0=1 BYTE 226 227 228 229 229 230 231 232 232 232 233 234 MNNDX1 235 236 237 237 239 239 239 240	BNE GETFMT LDY #\$80 LDA #\$500 TAX JDA FMT2,X STA FORMAT AND #\$03 ;, 1=2 BYTE, 2=3 STA LENOTH TYA AND LDY #\$63 CPX #\$64 BEG MNNDX3 LSR A CRA #\$20 DEY BNNDX2 INY NNNDX2	<pre>;SET PRINT FORMAT INDEX TO 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FIELD FORMATTING ;MASK FOR 2-BIT LENGTH TE ;OPCODE ;MASK FOR 1XXX1010 TEST ;SAVE IT ;OPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) 1XXX1010 => O0110XXX ; 2) XXXYYY10 => O0110XXX ; 3) XXXYYY10 => O0110XXX</pre>
FBA3: D0 04 FBA7 FBA5: A0 80 FBA7: A9 00 FBA7: A9 00 FBA7: A9 00 FBA7: A9 52 FBA7: A9 00 FBA7: B1 B5 2E FBA7: A9 FBB1: B5 2E FBA7: A9 03 FBB1: B5 2F FBB2: A9 FB58: A0 03 FBB2: A9 GA 03 FB82: A9 FB57: 90 68 FBC9 FBB5: A0 03 FB52: 4A FB57: 90 68 FBC9 FB61: 4A FB51: A1 GA 03 FB62: 4A FB62: 4A FB62: 4A FB62: A0 G3 FB62: 4A FB62: 4A FB62: 4A FB62: 4A FB62: A0 FB60: DF A FB62: B8 FB62: B8 FB62: B8 FB62: B8	218 219 ERR 220 221 GETFMT 222 223 224 227 227 227 227 229 229 229 229 229 232 233 232 234 MNNDX1 235 234 235 234 237 MNNDX2 238 239 239 239 239 240 241 242 MNNDX3 242 244 MNNDX3 344 MNNDX3 MNND	DNE GETFMT LDP #\$80 LDA #\$00 TAX STA LDA FMT2,X STA FORMAT AND #\$03 C, 1=2 BYTE, 2=3 BYTA AND AND #\$807 TAX TYA LDY #\$803 CDY #\$804 BEG MINNDX3 LSR A LSR A LSR A LSR A BEQ MINDX3 LSR A DEY BINE BINE MINDX2 INY A	;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;GAVE FOR ADR FILLD FORMATTING ;MASK FOR 2-BIT LENGTH ; ;MASK FOR 1XXX1010 TEST ; MASK FOR 1XXX1010 TEST ; SAVE IT ;DPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) 1XXX1010 => 00101XXX ; 2) XXXYYY01 => 00110XXX ; 4) XXXYY100 => 00100XXX
FBA3: D0 04 FBA9 FBA5: A0 80 FBA7: A7 00 FBA7: A7 00 FBA7: A4 90 FBA7: A4 90 FBA1: 52 FBA1: 52 FBB1: 52 FBB1: 52 FBB3: 78 FBB5: 40 FBB: 40 03 FBB: 80 03 FBB: 80 8 FBB: 90 8 FBB: 40 8 FBB: 40 8 FBB: 40 8 FBB: 50 08 FBC: 14A FBC9 FBC: 24A FBC2 FBC: 25 8 FBC: 20 FA FBC: 20	218 219 ERR 220 2219 ERR 220 222 223 224 225 226 227 228 229 229 229 230 231 232 232 232 232 232 232 232 233 MNNDX1 235 236 237 MNNDX1 237 238 239 240 239 240 241 242 MNNDX3 243 3	BNE GETFMT LDY #\$80 LDA #\$00 TAX TAX STA FDRMAT AND #\$02 TAX STA STA LENOTH TYA AND AND #\$03 CPX #\$8A BEG MINDX3 LSR A ORA #\$20 DEY BND BNNDX2 INY DEY BNE BNE <minndx1< td=""></minndx1<>	;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;GAVE FOR ADR FILLD FORMATTING ;MASK FOR 2-BIT LENGTH ; ;MASK FOR 1XXX1010 TEST ; MASK FOR 1XXX1010 TEST ; SAVE IT ;DPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) 1XXX1010 => 00101XXX ; 2) XXXYYY01 => 00110XXX ; 4) XXXYY100 => 00100XXX
FBA3: D0 04 FBA9 FBA5: A0 80 FBA7: A9 00 FBA7: A9 00 FBA7: A9 52 FBA7: A9 52 FBA7: B5 22 FBB1: B5 22 FBB1: B5 27 FBB2: 98 68 FBB4: 29 86 FBB4: 29 87 FBB4: 29 87 FBB4: 29 86 FBB4: 29 87 FBB5: A0 03 FB85: A0 03 FB85: A0 68 FB85: 70 86 FB85: 70 86 FB85: 70 90 FB70: 14A FB62: 4A FB62: 40 FA FB62: 00 FA FB62: 88 FB62: 60 FB62: 60 FB62: 60	218 219 ERR 220 221 GEFMT 222 223 224 227 227 227 227 227 227 229 223 233 232 233 233 234 MNNDX1 235 234 234 237 MNNDX2 238 239 229 229 229 223 233 234 237 MNNDX2 235 234 234 234 234 234 235 244 244 244 244 244 244 244 244 244 24	BNE GETFMT LDY #\$80 LDA #\$50 TAX FORMAT STA FORMAT STA FORMAT STA FORMAT STA FORMAT STA LENOTH TYA LDY LDY #\$63 CPX #\$8A BEG MNNDX3 LSR A BSC MNNDX3 LSR A BNE MNNDX2 INY BNE BNE MNNDX1 RTS KTS	;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;GAVE FOR ADR FILLD FORMATTING ;MASK FOR 2-BIT LENGTH ; ;MASK FOR 1XXX1010 TEST ; MASK FOR 1XXX1010 TEST ; SAVE IT ;DPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) 1XXX1010 => 00101XXX ; 2) XXXYYY01 => 00110XXX ; 4) XXXYY100 => 00100XXX
FBA3: D0 04 FBA9 FBA5: A0 80 FBA7: A7 00 FBB1: 52 FBB1: 52 FBB1: 52 FBB3: 70 FB FBB: 40 03 FBB: 40 03 FBB: 50 08 FBB: 40 03 FBB: 40 03 FBB: 40 03 FBB: 40 03 FBB: 70 08 FBC: 40 70 FBC: 50 70 FBC: 60	218 219 ERR 220 219 ERR 220 221 GETFMT 222 223 224 225 226 227 228 229 229 229 230 231 232 232 232 232 232 232 233 MNNDX1 235 236 237 MNNDX2 239 240 239 240 241 242 MNNDX3 243 244 244 245	BNE GETFMT LDY #\$80 LDA #\$00 TAX STA LDA FMT2.X STA FDRMAT AND #\$02 STA LENTH TYA AND AND #\$03 CPX #\$8A BEG MINDX3 LSR A BCC MINDX3 LSR A DEY BNE BNE MINDX2 INY DEY BNE MINDX1 RTS DFB DFB #FF, #FF, #FF, #FF, #FF	<pre>;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FILLD FORMATTING ;MASK FOR 2-BIT LENGTH TE ;OPCODE ;MASK FOR 1XXX1010 TEST ;SAVE IT ;DPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) IXX1010 => 00101XXX ; 2) XXXYY101 => 00110XXX ; 4) XXXYY100 => 00100XXX ; 5) XXXXX000 => 0000XXXXX</pre>
FBA3: D0 04 FBA9 FBA5: A0 80 FBA7: A9 00 FBA7: A9 52 FBA7: A9 52 FBA8: B0 A6 FBB1: B5 2F FBB4: 29 8F FBB4: 29 8F FBB4: 29 8F FBB4: 29 8F FBB4: 20 8A FBB5: A0 03 FB85: A0 03 FB85: 40 8 FB85: 40 8 FB85: 40 8 FB85: 40 8 FB61: 4A FBC2: 4A FBC2: 4A FBC3: B8 FBC4: 00 FA FBC2 FBC6: 00 FA FBC FBC0: 00 FF FF FBC0: 00 F2 FBB <td< td=""><td>218 219 ERR 220 221 GEFMT 222 223 224 227 227 227 227 227 227 229 229 229 229</td><td>BNE GETFMT LDY #\$80 LDA #\$90 TAX FDRMAT STA FDRMAT STA FDRMAT STA FDRMAT STA FDRMAT TAX STA LBOTH TAX LDY #\$03 STA LENOTH TVA LDY LDY #\$03 CPX #\$8A BEG MNNDX3 LSR A BRC MNNDX3 LSR A DFEY BNE MNNDX2 INY SR MNNDX1 RTS DFS DFS SF, \$FF, \$FF, \$FF</td><td><pre>;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FILLD FORMAT TABLE ;MASK FOR 2-BIT LENGTH re) ;OPCODE ;MASK FOR 1XXX1010 TEST ;SAVE IT ;OPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) 1XXX1010 => 00101XXX ; 2) XXXYYY01 => 00101XXX ; 4) XXXYYY01 => 00101XXX ; 5) XXXXYY00 => 0010XXX ; 5) XXXXYY00 => 0010XXX ; 5) XXXXYY00 => 0010XXX ; 5) XXXXYY00 => 0010XXX ; 5) XXXXY00 => 0010XXX ; 5) XXXY00 => 0010XXX ; 5) XXXXY00 => 0010XXX ; 5) XXXXY00 => 0010XXX ; 5) XXXXY00 => 0010XXX ; 5) XXXXY00 => 0010XXX ; 5) XXXY00 => 0010XXX ; 5) XXXY00 => 0010XXX ; 5) XXXY00 => 0010XXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 0000XXXX ; 5) XXXY00 => 0000XXXX ; 5) XXXY00 => 0000XXX ; 5) XXXY00 => 0000XXXX ; 5) XXXY00 => 000</pre></td></td<>	218 219 ERR 220 221 GEFMT 222 223 224 227 227 227 227 227 227 229 229 229 229	BNE GETFMT LDY #\$80 LDA #\$90 TAX FDRMAT STA FDRMAT STA FDRMAT STA FDRMAT STA FDRMAT TAX STA LBOTH TAX LDY #\$03 STA LENOTH TVA LDY LDY #\$03 CPX #\$8A BEG MNNDX3 LSR A BRC MNNDX3 LSR A DFEY BNE MNNDX2 INY SR MNNDX1 RTS DFS DFS SF, \$FF, \$FF, \$FF	<pre>;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FILLD FORMAT TABLE ;MASK FOR 2-BIT LENGTH re) ;OPCODE ;MASK FOR 1XXX1010 TEST ;SAVE IT ;OPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) 1XXX1010 => 00101XXX ; 2) XXXYYY01 => 00101XXX ; 4) XXXYYY01 => 00101XXX ; 5) XXXXYY00 => 0010XXX ; 5) XXXXYY00 => 0010XXX ; 5) XXXXYY00 => 0010XXX ; 5) XXXXYY00 => 0010XXX ; 5) XXXXY00 => 0010XXX ; 5) XXXY00 => 0010XXX ; 5) XXXXY00 => 0010XXX ; 5) XXXXY00 => 0010XXX ; 5) XXXXY00 => 0010XXX ; 5) XXXXY00 => 0010XXX ; 5) XXXY00 => 0010XXX ; 5) XXXY00 => 0010XXX ; 5) XXXY00 => 0010XXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 0000XXXX ; 5) XXXY00 => 0000XXXX ; 5) XXXY00 => 0000XXX ; 5) XXXY00 => 0000XXXX ; 5) XXXY00 => 000</pre>
FBA3: D0 04 FBA9 FBA5: A0 80 FBA7: A7 00 FBA7: A7 00 FBA7: A7 00 FBA7: A7 00 FBA7: A7 00 FBA7: A7 00 FBA7: A7 00 FBA7: B1 52 FBA7: A7 03 FBB1: 52 FBA7: A7 70 FBB1: 52 FBA7: A7 70 FBB1: 52 FB53: 78 FB85: 60 FBB2: A7 FB FB FB7 FBB2: A0 03 FBC9 FB85: 79 FBB2: A0 03 FBC9 FBC1: 4A FBB2: A4 FBC2: 4A FBC2: 4A FBC2: 4A FBC2: 4A FBC2: 4A FBC2: 4A FBC2: FBC2 FBC3: B8 FBC2: C8 FBC2: 5B FBC2: 5B FBC4: D0 F2 FBBE FBBE FB22: FB FBC2: C60 FBC2: FF FF FB30: 20 82 FB FBC0: C1 D0 F2 FBBE FBBE FB22 FB FBC0: C2 B2 <td>218 219 ERR 220 223 223 224 225 ; (0=1 BYTE 226 227 228 229 229 229 229 230 231 232 232 232 233 MNNDX1 233 234 MNNDX1 233 234 MNNDX1 233 234 244 243 244 244 245 245 245 245 245 245 245 245</td> <td>DNE GETFMT LDY #\$80 LDA #\$80 LDA #\$00 TAX STA STA FORMAT AND #\$03 J: 1=2 BYTE: 2=3 BYTE: 2=0 BYT TAX TAX TYA AND LDY #\$03 CPX #\$8A BEG MINDX3 LSR A BCC MINDX3 LSR A DEY BNE BNE MINDX2 INY DEY DEY BNE DEN MINDX1 RTS DFB DFB FFF, \$FF, \$FF JSR INSDS1</td> <td><pre>;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FILLD FORMATTING ;MASK FOR 2-BIT LENGTH TE ;OPCODE ;MASK FOR 1XXX1010 TEST ;SAVE IT ;DPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) IXX1010 => 00101XXX ; 2) XXXYY101 => 00110XXX ; 4) XXXYY100 => 00100XXX ; 5) XXXXX000 => 0000XXXXX</pre></td>	218 219 ERR 220 223 223 224 225 ; (0=1 BYTE 226 227 228 229 229 229 229 230 231 232 232 232 233 MNNDX1 233 234 MNNDX1 233 234 MNNDX1 233 234 244 243 244 244 245 245 245 245 245 245 245 245	DNE GETFMT LDY #\$80 LDA #\$80 LDA #\$00 TAX STA STA FORMAT AND #\$03 J: 1=2 BYTE: 2=3 BYTE: 2=0 BYT TAX TAX TYA AND LDY #\$03 CPX #\$8A BEG MINDX3 LSR A BCC MINDX3 LSR A DEY BNE BNE MINDX2 INY DEY DEY BNE DEN MINDX1 RTS DFB DFB FFF, \$FF, \$FF JSR INSDS1	<pre>;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FILLD FORMATTING ;MASK FOR 2-BIT LENGTH TE ;OPCODE ;MASK FOR 1XXX1010 TEST ;SAVE IT ;DPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) IXX1010 => 00101XXX ; 2) XXXYY101 => 00110XXX ; 4) XXXYY100 => 00100XXX ; 5) XXXXX000 => 0000XXXXX</pre>
FBA3: D0 04 FBA9 FBA5: A0 80 FBA7: A9 00 FBA7: A4 80 FBA7: A4 70 FBA7: A4 80 FBA7: A4 70 FBA7: A4 80 72 FBA7: A4 FBA7: A7 03 FBA7: A4 FBA7: A7 03 FB37: FB47: 27 FBB1: B5 27 FB38: 37 FBB5: A4 FB778 FB86: A4 FB85: A6 03 FBC7 FB85: C0 68 FBC7 FB85: C1 68 FBC7 FB85: C4 03 FBC7 FB62: C4 FBC7 FBC8 FBC2: C4 FBC2: C8 FBC2: C8 FBC2: C4 FBC2: B8 FBC2: B8 FBC2: C4 FBC8 FBE8 FBC0: FF FFBE8 FBC9: FF FBD0: 20 69 FF FBD0: 20 82 FF FBD1: 48 3A	218 219 ERR 220 221 GETFMT 222 223 224 225 (0=1 BYTE 226 227 228 229 229 230 231 232 233 234 MNNDX1 232 233 234 235 236 237 237 238 237 239 240 241 242 241 242 244 244 245 244 244 245 248 PRNTOP	BNE GETFMT LD4 #\$00 LDA #\$00 TAX STA FDRMAT STA STA FDRMAT STA FDRMAT STA FDRMAT STA FDRMAT TAX STA STA FDRMAT TAX STA LDY #\$03 CPX #\$8A BEG MNNDX3 LSR A DRA #\$20 DEY BNE BNM MNNDX2 INY BNE STA SF, \$FF, \$FF, \$FF JSR INSDS1 PHA LDA	<pre>;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FILLD FORMAT TABLE ;MASK FOR 2-BIT LENGTH re) ;OPCODE ;MASK FOR 1XXX1010 TEST ;SAVE IT ;OPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) 1XXX1010 => 00101XXX ; 2) XXXYYY01 => 00101XXX ; 4) XXXYYY01 => 00101XXX ; 5) XXXXYY00 => 0010XXX ; 5) XXXXYY00 => 0010XXX ; 5) XXXXYY00 => 0010XXX ; 5) XXXXYY00 => 0010XXX ; 5) XXXXY00 => 0010XXX ; 5) XXXY00 => 0010XXX ; 5) XXXXY00 => 0010XXX ; 5) XXXXY00 => 0010XXX ; 5) XXXXY00 => 0010XXX ; 5) XXXXY00 => 0010XXX ; 5) XXXY00 => 0010XXX ; 5) XXXY00 => 0010XXX ; 5) XXXY00 => 0010XXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 0000XXXX ; 5) XXXY00 => 0000XXXX ; 5) XXXY00 => 0000XXX ; 5) XXXY00 => 0000XXXX ; 5) XXXY00 => 000</pre>
FBA3: D0 04 FBA9 FBA5: A0 80 FBA7: A7 00 FBA7: A8 52 FBA1: B5 22 FBB1: FB1: B5 27 FBB3: 78 FB83: 78 FBB4: A7 78 FBB5: 70 08 FB85: 70 08 FB85: 70 90 FB85: 70 90 FB85: 90 68 FB62: 4A FB62: 4A FB62: 4A FB62: 4A FB62: 4A FB62: 4A FB62: 4A FB62: 5B FB62: 4A FB62: 4A FB62: 4A FB62: 5B FB70: 20 FB FB62: 4D FA FB62: 4D FA FB70: 20 FB FB70: 20 FB FB70: 20 FB FB70: 20 A FB70: 20 A FB7 FB FB7	218 219 ERR 220 219 ERR 220 223 223 225 ; (0=1 BYTE 226 227 228 229 229 229 231 232 232 232 232 232 232 234 MNNDX1 232 234 234 234 234 234 234 244 245 246 1NSTDSP 247 248 PRNTOP 249 PRNTOP 249 249	DNE GETFMT LDY #\$80 LDA #\$00 TAX STA STA FORMAT AND #\$03 J=2 BYTE: 2=3 BYT STA LENOTH TYA AND #\$03 CUPX CPX #\$03 CUPX LSR A BEG BEG MINIDX3 LSR LSR A BCC DEY #\$10 DEY DEK MINIDX3 LSR LSR A DEY DEY BNE MINIDX1 TTS UNDY STA DFB \$FF, \$FF, \$FF, \$FF JSR RINSDS1 PHA LDA (PCL), Y	<pre>;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FILLD FORMATTING ;MASK FOR 2-BIT LENGTH TE ;OPCODE ;MASK FOR 1XXX1010 TEST ;SAVE IT ;DPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) IXXI010 => 00101XXX ; 2) XXXYVY01 => 0011XXX ; 4) XXXYVY10 => 00110XXX ; 4) XXXYV10 => 0010XXX ; 5) XXXXX000 => 000XXXXX ;GEN FMT, LEN BYTES ;SAVE MNEMONIC TABLE INDEY</pre>
FBA3: D0 04 FBA7 FBA5: A0 80 FBA7: A7 00 FBA7: A7 00 FBA7: A4 90 FBA7: A4 90 FBA7: A7 00 FBA7: A4 90 FBA7: A7 00 FBA7: A4 90 FBA7: A7 00 FB1: 52 FB1: 52 FB1: 52 FB3: 90 FB7 FB8: 40 03 FB8: 40 03 FB8: 50 08 FB8: 60 64 FB8: 70 08 FB2: 40 78 FB2: 40	218 219 ERR 220 221 GETFMT 222 223 224 225 (0=1 BYTE 226 227 228 229 229 230 231 232 233 234 MNNDX1 235 234 237 238 237 238 239 239 239 239 240 237 239 240 241 243 243 244 245 245 244 PRNTOP 248 PRNTOP 248 PRNTOP 249 250	BNE GETFMT LDY #\$00 LDA #\$00 TAX STA FDRMAT STA FDRMAT STA LENOTH TYA AND #\$03 STA LENOTH TYA AND #\$8F TAX LDY #\$03 CPX #\$6A BE0 MINDX3 LSR A LSR A BE0 MINDX3 LSR A LSR A CPX #\$5A BE0 MINDX3 LSR A LSR A DRA #\$20 DEY BNE MINDX2 INY DEY BNE MINDX1 RTS DFB \$FF, \$FF, \$FF JSR INSDS1 PHA LDA (PCL), Y JSR PRBYTE LDX \$01	<pre>;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FILLD FORMAT TABLE ;MASK FOR 2-BIT LENGTH re) ;OPCODE ;MASK FOR 1XXX1010 TEST ;SAVE IT ;OPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) 1XXX1010 => 00101XXX ; 2) XXXYYY01 => 00101XXX ; 4) XXXYYY01 => 00101XXX ; 5) XXXXYY00 => 0010XXX ; 5) XXXXYY00 => 0010XXX ; 5) XXXXYY00 => 0010XXX ; 5) XXXXYY00 => 0010XXX ; 5) XXXXY00 => 0010XXX ; 5) XXXY00 => 0010XXX ; 5) XXXXY00 => 0010XXX ; 5) XXXXY00 => 0010XXX ; 5) XXXXY00 => 0010XXX ; 5) XXXXY00 => 0010XXX ; 5) XXXY00 => 0010XXX ; 5) XXXY00 => 0010XXX ; 5) XXXY00 => 0010XXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 000XXXX ; 5) XXXY00 => 0000XXXX ; 5) XXXY00 => 0000XXXX ; 5) XXXY00 => 0000XXX ; 5) XXXY00 => 0000XXXX ; 5) XXXY00 => 000</pre>
FBA3: D0 04 FBA9 FBA5: A0 80 FBA7: A7 00 FBA7: B5 22 FBA1: B5 22 FBB1: B5 27 FBB1: B5 27 FBB2: A7 87 FBB2: A7 87 FBB2: A0 03 FBB2: A0 03 FBB2: A0 03 FBB2: A4 FB26: C8 FBB2: A0 90 FB24: A4 FB24: A4 FB25: C9 08 FB27 FB26: C4 FB20: C9 00 FB26: C5 B8 FB26: C8 FB26: C8 FB26: C8 FB26: C8 FB26: C8 FB26: C8 FB26: C8 FB20: C100 FA FE FE FB20: C20 FE FF FB20: C20 FE FF FB20: C20 FE FF FB20: C20 AE FE <t< td=""><td>218 219 ERR 220 221 GETFMT 222 223 224 225 ; (0=1 BYTE 226 227 228 229 229 229 230 231 232 232 232 232 232 232 232 232 234 MNNDX1 232 234 234 234 234 234 234 234 244 245 246 1NSTDSP 246 1NSTDSP 247 248 PRNTOP 248 PRNTOP 247 251 PRNTBL</td><td>DNE GETFMT LDY 4*80 LDA 4*80 LDA 4*80 LDA FMT2,X STA FORMAT AND 4*03 STA LENOTH TYA AND 4*80 TAX TYA LDY 4*03 CPX 4*80 BEG MINIDX3 LSR A BEG MINIDX3 LSR A BCC MINIDX3 LSR A JSF FF, \$FF, \$FF JSR INSDS1 FHA LDA (PCL), Y JSR PRUTE LDX \$\$01 JSR PRL2</td><td><pre>;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FILL FORMATTING ;MASK FOR 2-BIT LENGTH TEP ;OPCODE ;MASK FOR 1XXX1010 TEST ;SAVE IT ;DPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) IXXI010 => 00101XXX ; 2) XXXYYV01 => 00111XXX ; 3) XXYYV10 => 00110XXX ; 4) XXYYV10 => 0010XXX ; 5) XXXXX000 => 000XXXXX ;GEN FMT; LEN BYTES ;SAVE MNEMONIC TABLE INDEX ;PRINT 2 BLANKS</pre></td></t<>	218 219 ERR 220 221 GETFMT 222 223 224 225 ; (0=1 BYTE 226 227 228 229 229 229 230 231 232 232 232 232 232 232 232 232 234 MNNDX1 232 234 234 234 234 234 234 234 244 245 246 1NSTDSP 246 1NSTDSP 247 248 PRNTOP 248 PRNTOP 247 251 PRNTBL	DNE GETFMT LDY 4*80 LDA 4*80 LDA 4*80 LDA FMT2,X STA FORMAT AND 4*03 STA LENOTH TYA AND 4*80 TAX TYA LDY 4*03 CPX 4*80 BEG MINIDX3 LSR A BEG MINIDX3 LSR A BCC MINIDX3 LSR A JSF FF, \$FF, \$FF JSR INSDS1 FHA LDA (PCL), Y JSR PRUTE LDX \$\$01 JSR PRL2	<pre>;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FILL FORMATTING ;MASK FOR 2-BIT LENGTH TEP ;OPCODE ;MASK FOR 1XXX1010 TEST ;SAVE IT ;DPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) IXXI010 => 00101XXX ; 2) XXXYYV01 => 00111XXX ; 3) XXYYV10 => 00110XXX ; 4) XXYYV10 => 0010XXX ; 5) XXXXX000 => 000XXXXX ;GEN FMT; LEN BYTES ;SAVE MNEMONIC TABLE INDEX ;PRINT 2 BLANKS</pre>
FBA3: D0 04 FBA9 FBA5: A0 80 FBA7: A7 00 FBA7: B5 22 FBA1: B5 22 FBB1: B5 27 FBB1: B5 27 FBB2: A7 87 FBB2: A7 87 FBB2: A0 03 FBB2: A0 03 FBB2: A0 03 FBB2: A4 FB26: C8 FBB2: A0 90 FB24: A4 FB24: A4 FB25: C9 08 FB27 FB26: C4 FB20: C9 00 FB26: C5 B8 FB26: C8 FB26: C8 FB26: C8 FB26: C8 FB26: C8 FB26: C8 FB26: C8 FB20: C100 FA FE FE FB20: C20 FE FF FB20: C20 FE FF FB20: C20 FE FF FB20: C20 AE FE <t< td=""><td>218 219 ERR 220 221 GETFMT 222 223 224 225 (0=1 BYTE 226 227 228 229 229 230 231 232 233 234 MNNDX1 235 234 237 238 237 238 239 239 239 239 240 237 239 240 241 243 243 244 245 245 244 PRNTOP 248 PRNTOP 248 PRNTOP 249 250</td><td>BNE GETFMT LDY #\$00 LDA #\$00 TAX STA FDRMAT STA FDRMAT STA LENOTH TYA AND #\$03 STA LENOTH TYA AND #\$8F TAX LDY #\$03 CPX #\$6A BE0 MINDX3 LSR A LSR A BE0 MINDX3 LSR A LSR A CPX #\$5A BE0 MINDX3 LSR A LSR A DRA #\$20 DEY BNE MINDX2 INY DEY BNE MINDX1 RTS DFB \$FF, \$FF, \$FF JSR INSDS1 PHA LDA (PCL), Y JSR PRBYTE LDX \$01</td><td><pre>;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FILLD FORMATTING ;MASK FOR 2-BIT LENGTH TE ;OPCODE ;MASK FOR 1XXX1010 TEST ;SAVE IT ;DPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) IXXI010 => 00101XXX ; 2) XXXYVY01 => 0011XXX ; 4) XXXYVY10 => 00110XXX ; 4) XXXYV10 => 0010XXX ; 5) XXXXX000 => 000XXXXX ;GEN FMT, LEN BYTES ;SAVE MNEMONIC TABLE INDEY</pre></td></t<>	218 219 ERR 220 221 GETFMT 222 223 224 225 (0=1 BYTE 226 227 228 229 229 230 231 232 233 234 MNNDX1 235 234 237 238 237 238 239 239 239 239 240 237 239 240 241 243 243 244 245 245 244 PRNTOP 248 PRNTOP 248 PRNTOP 249 250	BNE GETFMT LDY #\$00 LDA #\$00 TAX STA FDRMAT STA FDRMAT STA LENOTH TYA AND #\$03 STA LENOTH TYA AND #\$8F TAX LDY #\$03 CPX #\$6A BE0 MINDX3 LSR A LSR A BE0 MINDX3 LSR A LSR A CPX #\$5A BE0 MINDX3 LSR A LSR A DRA #\$20 DEY BNE MINDX2 INY DEY BNE MINDX1 RTS DFB \$FF, \$FF, \$FF JSR INSDS1 PHA LDA (PCL), Y JSR PRBYTE LDX \$01	<pre>;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FILLD FORMATTING ;MASK FOR 2-BIT LENGTH TE ;OPCODE ;MASK FOR 1XXX1010 TEST ;SAVE IT ;DPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) IXXI010 => 00101XXX ; 2) XXXYVY01 => 0011XXX ; 4) XXXYVY10 => 00110XXX ; 4) XXXYV10 => 0010XXX ; 5) XXXXX000 => 000XXXXX ;GEN FMT, LEN BYTES ;SAVE MNEMONIC TABLE INDEY</pre>
FBA3: D0 04 FBA7 FBA5: A0 80 FBA7: A7 00 FBA7: A7 00 FBA7: A4 90 FBA2: AA FBA2: AA FBA1: B5 2E FBA1: B5 2F FBB1: B5 2F FBB1: B5 2F FBB2: A7 8F FBB2: C0 8F FBB2: C0 8F FBB2: C0 8F FB2: C1: AA FBC2: FA FBC2: A4 FBC2: C4 FBC2: C4 FBC2: FA FBC4: C8 FBC2 FBC5: D0 FA FBC2 FBC6: D0 FA FBC2 FBC5: C8 FBC2 FBC3 FBC6: D0 FA FBC2 FBC6: D0 FA FBC2 FBC6: D0 FA FBC2 FBC6: B8 FBC4: B0 F2 FBD2: A4 F3 F3 FBD2: A4 F3 F3 FBD2: A2 A1 F3 FBD2: A2 C1 FBD2: A2	218 219 ERR 220 223 GETFMT 224 224 225 (0=1 BYTE 226 227 228 229 229 230 231 232 232 233 234 MNNDX1 232 235 234 237 239 230 237 239 240 237 239 240 237 239 240 241 243 243 244 245 244 245 244 245 246 247 247 248 247 249 249 249 249 249 241 242 241 243 243 244 243 244 245 244 245 244 245 244 245 244 245 252 253 254	BNE GETFMT LDY #\$80 LDA #\$00 TAX FORMAT STA FORMAT STA FORMAT STA FORMAT STA FORMAT TYA AND AND #\$03 TXA ENOTH TYA AND LDY #\$03 CPX #\$63 BEG MNNDX3 LSR A CC MNNDX3 LSR A DRE MNNDX1 BRE MNNDX1 PFB \$FF, \$FF, \$FF JSR INSDS1 PHA (PCL), Y JSR PRUTE LDX \$901 JSR PRL2 CPY LENCTH	<pre>;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FILLD FORMATTING ;MASK FOR 2-BIT LENGTH TE) ;DPCODE ;MASK FOR 1XXX1010 TEST ;SAVE IT ;DPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) 1XXX1010 => 00101XXX ; 2) XXXYY101 => 00110XXX ; 3) XXXYY101 => 00110XXX ; 4) XXXYY100 => 00100XXX ; 5) XXXXYY00 => 00100XXX ; 5) XXXXY000 => 0000XXXXX ;SAVE MNEMONIC TABLE INDEX ;SAVE MNEMONIC TABLE INDEX ;PRINT 2 BLANKS ;PRINT INST (1-3 BYTES) ;IN A 12 CHR FIELD</pre>
FBA3: D0 04 FBA9 FBA5: A0 80 FBA7: A7 00 FBA7: B5 22 FBA1: B5 22 FBB1: B5 27 FBB1: B5 27 FBB2: A7 87 FBB2: A7 90 FB2: A7 90 FB2: A7 90 FB2: A4 FB2: A7 FB2: A4 FB2: A7 FB2: C1 92 FB2: C2: A4 FB2: FB2: FB2 FB2: C2: A8 FB2: FB2 FB2: C2: A2 FF FB0: C2: A2 FF FB0: C3: D4 F7 FB0: C4: 25 FB2 FB2: C4: 27 FB2 <t< td=""><td>218 219 ERR 220 221 GETFMT 222 223 224 225 ; (0=1 BYTE 222 228 227 228 229 229 230 231 232 232 232 232 232 232 232 232 232</td><td>DNE GETFMT LDY #\$80 LDA #\$80 LDA #\$00 TAX STA LDA #\$00 TAX STA LDA #\$00 TAX STA STA LENTH TAX TAX TYA AND LDY #\$03 CDY #\$03 CDY #\$03 LSR A BCC MNNDX3 LSR A DCY #\$03 DEY DEY DEY DEY DFB \$FF, \$FF, \$FF JSR RNDX1 JSR PRBL2 CPY LNY LDX #\$03</td><td><pre>;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FILLD FORMATTING ;MASK FOR ADR FILLD FORMATTING ;MASK FOR 1XXX1010 TEST ; SAVE IT ;DPCODE ;DPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) 1XXX1010 => 00101XXX ; 2) XXXYY101 => 00110XXX ; 3) XXXYY101 => 00110XXX ; 4) XXXYY100 => 00100XXX ; 5) XXXXY000 => 00100XXX ; 5) XXXXX000 => 0000XXXXX ;GEN FMT, LEN BYTES ;SAVE MNEMONIC TABLE INDEX ;PRINT 2 BLANKS ;PRINT 1NST (1-3 BYTES)'</pre></td></t<>	218 219 ERR 220 221 GETFMT 222 223 224 225 ; (0=1 BYTE 222 228 227 228 229 229 230 231 232 232 232 232 232 232 232 232 232	DNE GETFMT LDY #\$80 LDA #\$80 LDA #\$00 TAX STA LDA #\$00 TAX STA LDA #\$00 TAX STA STA LENTH TAX TAX TYA AND LDY #\$03 CDY #\$03 CDY #\$03 LSR A BCC MNNDX3 LSR A DCY #\$03 DEY DEY DEY DEY DFB \$FF, \$FF, \$FF JSR RNDX1 JSR PRBL2 CPY LNY LDX #\$03	<pre>;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FILLD FORMATTING ;MASK FOR ADR FILLD FORMATTING ;MASK FOR 1XXX1010 TEST ; SAVE IT ;DPCODE ;DPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) 1XXX1010 => 00101XXX ; 2) XXXYY101 => 00110XXX ; 3) XXXYY101 => 00110XXX ; 4) XXXYY100 => 00100XXX ; 5) XXXXY000 => 00100XXX ; 5) XXXXX000 => 0000XXXXX ;GEN FMT, LEN BYTES ;SAVE MNEMONIC TABLE INDEX ;PRINT 2 BLANKS ;PRINT 1NST (1-3 BYTES)'</pre>
FBA3: D0 04 FBA9 FBA3: A0 80 FBA7: A7 00 FBA1: FBA7: A7 FBA1: 52 FBA1: 52 FBB1: 52 FBB1: 52 FBB1: 52 FBB2: 42 03 FBB2: 40 03 FBB2: 60 03 FBB2: 60 63 FBB2: 40 03 FBB2: 40 03 FB2: 4A FB22: 4A FB2: 50: 6B FB22 FB2: 6D: 7E FF FB0: 20 FE FB2: 4B 1A FB2: 4B 1A FB2: 20: 0A FD FB3: 4B 3A FB3: 20: 0A	218 219 ERR 220 221 GETFMT 222 223 224 225 226 227 228 229 229 229 230 231 232 232 233 234 MNNDX1 235 244 245 245 247 248 PRNTDP 249 249 249 249 241 244 244 245 249 249 249 241 241 241 241 242 241 242 241 242 242	BNE GETFMT LDY #\$80 LDA #\$00 TAX FORMAT STA FORMAT STA FORMAT STA FORMAT STA FORMAT STA FORMAT AND #\$03 TAX STA LDY #\$03 CPX #\$8A BEG MNNDX3 LSR A CCX #\$80 LSR A DEG MNNDX3 LSR A DEG MNNDX1 DFN \$FF, \$FF, \$FF JSR INSDS1 DFB \$PG1 JSR PRBL2 CPY LENOTH INY SCC JSR PRBL2 CPY ENOTH NY SC JSR \$PG3	<pre>;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FILLD FORMATTING ;MASK FOR 2-BIT LENGTH TE) ;DPCODE ;MASK FOR 1XXX1010 TEST ;SAVE IT ;DPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) 1XXX1010 => 00101XXX ; 2) XXXYY101 => 00110XXX ; 3) XXXYY101 => 00110XXX ; 4) XXXYY100 => 00100XXX ; 5) XXXXYY00 => 00100XXX ; 5) XXXXY000 => 0000XXXXX ;SAVE MNEMONIC TABLE INDEX ;SAVE MNEMONIC TABLE INDEX ;PRINT 2 BLANKS ;PRINT INST (1-3 BYTES) ;IN A 12 CHR FIELD</pre>
FBA3: D0 04 FBA9 FBA5: A0 80 FBA7: A7 00 FBA7: B1 B5 2E FBA7: 27 93 FBB1: B5 2F FB32: B5 2F FB32: B5 2F FB32: C7 03 FB34: E0 BA FB35: C7 03 FB35: C7 03 FB35: C7 08 F82: AA F82: AA F82: AA F82: AA F82: C7 08 F82: AA F82: AA F82: C7 08 F82: C4 A F82: C5 B8 F82: C6 C8 F82: C7 F88 F82: C8 F82: F82 F80: C9 F82 F80: C9 F82 F80: C8 F82 F80: C9 F82 F80: C0 82 F80: C0 82	218 219 219 219 219 221 221 221 222 223 224 227 227 228 227 228 229 229 230 231 232 233 234 MNNDX1 232 233 234 235 234 MNNDX2 238 239 234 239 234 235 244 245 245 245 245 245 245 245 245 24	DNE GETFMT LDY #\$80 LDA #\$80 LDA #\$00 TAX STA LDA #\$00 STA FCMAIT STA FCMAIT TYA STA ADD #\$03 CPX #\$03 CDY #\$03 CSR A BCC MNNDX3 LSR A DESY BNE DFB SFF, \$FF, \$FF JSR PRBVTE LDX #\$01 JSR PRBL2 CPY LBNG1 DX #\$03 CPY \$\$03 CPY \$\$03 </td <td><pre>;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FIELD FORMATTING ;MASK FOR 2-BIT LENGTH TEP ;OPCODE ;MASK FOR 1XXX1010 TEST ;SAVE IT ;DPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) 1XX1010 => 00101XXX ; 2) XXXYYV01 => 00111XXX ; 3) XXYYV10 => 00110XXX ; 4) XXXYVY10 => 0010XXX ; 5) XXXXX000 => 000XXXXX ; 5) XXXXX000 => 000XXXXX ;GEN FMT, LEN BYTES ;SAVE MNEMONIC TABLE INDEX ;PRINT 2 BLANKS ;PRINT INST (1-3 BYTES)' ;IN A 12 CHR FIELD ;CHAR COUNT FOR MNEMONIC INDEX</pre></td>	<pre>;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FIELD FORMATTING ;MASK FOR 2-BIT LENGTH TEP ;OPCODE ;MASK FOR 1XXX1010 TEST ;SAVE IT ;DPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) 1XX1010 => 00101XXX ; 2) XXXYYV01 => 00111XXX ; 3) XXYYV10 => 00110XXX ; 4) XXXYVY10 => 0010XXX ; 5) XXXXX000 => 000XXXXX ; 5) XXXXX000 => 000XXXXX ;GEN FMT, LEN BYTES ;SAVE MNEMONIC TABLE INDEX ;PRINT 2 BLANKS ;PRINT INST (1-3 BYTES)' ;IN A 12 CHR FIELD ;CHAR COUNT FOR MNEMONIC INDEX</pre>
FBA3: D0 04 FBA9 FBA5: A0 80 FBA7: A7 00 FBA1: 52 FBA1: 52 FBB1: 52 FBB1: 52 FBB1: 52 FBB2: 42 65 FBB2: 40 03 FBB2: 60 63 FB2: 40 63 FB2: 40 64 FB2: 40 F8 FB2: 42 F8 FB2: 42 64 FB2: 42 64 FB2: 42 64 FB2: 42 75 FB2: 42 <td>218 219 ERR 220 219 ERR 220 221 GETFMT 222 223 224 225 226 227 228 229 229 229 229 229 229 229 229 229</td> <td>BNE GETFMT LDY #\$80 LDA #\$00 TAX FDRMAT STA FDRMAT AND #\$02 STA FDRMAT AND #\$03 TAX TAX TAX TAX CPX #\$86 LSR A BCG MINDX3 LSR A BCC MINDX3 LSR A DCY #\$03 DFN MINDX3 LSR A BCC MINDX1 RTS DFD DFB #FF, \$FF, \$FF JSR INSDS1 JSR PRBYTE LDA (PCL).Y JSR PRBYTE LDA #\$03 JSR PRBYTE LDA #\$03 GCP #\$04 BCC PRNT0P LDX #\$03 CPY<l< td=""></l<></td> <td><pre>;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FILLD FORMATTING ;MASK FOR 2-BIT LENGTH TE) ;DPCODE ;MASK FOR 1XXX1010 TEST ;SAVE IT ;DPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) 1XXX1010 => 00101XXX ; 2) XXXYY101 => 00110XXX ; 3) XXXYY101 => 00110XXX ; 4) XXXYY100 => 00100XXX ; 5) XXXXYY00 => 00100XXX ; 5) XXXXY000 => 0000XXXXX ;SAVE MNEMONIC TABLE INDEX ;SAVE MNEMONIC TABLE INDEX ;PRINT 2 BLANKS ;PRINT INST (1-3 BYTES) ;IN A 12 CHR FIELD</pre></td>	218 219 ERR 220 219 ERR 220 221 GETFMT 222 223 224 225 226 227 228 229 229 229 229 229 229 229 229 229	BNE GETFMT LDY #\$80 LDA #\$00 TAX FDRMAT STA FDRMAT AND #\$02 STA FDRMAT AND #\$03 TAX TAX TAX TAX CPX #\$86 LSR A BCG MINDX3 LSR A BCC MINDX3 LSR A DCY #\$03 DFN MINDX3 LSR A BCC MINDX1 RTS DFD DFB #FF, \$FF, \$FF JSR INSDS1 JSR PRBYTE LDA (PCL).Y JSR PRBYTE LDA #\$03 JSR PRBYTE LDA #\$03 GCP #\$04 BCC PRNT0P LDX #\$03 CPY <l< td=""></l<>	<pre>;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FILLD FORMATTING ;MASK FOR 2-BIT LENGTH TE) ;DPCODE ;MASK FOR 1XXX1010 TEST ;SAVE IT ;DPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) 1XXX1010 => 00101XXX ; 2) XXXYY101 => 00110XXX ; 3) XXXYY101 => 00110XXX ; 4) XXXYY100 => 00100XXX ; 5) XXXXYY00 => 00100XXX ; 5) XXXXY000 => 0000XXXXX ;SAVE MNEMONIC TABLE INDEX ;SAVE MNEMONIC TABLE INDEX ;PRINT 2 BLANKS ;PRINT INST (1-3 BYTES) ;IN A 12 CHR FIELD</pre>
FBA3: D0 04 FBA7 FBA5: A0 80 FBA7: A7 00 FBA7: B1 B5 2E FBA7: 27 93 FBB1: B5 2F FB33: 98 78 FB34: E2 97 FB35: 60 03 FB36: C0 03 FB36: C0 03 FB36: C0 03 FB36: C0 90 F826: C4 8 F827: 79 78 F826: C4 78 F827: 60 90 F827: 79 78 F826: C4 78 F827: 68 78 F827: 68 78 F827: 68 78 F820: 20 78 F820: 20 82 F820: 20 82 F820: 20 82 F820: 20 42 F820: 20 42 F820: 20 44	218 219 219 219 219 221 221 221 222 223 224 227 227 232 232 233 234 NNDX1 232 234 234 235 234 NNDX2 232 234 234 235 234 235 240 241 242 245 242 245 242 245 245 245 245 245	DNE GETFMT LDP #\$80 LDA #\$80 LDA #\$00 TAX STA LDA #\$00 STA FCMAIT STA FCMAIT TYA STA ADD #\$03 CPX #\$03 CDY #\$03 CBR MNDX3 LSR A BCC MNNDX2 INY SER DFE SF JSR PRBYTE LDX #\$01 JSR PRB/2 CPY LENCTH DX #\$03 CPY \$03 CPY #\$03	<pre>;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FIELD FORMATTING ;MASK FOR 2-BIT LENGTH TEP ;OPCODE ;MASK FOR 1XXX1010 TEST ;SAVE IT ;DPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) 1XX1010 => 00101XXX ; 2) XXXYYV01 => 00111XXX ; 3) XXYYV10 => 00110XXX ; 4) XXXYVY10 => 0010XXX ; 5) XXXXX000 => 000XXXXX ; 5) XXXXX000 => 000XXXXX ;GEN FMT, LEN BYTES ;SAVE MNEMONIC TABLE INDEX ;PRINT 2 BLANKS ;PRINT INST (1-3 BYTES)' ;IN A 12 CHR FIELD ;CHAR COUNT FOR MNEMONIC INDEX</pre>
FBA3: D0 04 FBA9 FBA5: A0 80 FBA7: A7 00 FBA1: 52 FBA1: 52 FBB1: 52 FBB1: 52 FBB1: 52 FBB2: 42 65 FBB2: 40 03 FBB2: 60 63 FB2: 40 63 FB2: 40 64 FB2: 40 F8 FB2: 42 F8 FB2: 42 64 FB2: 42 64 FB2: 42 64 FB2: 42 75 FB2: 42 <td>218 219 ERR 220 219 ERR 220 221 GETFMT 222 223 224 225 226 227 228 229 229 229 229 229 229 229 229 229</td> <td>BNE GETFMT LDY #\$80 LDA #\$00 TAX FDRMAT STA FDRMAT AND #\$02 STA FDRMAT AND #\$03 TAX TAX TAX TAX CPX #\$86 LSR A BCG MINDX3 LSR A BCC MINDX3 LSR A DCY #\$03 DFN MINDX3 LSR A BCC MINDX1 RTS DFD DFB #FF, \$FF, \$FF JSR INSDS1 JSR PRBYTE LDA (PCL).Y JSR PRBYTE LDA #\$03 JSR PRBYTE LDA #\$03 GCP #\$04 BCC PRNT0P LDX #\$03 CPY<l< td=""></l<></td> <td><pre>;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FIELD FORMATTING ;MASK FOR 2-BIT LENGTH TEP ;OPCODE ;MASK FOR 1XXX1010 TEST ;SAVE IT ;DPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) 1XX1010 => 00101XXX ; 2) XXXYYV01 => 00111XXX ; 3) XXYYV10 => 00110XXX ; 4) XXXYVY10 => 0010XXX ; 5) XXXXX000 => 000XXXXX ; 5) XXXXX000 => 000XXXXX ;GEN FMT, LEN BYTES ;SAVE MNEMONIC TABLE INDEX ;PRINT 2 BLANKS ;PRINT INST (1-3 BYTES)' ;IN A 12 CHR FIELD ;CHAR COUNT FOR MNEMONIC INDEX</pre></td>	218 219 ERR 220 219 ERR 220 221 GETFMT 222 223 224 225 226 227 228 229 229 229 229 229 229 229 229 229	BNE GETFMT LDY #\$80 LDA #\$00 TAX FDRMAT STA FDRMAT AND #\$02 STA FDRMAT AND #\$03 TAX TAX TAX TAX CPX #\$86 LSR A BCG MINDX3 LSR A BCC MINDX3 LSR A DCY #\$03 DFN MINDX3 LSR A BCC MINDX1 RTS DFD DFB #FF, \$FF, \$FF JSR INSDS1 JSR PRBYTE LDA (PCL).Y JSR PRBYTE LDA #\$03 JSR PRBYTE LDA #\$03 GCP #\$04 BCC PRNT0P LDX #\$03 CPY <l< td=""></l<>	<pre>;SET PRINT FORMAT INDEX ID 0 ;INDEX INTO PRINT FORMAT TABLE ;SAVE FOR ADR FIELD FORMATTING ;MASK FOR 2-BIT LENGTH TEP ;OPCODE ;MASK FOR 1XXX1010 TEST ;SAVE IT ;DPCODE TO A AGAIN ;FORM INDEX INTO MNEMONIC TABLE ; 1) 1XX1010 => 00101XXX ; 2) XXXYYV01 => 00111XXX ; 3) XXYYV10 => 00110XXX ; 4) XXXYVY10 => 0010XXX ; 5) XXXXX000 => 000XXXXX ; 5) XXXXX000 => 000XXXXX ;GEN FMT, LEN BYTES ;SAVE MNEMONIC TABLE INDEX ;PRINT 2 BLANKS ;PRINT INST (1-3 BYTES)' ;IN A 12 CHR FIELD ;CHAR COUNT FOR MNEMONIC INDEX</pre>

	194 A 4			
F8EE: 85 2C F8F0: 89 00 FA	261	STA	LMNEM	FETCH 3-CHAR MNEMONIC
F8F3:85 2D	262 263	LDA STA	MNEMR, Y RMNEM	<pre>; (PACKED INTO 2-BYTES)</pre>
F8F5: A9 00	264 PRMN1	LDA	#\$00	
F8F7: A0 05	264 FREINT	LDA	#\$05	
F8F9:06 2D	266 PRMN2	ASL	RMNEM	SHIFT 5 BITS OF CHARACTER INTO 4
F8FB: 26 2C	267	ROL	LMNEM	SHIFT S BITS OF CHARMCHER 1915 -
F8FD: 2A	268	ROL	A	(CLEARS CARRY)
F8FE: 88	269	DEY		/ (CEERING CHRIST)
FBFF: DO F8 F8F9	270	BNE	PRMN2	
F901: 69 BF	271	ADC	#\$BF	ADD "?" OFFSET
F903:20 ED FD	272	JSR	COUT	DUTPUT A CHAR OF MNEM
F906: CA	273	DEX	0001	FOOT OF A GUAR OF TIMES
F907: DO EC F8F5		BNF	PRMN1	
F909:20 48 F9	275	JSR	PRBLNK	DUTPUT 3 BLANKS
F90C: A4 2F	276	LDY	LENGTH	Footi of o believe
F90E: A2 06	277	LDX	#\$06	CNT FOR 6 FORMAT BITS
F910:E0 03	278 PRADR1	CPX	#\$03	Juli Tuli D'Iuliu D'Iu
F912:F0 1C F930		BEQ	PRADR5	IF X=3 THEN ADDR.
F914:06 2E	280 PRADR2	ASL	FORMAT	A THE A G THEAT ADDITE
F916: 90 OE F926		BCC	PRADR3	
F918: BD B3 F9	282	LDA	CHAR1-1, X	
F918:20 ED FD	283	JSR	COUT	
F91E: BD B9 F9	284	LDA	CHAR2-1, X	
F921: F0 03 F926	285	BEQ	PRADR3	
F923: 20 ED FD	286	JSR	COUT	
F926: CA	287 PRADR3	DEX	0001	
F927: DO E7 F910		BNE	PRADR1	
F929:60	289	RTS		
F92A: 88	290 PRADR4	DEY		
F928:30 E7 F914	290 PRADR4	BMI	PRADR2	
F92D: 20 DA FD	292	JSR	PRBYTE	
F930: A5 2E	293 PRADR5	LDA	FORMAT	
F932:C9 E8	294	CMP	#\$E8	HANDLE REL ADR MODE
F934:B1 3A	295	LDA	(PCL),Y	SPECIAL (PRINT TARGET,
F936:90 F2 F92A	295	BCC	PRADR4	; NDT OFFSET)
F938: 20 56 F9	297 RELADR			; NUT UFFSET)
		JSR	PCADJ3	
F93B: AA	298	TAX		PCL, PCH+OFFSET+1 TO A, Y
F93C: E8	299	INX		as and amount this last
F93D: D0 01 F940	300	BNE	PRNTYX	; +1 TO Y, X
F93F: C8	301	INY		
F940: 98	302 PRNTYX	TYA		remember and the end
F941:20 DA FD	303 PRNTAX	JSR	PRBYTE	DUTPUT TARGET ADR
F944: 8A	304 PRNTX	TXA		; OF BRANCH AND RETURN
F945:4C DA FD	305	JMP	PRBYTE	
F948: A2 03	306 PRBLNK	LDX	#\$03	BLANK COUNT
F94A: A9 A0	307 PRBL2	LDA	#\$A0	LOAD A SPACE
F94C:20 ED FD	308 PRBL3	JSR	COUT	DUTPUT A BLANK
F94F: CA	309	DEX		
			PRBL2	LOOP UNTIL COUNT=0
F950: D0 F8 F94A	310	BNE		
F950: D0 F8 F94A F952: 60	311	BNE	TROLL	
F950: D0 F8 F94A F952: 60 F953: 38	311 312 PCADJ			;0=1 BYTE, 1=2 BYTE,
F950: D0 F8 F94A F952: 60 F953: 38	311	RTS	LENGTH	;0=1 BYTE, 1=2 BYTE, ; 2=3 BYTE
F950: D0 F8 F94A F952: 60 F953: 38 F954: A5 2F F956: A4 3B	311 312 PCADJ 313 PCADJ2 314 PCADJ3	RTS		
F950: D0 F8 F94A F952: 60 F953: 38 F954: A5 2F F956: A4 3B	311 312 PCADJ 313 PCADJ2	RTS SEC LDA	LENGTH	; 2=3 BYTE ; TEST DISPLACEMENT SIGN
F950: D0 F8 F94A F952: 60 F953: 38 F954: A5 2F F956: A4 38 F958: AA F959: 10 01 F950	311 312 PCADJ 313 PCADJ2 314 PCADJ3	RTS SEC LDA LDY	LENGTH	; 2=3 BYTE ; TEST DISPLACEMENT SIGN
F950: D0 F8 F94A F952: 60 F953: 38 F954: A5 2F F956: A4 38 F958: AA F959: 10 01 F95C F958: 88	311 312 PCADJ 313 PCADJ2 314 PCADJ3 315	RTS SEC LDA LDY TAX	LENGTH PCH	; 2=3 BYTE
F950: D0 F8 F94A F952: 60 F953: 38 F954: A5 2F F956: A4 38 F958: AA F959: 10 01 F95C F958: 88	311 312 PCADJ 313 PCADJR 314 PCADJ3 315 316	RTS SEC LDA LDY TAX BPL	LENGTH PCH	; 2=3 BYTE ; TEST DISPLACEMENT SIGN
F950: D0 F8 F94A F952: 60 F953: 38 F954: A5 2F F956: A4 38 F958: AA F959: 10 01 F950	311 312 PCADJ 313 PCADJ2 314 PCADJ3 315 316 317	RTS SEC LDA LDY TAX BPL DEY	LENGTH PCH PCADJ4 PCL	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH
F950: D0 F8 F94A F952: 60 F953: 38 F954: A5 2F F954: A5 2F F958: AA F959: 10 01 F950 F958: 88 F959: 65 3A	311 312 PCADJ 313 PCADJ2 314 PCADJ3 315 316 317 318 PCADJ4	RTS SEC LDA LDY TAX BPL DEY ADC	LENGTH PCH PCADJ4	; 2=3 BYTE ; TEST DISPLACEMENT SIGN
F950: D0 F8 F94A F952: 60 F953: 38 F954: A5 F956: A4 38 F958: A4 F959: 10 01 F95C F952: 65 3A F952: 90 01 F952: 70 01 F961 F961	311 312 PCADJ 313 PCADJ2 314 PCADJ3 315 315 316 317 318 PCADJ4 319 320	RTS SEC LDA LDY TAX BPL DEY ADC BCC	LENGTH PCH PCADJ4 PCL. RTS2	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH)
F950:D0 F84 F952:60 F953:38 F953:A4 38 F958:A4 38 F958:A5 F957:10 F959:10 17 F955:53 53 F955:65 34 F955:65 34 F955:65 34 F955:65 34 F955:65 34 F956:16 1761 F946:16 F941:60	311 312 PCADJ 313 PCADJ2 314 PCADJ3 315 315 316 317 318 PCADJ4 319 320 321 RTS2 322 : EMT1 RVT	RTS SEC LDA LDY TAX BPL DEY ADC BCC INY RTS	LENGTH PCH PCADJ4 PCL. RTS2	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH)
F950:D0 F8 F94A F952:60 F953:38 F955:A4 F958:A4 38 F958:A4 F958:A6 58 F959:10 F958:B8 F956:65 34 F952:65 34 F952:65 F952:65 34 F941 F956:10:01 F961 F962 F962:65 34 F941 F964:60 F962: F962: F962: 59 F962:	311 312 PCADJ 313 PCADJ2 314 PCADJ3 315 315 316 317 318 PCADJ4 319 320 321 RTS2 322 : EMT1 RVT	RTS SEC LDA LDY TAX BPL DEY ADC BCC INY RTS	LENGTH PCH PCADJ4 PCL RT52 XXXXXXYO INS THEN LEFT HAI	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950:D0 F84A F952:60 F954A F956:A4 38 F958:A4 52 F959:A5 F957 F959:B3 F955:65 F955:F90 01 F957 F955:F90 01 F9461 F940:C6 F942: F9422:	311 312 PCADJ 313 PCADJ3 314 PCADJ3 315 315 316 317 318 PCADJ4 319 320 321 RTS2 322 ; FMT1 BYT 323 ; JF Y=0 324 ; JF Y=1	RTS SEC LDA LDY TAX BPL DEY ADC BCC INY RTS	LENGTH PCH PCADJ4 PCL RT52 XXXXXXY0 INS THEN LEFT HAI THEN RIGHT H	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950:D0 F8 F94A F952:60 F953:38 F954:A5 2F F958:A4 F958:A4 B F958:A0 F957:10.01 F958:B8 F956:65 F956:C8 F962:50 F962:65 GA F946:60 F962: F962:F962: F962: F962: F962:	311 312 PCADJ 313 PCADJ3 314 PCADJ3 315 316 317 318 PCADJ4 319 320 321 RTS2 322 : FMT1 BYT 323 : FF V=0 324 : FF V=1 325 :	RTS SEC LDA LDY TAX BPL DEY ADC BCC INY RTS TES:	LENGTH PCH PCADJ4 PCL RT52 XXXXXYO INS THEN LEFT HAI THEN RIGHT H. (X=INGEX)	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950:D0 F84A F952:60 F954 F954:A5 2F F958:A4 3B F958:A5 F955 F958:A6 F957 F958:A7 F957 F958:A8 F955 F958:B8 F955:65 F955:F9001 F961 F962:C6 F962 F962:F962: F962:F962 F962:04 F962:04	311 312 PCADJ 313 PCADJ3 314 PCADJ3 315 315 316 317 318 PCADJ4 319 320 321 RTS2 322 ; FMT1 BYT 323 ; IF Y=0 324 ; IF Y=1 325 ; 326 FMT1	RTS SEC LDA LDY TAX BPL DEY ADC BCC INY RTS ES: DFB	LENGTH PCH PCLQL RT52 XXXXXYO INS THEN LEFT HA (X=INDEX) \$04	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950:D0 F8 F94A F952:40 2F F953:38 F955:A4 3 F958:A4 3 F959:10 01 F958:B8 F950:65 3A F956:160 F956:C5 5A F960:C8 F961:60 F962:F960:C8 F961:60 F962:F962:F962:F962:F962:F962:F962:F962:	311 312 PCADJ 313 PCADJ3 314 PCADJ3 315 316 317 318 PCADJ4 317 320 321 RT52 322 ; FMT1 BYT 323 ; IF Y=0 324 ; IF Y=1 325 ; 326 FMT1 327	RTS SEC LDA LDY TAX BPL DEY ADC BCC INY RTS TES: DFB DFB	LENGTH PCH PCL PCL RTS2 XXXXXYO INS THEN LEFT HAI THEN RIGHT HI (X=INDEX) \$20	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950:D0 F84 F952:60 F944 F954:45 2 F958:A4 3 F958:A4 3 F958:A5 5 F958:A6 5 F958:B8 - F955:C5 3 F955:C5 3 F946:C8 - F942: - F943:3:0 - F943:54 -	311 312 PCADJ 313 PCADJ3 314 PCADJ3 315 315 316 317 318 PCADJ4 319 320 321 RTS2 322 : FMT1 BYT 323 ; IF Y=0 324 ; IF Y=1 325 ; 326 FMT1 327 328	RTS SEC LDA LDY TAX BPL DEY ADC BCC BCC BCC INY RTS TES: DFB DFB DFB	LENGTH PCH PCLJ4 PCL RTS2 XXXXXY0 INS THEN LEPT HA THEN RIGHT H (X=INDEX) \$04 \$30	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950:D0 F8 F94A F952:40 F953:38 F955:44 38 F958:A4 38 F959:10 01 F950:65 F959:10 01 F950:65 3A F950:16:01 F950:16 F950:16:01 F950:16:03 F961:54 F962:54 F962:54 F962:54 F960:16:0 F962: F962:54 F962:54 F962:54 F962:54 F962:20 F962:20 F964:54 F963:30 F965:30 F965:30	311 312 PCADJ 313 PCADJ3 314 PCADJ3 315 316 317 318 PCADJ4 317 320 321 RT52 322 ; FMT1 BYT 323 ; IF Y=0 324 ; IF Y=0 324 ; IF Y=1 325 ; 326 FMT1 327 328	RTS SEC LDA LDY TAX BPL DEY ADC BCC INY RTS ES: DFB DFB DFB DFB	LENGTH PCH PCADJ4 PCL RTS2 XXXXXXYO INS THEN LEFT HA THEN LEFT HA (X=INDEX) \$04 \$20 \$34	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950:D0 F8 F94A F952:60 F953:38 F953:45 F958:44 F958:44 38 F958:63 F958:70 F958:70 01 F950:65 F958:70 01 F961:76 F940:C8 F942: F942: F942:204 F942:04 F943:20 F943:204 F943:20 F943:20 F943:54 F945:00 F944:54	311 312 PCADJ 313 PCADJ3 314 PCADJ3 315 316 317 318 PCADJ4 319 320 321 RT52 322 ; FMT1 BYT 323 ; IF Y=0 324 ; IF Y=1 325 ; 326 FMT1 327 328 328 329 330	RTS SEC LDA LDY TAX BPL DEY ADC BCC BCC BCC INY RTS TES: DFB DFB DFB	LENGTH PCH PCLJ4 PCL RTS2 XXXXXY0 INS THEN LEPT HA THEN RIGHT H (X=INDEX) \$04 \$30	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950:D0 F8 F94A F952:40 F953:38 F955:44 38 F958:A4 38 F959:10 01 F950:65 F959:10 01 F950:65 3A F950:16:01 F950:16 F950:16:01 F950:16:03 F961:54 F962:54 F962:54 F962:54 F960:16:0 F962: F962:54 F962:54 F962:54 F962:54 F962:20 F962:20 F964:54 F963:30 F965:30 F965:30	311 312 PCADJ 313 PCADJ3 314 PCADJ3 315 316 317 318 PCADJ4 317 320 321 RT52 322 ; FMT1 BYT 323 ; IF Y=0 324 ; IF Y=0 324 ; IF Y=1 325 ; 326 FMT1 327 328	RTS SEC LDA LDY TAX BPL DEY ADC BCC INY RTS ES: DFB DFB DFB DFB	LENGTH PCADJ4 PCL RTS2 XXXXXXYO INS THEN LEFT HAI THEN LEFT HAI (X=INDEX) \$20 \$30 \$30 \$30 \$50	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950:D0 F8 F94A F952:60 F953:38 F953:45 F959:44 F953:10 01 F950:65 F953:68 F950:65 F955:65 3A F955:70 01 F961 F940:C8 F942: F942: F942: F942: F942: F942: F942: F942: F942: F943:204 F943:20 F943:204 F943:20 F944:54 F945:30 F945:01 F946:04	311 312 PCADJ 313 PCADJ3 314 PCADJ3 315 316 317 318 PCADJ4 319 320 321 RTS2 322 ; FMT1 BYT 322 ; FMT1 BYT 322 ; FMT1 BYT 324 ; IF Y=0 324 ; IF Y=1 325 ; 326 FMT1 327 328 FMT1 327 328 GMT1 327 330 331 332	RTS SEC LDA LDY TAX BPL DEY ADC BCC INYS RTS ES: DFB DFB DFB DFB DFB DFB DFB	LENGTH PCH PCL RTS2 XXXXXYYO INS THEN LEFT HA THEN RIGHT H (XTINDEX) \$04 \$20 \$54 \$30	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950:D0 F8 F94A F952:40 F953:38 F955:44 38 F953:64 38 F955:10 01 F957 F959:10 01 F957 F957:15 75 F957:10 01 F957 F957:16 75 F957:15 01 F957 F962 F962:16 F962: F962: F962: F962:20 F962: F962: F962: F962:04 F963:20 F964:54 F963:30 F964:54 F964:54 F964:54 F964:54 F964:50 F964:54 F964:90 F964:90	311 312 PCADJ 313 PCADJ3 314 PCADJ3 315 316 317 318 PCADJ4 317 320 321 RT52 322 ; FMT1 BYT 323 ; IF Y=0 324 ; IF Y=1 325 ; 326 FMT1 327 328 327 320 331	RTS SEC LDA LDY TAX BPL DEY ADC INY RTS ES: DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	LENGTH PCADJ4 PCL RTS2 XXXXXXYO INS THEN LEFT HAI THEN LEFT HAI (X=INDEX) \$20 \$30 \$30 \$30 \$50	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950:D0 F8 F94A F952:60 F953:38 F953:45 F959:44 F953:10 01 F950:65 F953:68 F950:65 F955:65 3A F955:70 01 F961 F940:C8 F942: F942: F942: F942: F942: F942: F942: F942: F942: F943:204 F943:20 F943:204 F943:20 F944:54 F945:30 F945:01 F946:04	311 312 PCADJ 313 PCADJ3 314 PCADJ3 315 316 317 318 PCADJ4 319 320 321 RTS2 322 ; FMT1 BYT 322 ; FMT1 BYT 322 ; FMT1 BYT 324 ; IF Y=0 324 ; IF Y=1 325 ; 326 FMT1 327 328 FMT1 327 328 GMT1 327 330 331 332	RTS SEC LDA LDY TAX BPL DEY ADC BCC INYS RTS ES: DFB DFB DFB DFB DFB DFB DFB	LENGTH PCADJ4 PCL RTS2 XXXXXXYO INS THEN LEFT HAI THEN LEFT HAI THEN LEFT HAI (X=INDEX) \$04 \$20 \$54 \$30 \$80 \$94	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950:D0 F8 F94A F952:40 F953:38 F956:A4 38 F958:A4 38 F958:C F957:10 11 F957:65:65 34 F958:E8 F957:10 01 F957:65:70 17 F956:16:20 F957:16:20 F961:16:20 F962:16:20 F962:16:20 F962:16:20 F962:16:20 F962:16:20 F964:32:20 F964:32:20 F964:30 F965:30 F964:30 F964:30 F964:30 F964:20 F966:20 F964:20 F964:20	311 312 PCADJ 313 PCADJ3 314 PCADJ3 315 316 317 318 PCADJ4 319 320 322 RT52 322 RT52 322 FMT1 BYT 323 / FF Y=1 325 / 326 FMT1 327 328 327 331 332 333	RTS SEC LDA LDY TAX BPL DEY ADC INY RTS ES: DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	LENGTH PCADJ4 PCL RTS2 XXXXXXYO INS THEN LEFT HA THEN RIGHT H (X=INDEX) \$20 \$30 \$30 \$30 \$30 \$30 \$30 \$30 \$30	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950:D0 F8 F94A F952:40 F F953:38 F F958:A4 38 F958:A4 38 F958:A4 38 F958:A5 F F958:A6 7950:C5 F958:B8 F F956:C5 3A F956:C5 3A F956:C5 3A F940:C5 5A F942:C1 F F942:C2 F F942:C2 F F942:C3 F F943:20 F F944:54 F F945:30 F F946:01 F F946:03 F F946:04 F F946:03 F F946:24 F F946:25 F F946:03 F F946:24 F F946:25 F	311 312 PCADJ 313 PCADJ2 314 PCADJ3 315 316 317 318 PCADJ4 317 320 321 RTS2 322 ; FMT1 BYT 322 ; FMT1 BYT 322 ; FMT1 BYT 322 ; FMT1 BYT 324 ; IF Y=0 324 ; IF Y=1 325 ; 326 FMT1 327 328 330 331 332 333 334 335 336	RTS SECA LDV TAX BPL DEB BCC RTS ES: DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	LENGTH PCH PCL RTS2 XXXXXXYO INS THEN LEFT HAI THEN LEFT HAI THEN LEFT HAI (X=INDEX) \$04 \$54 \$50 \$50 \$50 \$50 \$50 \$50	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950:D0 F8 F94A F952:40 F953:38 F956:A4 38 F958:A4 38 F958:C F957:10 11 F957:65:65 34 F958:B8 F955:65 3A F956:45:34 F950:65:34 F950:65:24 F962:57:00 F964:54 F962:57:00 F964:54 F962:24 F962:24 F962:24 F962:30 F964:54 F963:30 F964:54 F963:30 F964:54	311 312 PCADJ 313 PCADJ3 314 PCADJ3 315 316 317 318 PCADJ4 318 PCADJ4 320 320 321 RT52 322 ; FF Y=0 324 ; IF Y=1 325 ; 326 FMT1 327 328 327 328 327 331 332 331 333 334	RTS SEC LDA LDY TAX BPL DEF BCC INY RTS ES: DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	LENGTH PCADJ4 PCL RTS2 XXXXXXY0 INS THEN LEFT HAI THEN RIGHT H (X=INDEX) \$20 \$30 \$30 \$30 \$30 \$54 \$30 \$54 \$30 \$50 \$54 \$30 \$50 \$50 \$50 \$50 \$50 \$50 \$50 \$50 \$50 \$5	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950:D0 F8 F94A F952:40 F F953:38 F F958:A4 38 F958:A4 38 F958:A4 38 F958:A5 F F958:A6 7950:C5 F958:B8 F F950:C5 3A F950:C5 3A F960:C8 F F942: F F943:20 F F943:20 F F946:01 F F947:00 F F948:03 F F948:03 F F946:03 F F946:03 F F946:04 F F946:05 F F946:06 F F946:07 F F946:08 F F946	311 312 PCADJ 313 PCADJ2 314 PCADJ3 315 316 317 318 PCADJ4 317 320 321 RTS2 322 ; FMT1 BYT 322 ; FMT1 BYT 322 ; FMT1 BYT 322 ; FMT1 BYT 324 ; IF Y=0 324 ; IF Y=1 325 ; 326 FMT1 327 328 330 331 332 333 334 335 336	RTS SECA LDV TAX BPL DEB BCC RTS ES: DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	LENGTH PCADJ4 PCL RTS2 XXXXXXYO INS THEN LEFT HAI THEN LEFT HAI THEN LEFT HAI (X=INDEX) \$20 \$20 \$30 \$30 \$30 \$30 \$30 \$30 \$30 \$30 \$30 \$3	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950:D0 F8 F94A F952:60 F953:38 F953:A4 33 F959:10 F953:A4 F955:A5 F953:A5 F957 F955:A7 F957 F955:A8 F957 F955:A9 F957 F955:A5 F957 F957:C55 F94 F942: F942 F942: F942: F942: F942: F942: F942: F942: F942: F943:20 F943:20 F944:54 F945:30 F945:30 F946:70 F946:01 F947:90 F946:70 F946:22 F946:22 F946:33 F946:33 F946:23	311 312 PCADJ 313 PCADJ3 314 PCADJ3 315 316 317 318 PCADJ4 317 320 321 RTS2 322 ; FMT1 BYT 322 ; FMT1 BYT 323 ; IF Y=1 325 ; 324 ; IF Y=1 325 ; 324 ; IF Y=1 325 ; 326 FMT1 327 328 329 330 331 332 334 335 336 337	RTS SEC LDA LDY TAX BPL ADC BCC INY RTS TS CS DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	LENGTH PCH PCAJ4 PCL RTS2 XXXXXY0 INS THEN LEFT HAI THEN RIGHT H (X=INDEX) \$04 \$00 \$54 \$50 \$50 \$50 \$00 \$50 \$50 \$50 \$50 \$50 \$50	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950:D0 F8 F94A F952:40 F953:38 F953:43 F955:44 F955:44 38 F958:43 F955: F959:10 01 F950: F958:88 F950:55 53 F955:10 01 F950: F957:10 17 F950: F957:10 01 F950: F950:15:30 F942: F942: F942:04 F942: F942: F942:04 F942: F942: F942:04 F942: F943: F943:20 F944:54 F945:30 F946:03 F946:04 F946:02 F946:04 F946:03 F946:03 F946:03 F946:00 F946:01 F946:01 F946:00 F946:00 F946:00 F946:00 F946:00 F946:00 F946:00 F946:00	311 312 PCADJ 313 PCADJ3 314 PCADJ3 315 316 317 318 PCADJ4 317 320 321 RT52 322 FMT1 BYT 323 ; FF Y=0 324 ; IF Y=1 325 ; 326 FMT1 327 326 FMT1 327 328 331 332 333 334 335 336 337 338 339	RTS SECA LDA LDY TAX BPL ADC BCC BCC BCC BCC BCC BCC BCC BCC BCC B	LENGTH PCAJJ4 PCL RTS2 XXXXXXYO INS THEN LEFT HAI THEN LEFT HAI THEN LEFT HAI (X=INDEX) \$20 \$30 \$30 \$30 \$90 \$90 \$90 \$23 \$33 \$03 \$54 \$33 \$23 \$54 \$54 \$54 \$54 \$54 \$54 \$54 \$54 \$54 \$54	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950:D0 F8 F94A F952:60 F953:38 F953:A4 38 F953:A4 38 F953:A4 38 F953:A4 38 F953:A5 57 F953:A6 75 F953:C5 34 F955:C5 34 F955:C5 34 F956:A7 F946:C5 F942: F942: F942: F942: F942: F942: F942:04 F943:20 F943:04 F943:40 F944:03 F944:03 F945:30 F946:22 F946:03 F946:23 F946:03 F946:54 F946:03 F946:54 F946:03 F946:80 F946:03 F946:80 F946:04 F946:54 F946:03 F946:80 F946:03 F946:80 F946:03 F946:80 F947:190 F947:190	311 312 PCADJ 313 PCADJ3 314 PCADJ3 315 316 317 318 PCADJ4 319 320 321 RTS2 322 ; FMT1 BYT 323 ; IF Y=0 324 ; IF Y=1 325 ; 324 ; IF Y=1 325 ; 326 FMT1 327 328 329 330 331 332 333 334 335 336 337 338 339 340	RTS SEC LDA LDA TAX BPLY ADC INYS RTS DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	LENGTH PCADJ4 PCL RTS2 XXXXXY0 INS THEN LEFT HA THEN RIGHT H (X=INDEX) \$04 \$20 \$54 \$30 \$54 \$50 \$50 \$50 \$54 \$50 \$54 \$53 \$50 \$52 \$54 \$53 \$53 \$50 \$52 \$54 \$53 \$53 \$52 \$53 \$55 \$52 \$53 \$55 \$55 \$55 \$55 \$55 \$55 \$55 \$55 \$55	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950:D0 F8 F94A F952:40 F953:38 F953:43 F955:44 F955:44 38 F958:43 F955: F959:10 01 F950: F958:88 F950:55 53 F955:10 01 F950: F957:10 17 F950: F957:10 01 F950: F950:15:30 F942: F942: F942:04 F942: F942: F942:04 F942: F942: F942:04 F942: F943: F943:20 F944:54 F945:30 F946:03 F946:04 F946:02 F946:04 F946:03 F946:03 F946:03 F946:00 F946:01 F946:01 F946:00 F946:00 F946:00 F946:00 F946:00 F946:00 F946:00 F946:00	311 312 PCADJ 313 PCADJ3 314 PCADJ3 315 316 317 318 PCADJ4 317 320 321 RT52 322 FMT1 BYT 323 ; FF Y=0 324 ; IF Y=1 325 ; 326 FMT1 327 326 FMT1 327 328 331 332 333 334 335 336 337 338 339	RTS SEC LDA LDY TAX BPL ADC BCC BCC BCY RTS CS: DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	LENGTH PCAJJ4 PCL RTS2 XXXXXXYO INS THEN LEFT HAI THEN LEFT HAI THEN LEFT HAI (X=INDEX) \$20 \$30 \$30 \$30 \$30 \$30 \$34 \$30 \$34 \$33 \$30 \$34 \$33 \$30 \$34 \$33 \$30 \$34 \$33 \$30 \$34 \$33 \$35 \$34 \$33 \$35 \$35 \$35 \$35 \$36 \$36 \$36 \$36 \$36 \$36 \$36 \$36 \$36 \$36	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950:D0 F8 F94A F952:60 F953:38 F953:A4 38 F953:A4 38 F953:A4 38 F953:A4 38 F953:A5 57 F953:A6 75 F953:C5 34 F955:C5 34 F955:C5 34 F956:A7 F946:C5 F942: F942: F942: F942: F942: F942: F942:04 F943:20 F943:04 F943:40 F944:03 F944:03 F945:30 F946:22 F946:03 F946:23 F946:03 F946:54 F946:03 F946:54 F946:03 F946:80 F946:03 F946:80 F946:04 F946:54 F946:03 F946:80 F946:03 F946:80 F946:03 F946:80 F947:190 F947:190	311 312 PCADJ 313 PCADJ3 314 PCADJ3 315 316 317 318 PCADJ4 319 320 321 RTS2 322 ; FMT1 BYT 323 ; IF Y=0 324 ; IF Y=1 325 ; 324 ; IF Y=1 325 ; 326 FMT1 327 328 329 330 331 332 333 334 335 336 337 338 339 340	RTS SEC LDA LDA TAX BPLY ADC INYS RTS DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	LENGTH PCADJ4 PCL RTS2 XXXXXXY0 INS THEN LEFT HAI THEN RIGHT H. (X=INDEX) \$04 \$20 \$54 \$30 \$54 \$30 \$54 \$33 \$02 \$54 \$33 \$02 \$54 \$33 \$00 \$54 \$54 \$54 \$54 \$54 \$54 \$54 \$54 \$54 \$54	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950:D0 F8 F94A F952:40 F953:38 F953:43 F955:44 F955:44 38 F958:64 755 F959:10 11 F95C F955:45 30 F95C:65 F955:50:65 30 F962: F962:40 F962: F962: F962:50 F962: F962: F962:61:60 F964:30 F964:54 F964:20 F964:54 F964:54 F964:30 F964:54 F964:54 F964:54 F964:54 F964:54 F964:53 F964:54 F966:03 F964:54 F964:53 F964:54 F966:54 F967:30 F966:54 F966:51 F966:54 F966:54 F966:51 F966:54 F966:54 F977:04 F977:04 F977:04	311 312 PCADJ 313 PCADJ3 314 PCADJ3 315 316 317 318 PCADJ4 317 320 321 RTS2 322 ; FMT1 BYT 323 ; FF Y=0 324 ; FF Y=0 324 ; FF Y=0 324 ; FF Y=1 325 ; 326 FMT1 327 328 331 332 333 334 335 334 335 336 337 338 339 339 340 341	RTS SEC LDA LDA TAX BPLY ADCC INY RTS FS: DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	LENGTH PCADJ4 PCL RTS2 XXXXXXYO INS THEN LEFT HAI THEN LEFT HAI THEN LEFT HAI (X=INDEX) \$20 \$30 \$30 \$30 \$30 \$30 \$33 \$30 \$34 \$30 \$30 \$34 \$30 \$34 \$30 \$54 \$33 \$50 \$54 \$33 \$54 \$30 \$54 \$54 \$54 \$50 \$54 \$54 \$50 \$54 \$54 \$50 \$54 \$50 \$50 \$54 \$50 \$50 \$50 \$50 \$50 \$50 \$50 \$50 \$50 \$50	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950:D0 F8 F94A F952:60 F953:38 F954:A3 F958:A4 F958:A4 B F958:A5 F956:A5 F958:B8 F956:65 F955:C0 F957 F952:F0 01 F952 F952:C1 F942 F942:F942:F942 F942 F942:C1 F943:320 F942:C4 F943:20 F942:C4 F943:20 F943:C0 F943:20 F944:S1 F944:54 F945:30 F946:21 F946:C3 F946:22 F946:24 F946:33 F946:25 F946:54 F946:25 F946:23 F946:23 F946:23 F946:23 F946:23 F946:20 F946:23 F946:20 F946:23 F946:20 F946:20 F947:20 F947:20 F947:20 F947:20 F947:320 F947:33	311 312 PCADJ 313 PCADJ2 314 PCADJ3 315 316 317 318 PCADJ4 317 320 321 RTS2 322 : FMT1 BYT 322 : FMT1 BYT 323 : IF Y=0 324 : IF Y=1 325 : 326 FMT1 327 328 329 331 332 333 333 334 335 337 339 340 341 342 343 344	RTS SEC LDA LDA TAX BPLY ADC INYS RTS DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	LENGTH PCADJ4 PCL RTS2 XXXXXV0 INS THEN LEFT HAI THEN RIGHT H (X=INDEX) \$04 \$20 \$54 \$30 \$54 \$30 \$50 \$50 \$54 \$53 \$00 \$54 \$53 \$00 \$54 \$53 \$00 \$54 \$50 \$54 \$50 \$50 \$54 \$50 \$50 \$50 \$50 \$50 \$50 \$50 \$50 \$50 \$50	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950:D0 F8 F94A F952:60 F953:38 F954:A3 F958:A4 F958:A4 B F958:A5 F956:A5 F958:B8 F956:65 F955:C0 F957 F952:F0 01 F952 F952:C1 F942 F942:F942:F942 F942 F942:C1 F943:320 F942:C4 F943:20 F942:C4 F943:20 F943:C0 F943:20 F944:S1 F944:54 F945:30 F946:21 F946:C3 F946:22 F946:24 F946:33 F946:25 F946:54 F946:25 F946:23 F946:23 F946:23 F946:23 F946:23 F946:20 F946:23 F946:20 F946:23 F946:20 F946:20 F947:20 F947:20 F947:20 F947:20 F947:320 F947:33	311 312 PCADJ 313 PCADJ2 314 PCADJ3 315 316 317 318 PCADJ4 317 318 PCADJ4 319 320 321 RTS2 322 FMT1 BYT 322 i FMT1 BYT 322 i FMT1 BYT 322 i FMT1 BYT 324 i FF Y=0 324 i FF Y=1 325 i 326 FMT1 327 328 327 331 334 335 337 334 335 337 338 339 340 341 342 343 344	RTSC SECA LDAY LDY RTSX BPEY ADCC INY RTS CFS: DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	LENGTH PCADJ4 PCL RTS2 XXXXXXYO INS THEN LEFT HAI THEN LEFT HAI THEN LEFT HAI (X=INDEX) \$20 \$54 \$50 \$54 \$53 \$54 \$54 \$54 \$54 \$54 \$54 \$54 \$54 \$54 \$54	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950: D0 FB F94A F952: 60 F953: 38 F954: A5 2F F956: A4 38 F958: A6 F957: 10 01 F958: C6 3A F956: 65 3A F955: 70 01 F961: 65 F942: 56 F942: 56 F942: 56 F942: 56 F942: 5742: 5742 F942: 5742 F942: 6742: 5742 F942: 5742 F942: 60 F943: 20 F942: 60 F943: 20 F943: 20 F943: 20 F944: 54 F944: 54 F945: 61 F946: 21 F946: 22 F946: 22 F946: 23 F946: 31 F946: 24 F946: 33 F946: 25 F946: 33 F946: 80 F947: 30 F947: 80 F947: 20 F947: 90 F947: 33 F947: 54 F947: 54 F947: 50 F947: 50 F947: 80 F947: 50	311 312 PCADJ 313 PCADJ2 314 PCADJ3 315 316 317 318 PCADJ4 317 320 321 RTS2 322 FMT1 BYT 322 i FWT2 322 i FWT1 BYT 323 i IF Y=0 324 i IF Y=1 325 i 326 FMT1 327 328 329 330 331 332 333 333 334 335 334 335 334 335 334 335 339 340 341 342 343 344 343 344 343 344 345 346 347 347 348 346 347 347 348 346 347 348 346 347 348 346 347 348 346 347 347 346 347 347 346 347 347 347 346 347 347 346 347 346 347 347 347 346 347 347 347 347 347 347 347 347	RTS SEC LDA LDA TAX BPLY ADCC INYS RTS DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	LENGTH PCADJ4 PCL RTS2 XXXXXXY0 INS THEN LEPT HAI THEN RIGHT H (X=INDEX) \$04 \$20 \$54 \$30 \$00 \$50 \$50 \$50 \$50 \$50 \$50 \$50 \$50 \$5	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950: D0 F84 F944 F952: 40 F953: 38 F954: 43 F955: 44 38 F953: 44 38 F956: 44 38 F957: 10 01 F957: 10 F956: 10 F956: 10 F962: 10 F962: 10 F962: 10 F962: 10 F962: 10 F962: 10 F964: 10 F972: 10 F964: 10 F972: 10	311 312 PCADJ 313 PCADJ2 314 PCADJ3 315 316 317 318 PCADJ4 317 320 321 RTS2 322 FMT1 BYT 323 ; IF Y=0 324 ; IF Y=0 324 ; IF Y=1 325 ; 326 FMT1 BYT 327 328 327 330 331 332 333 334 335 337 338 337 338 339 334 337 338 339 334 337 338 339 340 341 342 343 344 343 344 345 344 345 346 346 346 346 346 346 346 346	RTSC SECAPT LDAY LDAY LDY ADCC INY RTS CS: DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	LENGTH PCADJ4 PCL RTS2 XXXXXXYO INS THEN LEFT HAI THEN LEFT HAI THEN LEFT HAI (X=INDEX) \$20 \$54 \$53 \$50 \$54 \$53 \$50 \$54 \$33 \$50 \$54 \$33 \$50 \$54 \$33 \$50 \$54 \$54 \$53 \$54 \$53 \$54 \$54 \$54 \$54 \$54 \$54 \$54 \$54 \$54 \$54	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950: D0 FB F94A F952: 60 F933: 38 F953: 64 38 F958: 64 38 F953: 64 38 F959: 70 F953: 65 3A F950: 65 3A F955: 70 01 F961: 65 F942: 55 F942: 55 F942: 54 F942: 55 F942: 54 F942: 55 F942: 65 F942: 55 F942: 60 F942: 55 F942: 04 F942: 54 F942: 04 F943: 30 F942: 04 F943: 54 F943: 01 F944: 54 F944: 54 F945: 30 F946: 03 F946: 33 F946: 03 F946: 33 F946: 03 F946: 33 F946: 04 F946: 33 F946: 05 F946: 34 F946: 03 F946: 34 F947: 190 F947: 34 F947: 20 F947: 54 F947: 54 F947: 54 F947: 53 F947: 54 F947: 53 F947: 54 F947: 50 F947: 54 F947: 50	311 312 PCADJ 313 PCADJ2 314 PCADJ3 315 316 317 318 PCADJ4 317 320 321 RTS2 322 FMT1 BYT 322 i FWT2 322 i FWT1 BYT 323 i IF Y=0 324 i IF Y=1 325 i 326 FMT1 327 328 329 330 331 332 333 333 334 335 334 335 334 335 334 335 339 340 341 342 343 344 343 344 343 344 345 346 347 347 348 346 347 347 348 346 347 348 346 347 348 346 347 348 346 347 347 346 347 347 346 347 347 347 346 347 347 346 347 346 347 347 347 346 347 347 347 347 347 347 347 347	RTS SEC LDA LDA TAX BPLY ADCC INY RTS TS: DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	LENGTH PCH PCAJJ4 PCL RTS2 XXXXXXYO INS THEN LEFT HAI THEN LEFT HAI THEN LEFT HAI (X=INDEX) \$04 \$00 \$54 \$00 \$50 \$50 \$50 \$50 \$50 \$50 \$50 \$50 \$50	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950: D0 F8 F94A F952: 40 F953: 38 F953: A4 38 F955: A4 38 F958: A4 38 F957: 10 01 F958: A5 2F F957: 10 01 F958: A5 2F F957: 10 01 F957: 10 01 F957: 15 70 1 F957: 10 01 F957 F957: 10 01 F957 F957: 10 01 F961 F956: 20 F962: F962: 04 F962: F962: 04 F963: 20 F964: 154 F963: 30 F964: 03 F964: 03 F964: 03 F964: 03 F964: 03 F964: 03 F964: 22 F962: 54 F964: 03 F964: 03 F964: 03 F964: 04 F970: 04 F973: 33 F964: 100 F973: 20 F977: 20 F973: 33 F977: 30 F977: 33 F977: 80 F977: 80 F977: 80 F977: 90 F977: 90 F977: 90	311 312 PCADJ 313 PCADJ2 314 PCADJ3 314 PCADJ3 315 315 316 317 319 320 321 RTS2 322 ; FMT1 BYT 323 ; IF Y=0 324 ; IF Y=0 324 ; IF Y=1 325 ; 326 FMT1 327 328 329 331 332 334 335 334 335 336 337 338 334 337 339 339 339 340 341 342 343 344 345 344 345 346 347 350	RTSC SECA LDAY LDY RTS BCCC INY RTS CS: CS DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	LENGTH PCADJ4 PCL RTS2 XXXXXXYO INS THEN LEFT HAI THEN LEFT HAI (X=INDEX) \$04 \$54 \$53 \$00 \$54 \$53 \$00 \$54 \$53 \$00 \$54 \$53 \$00 \$54 \$53 \$00 \$54 \$53 \$00 \$54 \$53 \$00 \$54 \$53 \$00 \$54 \$54 \$53 \$00 \$54 \$55 \$54 \$53 \$00 \$54 \$55 \$55 \$56 \$56 \$56 \$56 \$56 \$56 \$56 \$56	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950: D0 FB F94A F952: 60 F953: 38 F952: 60 F953: 38 F953: A1 38 F957: 10 01 F953: 68 F957: 10 01 F957: 10 01 F957: 10 70 F957: 65 3A F952: 65 3A F952: 65 3A F942: 7942: 7942: 7942: 7942: 7942: 7942: 7942: 7942: 7942: 7942: 7943: 20 F942: 60 F943: 54 F943: 74 F943: 20 F944: 54 F945: 30 F945: 60 F946: 20 F946: 78 F946: 21 F946: 80 F946: 34 F946: 50 F946: 34 F946: 01 F946: 34 F946: 02 F946: 34 F946: 30 F946: 34 F946: 30 F946: 34 F946: 30 F946: 34 F947: 30 F947: 34 F947: 30 F947: 33 F977: 30 F977: 30 F977: 30 F977: 30 F977: 90 F977: 90 F977: 90 F977: 90 F977: 90 F977: 90 F977: 90	311 312 PCADJ 313 PCADJ3 314 PCADJ3 315 316 317 318 PCADJ4 319 320 321 RTS2 322 ; FMT1 BYT 322 ; FMT1 BYT 322 ; FMT1 BYT 324 ; IF Y=0 324 ; IF Y=1 325 ; 326 FMT1 327 328 331 332 333 334 335 339 334 335 339 334 335 339 334 335 336 337 338 339 334 335 336 337 338 339 334 335 336 337 338 339 334 335 336 337 338 339 334 335 336 337 337	RTSC SEC LDAY LDAY TAX BPLY ADCC INY RTS TS: DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	LENGTH PCH PCAJJ4 PCL RT52 XXXXXXYO INS THEN LEFT HAI THEN LEFT HAI THEN LEFT HAI (X=INDEX) \$20 \$54 \$30 \$54 \$30 \$54 \$33 \$00 \$54 \$33 \$22 \$54 \$33 \$00 \$54 \$33 \$24 \$33 \$00 \$54 \$33 \$25 \$54 \$33 \$30 \$54 \$54 \$33 \$30 \$54 \$54 \$33 \$50 \$54 \$54 \$55 \$54 \$55 \$54 \$55 \$54 \$55 \$54 \$55 \$55	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE
F950: D0 F8 F94A F952: 40 F953: 38 F953: A4 38 F955: A4 38 F958: A4 38 F957: 10 01 F958: A5 2F F957: 10 01 F958: A5 2F F957: 10 01 F957: 10 01 F957: 15 70 1 F957: 10 01 F957 F957: 10 01 F957 F957: 10 01 F961 F956: 20 F962: F962: 04 F962: F962: 04 F963: 20 F964: 154 F963: 30 F964: 03 F964: 03 F964: 03 F964: 03 F964: 03 F964: 03 F964: 22 F962: 54 F964: 03 F964: 03 F964: 03 F964: 04 F970: 04 F973: 33 F964: 100 F973: 20 F977: 20 F973: 33 F977: 30 F977: 33 F977: 80 F977: 80 F977: 80 F977: 90 F977: 90 F977: 90	311 312 PCADJ 313 PCADJ2 314 PCADJ3 314 PCADJ3 315 315 316 317 319 320 321 RTS2 322 ; FMT1 BYT 323 ; IF Y=0 324 ; IF Y=0 324 ; IF Y=1 325 ; 326 FMT1 327 328 329 331 332 334 335 334 335 336 337 338 334 337 339 339 339 340 341 342 343 344 345 344 345 346 347 350	RTSC SECA LDAY LDY RTS BCCC INY RTS CS: CS DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	LENGTH PCADJ4 PCL RTS2 XXXXXXYO INS THEN LEFT HAI THEN RIGHT H (X=INDEX) \$20 \$54 \$30 \$00 \$54 \$30 \$00 \$54 \$30 \$00 \$54 \$33 \$00 \$04 \$50 \$50 \$33 \$00 \$04 \$22 \$54 \$33 \$00 \$04 \$50 \$33 \$00 \$04 \$50 \$50 \$50 \$50 \$50 \$50 \$50 \$50 \$50 \$50	; 2=3 BYTE ;TEST DISPLACEMENT SIGN ; (FOR REL BRANCH) ;EXTEND NEG BY DECR PCH ;PCL+LENGTH(OR DISPL)+1 TO A ; CARRY INTO Y (PCH) TRS LF BYTE

F97D: 3B	353	DFB	\$3B
F97E: 0D	354	DFB	\$OD
F97F: 80 F980: 04	355 356	DFB	\$80 \$04
F981: 90		DFB	
F981:90	357		\$90
F982:00	358	DFB	\$00
F983: 22 F984: 44	359	DFB	\$22 \$44
F985:33	360 361	DFB	\$33
F986: 0D	362	DFB	\$0D
F987: CB	363	DFB	\$CB
F988: 44	364	DFB	\$44
F989:00	365	DFB	\$00
F98A: 11	366	DFB	\$11
F988: 22	367	DFB	\$22
F98C: 44	368	DFB	\$44
F98D: 33	369	DFB	\$33
F98F 0D	370	DFB	\$0D
F98F: C8	371	DFB	\$C8
F990:44	372	DFB	\$44
F991: A9	373	DFB	\$A9
F992:01	374	DFB	\$01
F993: 22	375	DFB	\$22
F994 44	376	DFB	\$44
F995: 33	377	DFB	\$33
F996: 0D	378	DFB	\$OD
F997:80	379	DFB	\$80
F998:04	380	DFB	\$04
F999: 90	381	DFB	\$90
F99A: 01	382	DFB	\$01
F99B: 22	383	DFB	\$22
F99C:44	384	DFB	\$44
F99D: 33	385	DFB	\$33
F99E: 0D	386	DFB	\$OD
F99F: 80	387	DFB	\$80
F9A0: 04	388	DFB	\$04
F9A1: 90	389	DFB	\$90
F9A2: 26	390	DFB	\$26
F9A3: 31	391	DFB	\$31
F9A4: 87	392	DFB	\$87
F9A5: 9A	393	DFB	\$9A
F9A6:	394 ; ZZXXXYO1	INSTR '	
F9A6: 00 F9A7: 21	395 FMT2	DFB	\$00
F9A8: 81	396	DFB	\$21
F9A9: 82	397 398	DFB DFB	\$81 \$82
F9AA: 00	399	DFB	\$00
F9AB: 00	400	DFB	\$00 \$00
F9AC: 59	401	DFB	\$59
F9AD: 4D	402	DFB	\$4D
F9AE: 91	403	DFB	\$91
F9AF: 92	404	DFB	\$92
F980:86	405	DFB	\$86
F7B1: 4A	406	DFB	\$4A
F9B2:85	407	DFB	\$85
F983: 9D	40B	DFB	\$9D
F9B4: AC	409 CHAR1	DFB	\$AC
F9B5: A9	410	DFB	\$A7
F9B6: AC	411	DFB	\$AC
F987: A3	412	DFB	\$A3
F988: A8	413	DFB	\$AB
F9B9: A4	414	DFB	\$A4
F9BA: D9	415 CHAR2	DFB	\$D9
F9BB: 00	416	DFB	\$00
F9BC: D8	417	DFB	\$D8
F9BD: A4	418	DFB	\$A4
F9BE: A4 F9BF: 00	419 420	DFB	\$A4 \$00
F78F:00		DFB	
F9C0: 1C F9C1: BA	421 MNEML		\$1C \$8A
F9C2: 1C	422 423	DFB DFB	\$1C
F9C3: 23	423	DFB	\$23
F9C4: 5D	425	DFB	\$5D
F9C5:8B	425	DFB	\$8B
F9C6: 1B	427	DFB	\$1B
E9C7: A1	428	DFB	\$A1
F9C8: 9D	429	DFB	\$9D
F9C9: 8A	430	DFB	\$BA
F9CA: 1D	431	DFB	\$1D
F9CB: 23	432	DFB	\$23
F9CC: 9D	433	DFB	\$9D
F9CD: 8B	434	DFB	\$8B
ERCE ID	435	DFB	\$1D
F9CF: A1	436	DFB	\$A1
F9D0:00	437	DFB	\$00
F9D1: 29	438	DFB	\$29
F9D2: 19	439	DFB	\$19
F9D3: AE	440	DFB	\$AE
F9D4: 69	441	DFB	\$69
F9D5: A8	442	DFB	\$A8
F9D6:19	443	DFB	\$17
F9D7: 23	444	DFB	\$23

	FRR
	IMM
	Z-PAGE
	ABS
	IMPLIED
	ACCUMULATOR
	(ZPAG, X)
	(ZPAG), Y
	ZPAG, X
	ABS, X
	ABS, Y
	(ABS)
	ZPAG, Y
	RELATIVE
	1,1
	') '
;	1,1
;	'# '
;	"("
;	'\$'
,	'Y'
;	'Y'
;	151
	1\$1

F9D8: 24	445	DFB	\$24		
F9D9: 53 F9DA: 1B	446 447	DFB	\$53 \$1B		
F9DB: 23	448	DFB	\$23		
F9DC: 24 F9DD: 53	449 450	DFB	\$24 \$53		
F9DE: 19	451	DFB	\$17 \$A1	; (A) FORMAT ABOVE
F9DF: A1 F9E0: 00	452 453	DFB	\$00		
F9E1: 1A F9E2: 5B	454 455	DFB DFB	\$1A \$5B		
F9E3: 5B	456	DFB	\$5B		
F9E4: A5 F9E5: 69	457 458	DFB	\$A5 \$69		
F9E6: 24	459	DFB	\$24	; (B) FORMAT
F9E7: 24 F9E8: AE	460 461	DFB DFB	\$24 \$AE		
F9E9: AE F9EA: A8	462 463	DFB	\$AE \$A8		
F9EB: AD	464	DFB	\$AD		
F9EC: 29 F9ED: 00	465 466	DFB DFB	\$29 \$00		
F9EE: 7C	467 468	DFB	\$7C \$00	, (C) FORMAT
F9EF: 00 F9F0: 15	469	DFB	\$15		
F9F1:9C F9F2:6D	470 471	DFB	\$90 \$60		
F9F3:9C	472	DFB	\$9C		
F9F4: A5 F9F5: 69	473 474	DFB DFB	\$A5 \$69		
F9F6: 29	475 476	DFB	\$29 \$53	; (D) FORMAT
F9F7: 53 F9F8: 84	477	DFB	\$84		
F9F9:13 F9FA:34	478 479	DFB	\$13 \$34		
F9FB: 11	480	DFB	\$11		
F9FC: A5 F9FD: 69	481 482	DFB DFB	\$A5 \$69		
F9FE: 23 F9FF: A0	483 484	DFB	\$23 \$A0	1	E) FORMAT
FA00: D8	485 MNEMR	DFB	\$D8		
FA01: 62 FA02: 5A	486 487	DFB	\$62 \$5A		
FA03: 48	488	DFB	\$48		
FA04: 26 FA05: 62	489 490	DFB	\$26 \$62		
FA06: 94 FA07: 88	491	DFB	\$94 \$88		
FA08: 54	493	DFB	\$54		
FA09: 44 FA0A: C8	494 495	DFB DFB	\$44 \$C8		
FAOB: 54	496	DFB	\$54 \$68		
FAOC: 68 FAOD: 44	497 498	DFB	\$44		
FAOE: E8 FAOF: 94	499 500	DFB DFB	\$E8 \$94		
FA10:00	501	DFB	\$00		
FA11: B4 FA12: 08	502 503	DFB	\$B4 \$08		
FA13:84	504	DFB	\$84 \$74		
FA14:74 FA15:B4	505 506	DFB DFB	\$B4		
FA16:28 FA17:6E	507 508	DFB	\$28 \$6E		
FA18: 74	509	DFB	\$74		
FA19: F4 FA1A: CC	510 511	DFB DFB	\$F4 \$CC		
FA1B: 4A FA1C: 72	512 513	DFB	\$4A \$72		
FA1D: F2	514	DFB	\$F2		
FA1E: A4 FA1F: 8A	515 516	DFB	\$A4 \$8A	;	(A) FORMAT
FA20:00	517 518	DFB DFB	\$00 \$AA		
FA21: AA FA22: A2	519	DFB	\$A2		
FA23: A2	520 521	DFB	\$A2 \$74		
FA24: 74 FA25: 74	522	DFB	\$74		(B) FORMAT
FA26: 74 FA27: 72	523 524	DFB DFB	\$74 \$72		CD7 CURPAT
FA28: 44 FA29: 68	525 526	DFB	\$44 \$68		
FA2A: B2	527	DFB	\$82		
FA2B: 32 FA2C: 82	528 529	DFB DFB	\$32 \$B2		
FA2D: 00	530	DFB	\$00 \$22		(C) FORMAT
FA2E: 22 FA2F: 00	531 532	DFB	\$00		Ser Conten
FA30: 1A FA31: 1A	533 534	DFB DFB	\$1A \$1A		
FA32: 26	535	DFB	\$26		
FA33: 26	536	DFB	\$26		

FA34: 72	537	DFB	\$72	
FA35: 72	538	DFB	\$72	
FA36: 88	539	DFB	\$88	; (D) FORMAT
FA37: CB	540	DFB	\$C8	
FA38: C4	541	DFB	\$C4	
FA39: CA	542	DFB	\$CA	
FA3A: 26	543	DFB	\$26	
FA3B: 48	544	DFB	\$48	
FA3C: 44	545	DFB	\$44	
FA3D: 44	546	DFB	\$44	
FA3E: A2	547	DFB	\$A2	(E) FORMAT
FA3F: C8	548	DFB	\$C8	
FA40:85 45	549 II	RQ STA	ACC	; *** IRQ HANDLER
FA42: 68	550	PLA		
FA43: 48	551	PHA		
FA44: 0A	552	ASL	A	
FA45: 0A	553	ASL	A	
FA46: OA	554	ASL	A	
FA47: 30 03 FA4C	555	BMI	BREAK	; TEST FOR 'BRK'
FA49: 6C FE 03	556	JMP	(IRGLOC)	USER ROUTINE VECTOR IN RAM
FA4C: 28	557 BF	REAK PLP		
FA4D: 20 4C FF	558	JSR	SAV1	; SAVE REG'S ON BREAK
FA50: 68	559	PLA		; INCLUDING PC
FA51:85 3A	560	STA	PCL.	
FA53: 68	561	PLA		
FA54:85 3B	562	STA	PCH	
FA56:6C FO 03	563	JMP	(BRKV)	BRKV WRITTEN OVER BY DISK BOOT
FA59:20 B2 F8	564 OL		INSDS1	PRINT USER PC
FASC: 20 DA FA	565	JSR	RGDSP1	; AND REGS
FA5F: 4C 65 FF	566	JMP	MON	(GO TO MONITOR (NO PASS GO, NO \$200!)
FA62: D8	567 RE			;GO TO MONITOR (NO PASS GO, NO \$200!) ;DO THIS FIRST THIS TIME
FA63: 20 84 FE	568	JSR	SETNORM	
FA66: 20 25 FB	569	JSR	INIT	
FA69:20 93 FE	570	JSR	SETVID	
FA6C: 20 89 FE	571	JSR	SETKBD	
FAGF: AD 58 CO	572 II	NITAN LDA	SETANO	; ANO = TTL HI
FA72: AD 5A CO	573	LDA	SETAN1	; ANO = TTL HI ; AN1 = TTL HI
FA75: 0001	574	DO	APPLEZE	; /RRA09B1
FA75: A0 05	575	LDY	#5	CODE=INIT/RRA0981
FA77: 20 B4 FB	576	JSR	GOTOCX	DO APPLEZE INIT/RRA0981
FA7A: EA	577	NOP		/RRA0981
FA7B:	578	ELSE	•	/RRA0781
S	579	LDA	CLRAN2	AN2 = TTI I D
S	580	LDA	CLRANG	AN3 = TTL LO
FA7B:	581	FIN	ULIVINO .	7 RRA0981
FA7B: AD FF CF	582	LDA	CLRROM	; TURN OFF EXTNSN ROM
FA7E: 2C 10 CO	583	BIT	KBDSTRB	CLEAR KEYBOARD
FA81: D8	584 NE	EWMON CLD	KBDBIKB	CLEAR RETBOARD
FA82: 20 3A FF	585	JSR	BELL	; CAUSES DELAY IF KEY BOUNCES
FA85: AD F3 03	586	LDA		, CRUSES DELAT IF KET BUUNCES
FA88: 49 A5	587		SOFTEV+1 #\$A5	; IS RESET HI ; A FUNNY COMPLEMENT OF THE
FABA: CD F4 03	588	EDR	PWREDUP	PWR UP BYTE ???
		BNE		NO SO PWRUP
EAOD DO 17 EAA/			PWRUP SOFTEY	
FABD: DO 17 FAA6	589			NEC OFF IC OC D OT OT
FABF: AD F2 03	590	LDA	SOFTEN	YES SEE IF COLD START
FABF: AD F2 03 FA92: DO OF FAA3	590 591	BNE	NOFIX	; YES SEE IF COLD START ; HAS BEEN DONE YET?
FA8F:AD F2 03 FA92:D0 0F FAA3 FA94:A9 E0	590 591 592	BNE LDA	NOFIX #\$E0	; YES SEE IF COLD START ; HAS BEEN DONE YET? ; DOES SOFT ENTRY VECTOR POINT AT BASIC
FABF: AD F2 03 FA92: D0 0F FAA3 FA94: A9 E0 FA96: CD F3 03	590 591 592 593	BNE LDA CMP	NOFIX #\$E0 SOFTEV+1	; YES SEE IF COLD START ; HAS BEEN DONE YET? ; DOES SOFT ENTRY VECTOR POINT AT BASIC:
FABF: AD F2 03 FA92: D0 0F FAA3 FA94: A9 E0 FA96: CD F3 03 FA99: D0 08 FAA3	590 591 592 593 594	BNE LDA CMP BNE	NDFIX #\$E0 SOFTEV+1 NDFIX	; YES SEE IF COLD START ; HAS BEEN DONE VET? ; DOES SOFT ENTRY VECTOR POINT AT BASIC ; YES SO REENTER SYSTEM
FA8F: AD F2 03 FA92: D0 0F FAA3 FA94: A9 E0 FA95: CD F3 03 FA99: D0 08 FAA3 FA98: A0 03	590 591 592 593 594 595 F:	BNE LDA CMP BNE IXSEV LDY	NOFIX #\$E0 SOFTEV+1 NOFIX #3	; YES SEE IF COLD START ; HAS BEEN DONE YET? ; DOES SOFT ENTRY VECTOR POINT AT BASIC ; YES SO REENTER SYSTEM ; NO SO POINT AT WARM START
FABF: AD F2 03 FA92: D0 0F FAA3 FA94: A9 E0 FA96: CD F3 03 FA97: D0 08 FAA3 FAA3 FAA95: A0 33 FA98: A0 03 FA98: A0 03 FA98: A0 34	590 591 592 593 594 595 F: 596	BNE LDA CMP BNE IXSEV LDY STY	NOFIX #\$E0 SOFTEV+1 NOFIX #3 SOFTEV	 YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC' YES SO REENTER SYSTEM NO SO POINT AT WARM START FOR NEXT RESET
FABF: AD F2 03 FA92: D0 0F FAA3 FA94: A9 E0 FA96: CD F3 03 FA97: D0 08 FAA3 FAA3 FAA95: A0 33 FA98: A0 03 FA98: A0 03 FA98: A0 34	590 591 592 593 594 595 F: 596 597	BNE LDA CMP BNE IXSEV LDY STY JMP	NOFIX #\$EO SOFTEV+1 NOFIX #3 SOFTEV BASIC	 YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC' YES SO REENTER SYSTEM NO SO POINT AT WARM START FOR NEXT RESET
FABF:AD F2 03 FA72:D0 0F FAA3 FA74:A7 E0 FA76:CD F3 03 FA79:D0 08 FAA3 FA79:A0 03 FA79:A0 03 FA79:C F2 03 FAA0:4C 00 E0 FAA3:6C F2 03	590 591 592 593 594 595 F: 596 597 598 NO	BNE LDA CMP BNE IXSEV LDY STY JMP DFIX JMP	NOFIX #\$EO SOFTEV+1 NOFIX #3 SOFTEV BASIC (SOFTEV)	; YES SEE IF COLD START ; HAS BEEN DONE YET? ; DOES SOFT ENTRY VECTOR POINT AT BASIC ; YES SO REENTER SYSTEM ; NO SO POINT AT WARM START
FABF:AD F2 03 FA72:D0 OF FAA3 FA74:A7 E0 FA75:CD F3 03 FA79:A0 08 FAA3 FA79:D0 08 FAA3 FA79:D0 08 FAA3 FA79:BC 03 FA70:BC F2 03 FAA0:4C 00 E0 FAA3:6C F2 03 FAA6:	590 591 592 593 594 595 F: 596 597 598 NC 598 NC	BNE LDA CMP BNE IXSEV LDY STY JMP OFIX JMP	NOFIX #SEO SOFTEV+1 NOFIX #3 SOFTEV BASIC (SOFTEV)	 YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC' YES SO REENTER SYSTEM NO SO POINT AT WARM START FOR NEXT RESET
FABF:AD F2 03 FA72:D0 0F FAA3 FA74:A7 E0 FA79:CD F3 03 FA79:D0 08 FAA3 FA79:B0 08 FAA3 FA79:B0 F2 03 FAA0:RC 72 03 FAA0:4C 00 E0 FAA3:6C F2 03 FAA6: FAA6:5	590 591 592 593 594 595 F: 596 597 598 NC 598 NC 599 **	BNE LDA CMP BNE IXSEV LDY JMP DFIX JMP ************************************	NDFIX #\$EO SDFTEV+1 NDFIX #3 SDFTEV BASIC (SDFTEV) *** APPLEII	 YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC YES SO REENTER SYSTEM ND SO POINT AT WARM START FOR NEXT RESET AND DO THE COLD START SOFT ENTRY VECTOR
FABE: AD F2 03 FAP2: DO OF FAA3 FAP4: AP E0 FAA3 FAP4: AP E0 FAA3 FAP5: DD F3 03 FAP7: DD 08 FAA3 FAP7: BC FAA3 FAA3 FAP5: BC F2 03 FAA0: 4C 00 E0 FAA3: 6C F2 03 FAA5: FAA5 FAA5	590 591 592 593 594 595 F: 595 F: 597 598 NC 597 *** 600 PL 601 SE	BNE LDA CMP BNE IXSEV LDY STY JMP OFIX JMP OFIX JMP UFIX JSR ETPC3 EGU	NOFIX #\$EO SDFTEV+1 NDFIX #3 SDFTEV BASIC (SDFTEV) *** APPLEII *	 YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC' YES SO REENTER SYSTEM NO SO POINT AT WARM START FOR NEXT RESET
FABF:AD F2 03 FA92:D0 OF FAA3 FA94:A9 E0 FA95:C0 F3 03 FA97:D0 08 FAA3 FA97:D0 08 FAA3 FA97:BC F2 03 FAA0:4C 00 E0 FAA3:EC F2 03 FAA6: FAA6:C F2 03 FAA6: FAA7: FAA7 FAA7:A2 05	590 591 592 593 594 595 F: 596 597 598 NC 597 ** 600 PC 602	BNE LDA CMP IXSEV LDY STY OFIX JMP ************************************	NDFIX #\$EO SDFTEV+1 NDFIX #3 SDFTEV BASIC (SDFTEV) ** APPLEII * 5	 YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC' YES SO REENTER SYSTEM NO SO POINT AT WARM START FOR NEXT RESET AND DO THE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS
FABE: AD F2 03 FAP2: DO OF FAA3 FAP4: AP E0 FAA3 FAP4: DO OF FAA3 FAP5: DD 03 FAA3 FAP9: BO 08 FAA3 FA97: BC FAA G FAP4: AC 00 E FAA3: 4C 00 E FAA6: 20 60 FB FAA2: AC 50 FAA2 FAA2: AS 55 FAA5	590 591 592 593 594 595 F: 596 597 598 NC 597 *1 600 PC 602 SE 602 SE	BNE LDA CMP BNE IXSEV LDY JMP DFIX JMP FIX JMP FTPG3 EQU LDX ETPG3 EQU LDX	NDFIX #\$E0 SDFTEV+1 NDFIX #3 SDFTEV BASIC (SDFTEV) *** APPLEII * #5 PWRCON-1, X	 YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC YES SO REENTER SYSTEM NO SO POINT AT WARM START FOR NEXT RESET AND DO THE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS WITH CNTRL B ADRS
FABF:AD F2 03 FA92:D0 OF FAA3 FA94:A9 E0 FA96:C0 F3 03 FA97:D0 08 FAA3 FA97:D0 08 FAA3 FA97:D0 08 FAA3 FA97:D0 C2 FAA3:C0 F2 03 FAA6:C72 03 FAA6:C72 03 FAA7:C72 FAA9 FAA7:C72 FAA9 FAA9:A2 05 FAA8:BD FC FA FAA8:D FC FA	590 591 592 593 594 595 F: 596 597 598 NC 599 ** 600 PL 601 SE 602 603 SE	BNE LDA CMP BNE IXSEV LDY OFIX JMP OFIX JMP OFIX JMP ETPC3 EQU LDX ETPLP LDA STA	NDFIX #\$EO SDFTEV+1 NDFIX #3 SDFTEV BASIC (SDFTEV) ** APPLEII * 5	 YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC' YES SO REENTER SYSTEM NO SO POINT AT WARM START FOR NEXT RESET AND DO THE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS
FABE: AD F2 03 FAP2: DO 0F FAA3 FAP4: AP EO FAA3 FAP4: AP EO FAA3 FAP4: DO DF3 03 FAP5: DO D6 FAA3 FAP9: AO 03 FAA3 FAP9: AC 00 E FAA0: 4C 00 EO FAA3: AC F0 EO FAA3: AC 03 FAA3 FAA3: AC 04 EO FAA3: AC 05 FAA5 FAA5: CO 60 FB FAA7: AC 05 FAA9 FAA8: BD FC FA FAA8: PD EF 03 FAA8: PD EF 03 FAB1: CA CA CA	570 571 572 573 574 575 F: 576 577 578 NC 577 * 600 PL 601 SE 602 603 SE 604 605	BNE LDA CMP IXSEV LDY JMP FFIX JMP FFIX JMP FFIX GU ETPG3 EGU LDX ETPLP LDA STA DEX	NDFIX ##EO SDFTEV+1 NDFIX #3 SDFTEV BASIC (SDFTEV) ** APPLEII * # BRKU-1,X	 YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC YES SO REENTER SYSTEM NO SO POINT AT WARM START FOR NEXT RESET AND DO THE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS WITH CNTRL B ADRS
FABF:AD F2 03 FA92:D0 OF FAA3 FA94:A9 EO FA94 FA95:CD FG 03 FA97:D0 08 FAA3 FA97:D0 06 FAA3 FAA3:C0 03 FAA3 FAA3:C6 F2 03 FAA4: FAA5 FAA5 FAA7:A2 05 FAA5 FAA8:B0 FC FA FAA2:O DE F03 FAB1:CA FAA5 FAA5	590 591 592 593 594 595 F: 596 NC 597 ** 600 PL 601 SE 603 SE 605 606	BNE LDA CMP BNE IXSEV LDY OFIX JMP OFIX JMP OFIX JMP ETPC3 EQU LDX ETPLP LDA STA DEX BNE	NOFIX #*E0 SOFTEV+1 NOFIX #3 SOFTEV BASIC (SOFTEV) ** APPLEII * # 5 PWRCON-1,X BRKV-1,X SETPLP	 YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC' YES SO REENTER SYSTEM NO SO POINT AT WARN START FOR NEXT RESET AND DO IHE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS WITH CNTRL B ADRS OF CURRENT BASIC
FABE: AD F2 03 FAP2: DO OF FAA3 FAP4: AP EO FAP3 FAP4: AP EO FAP3 FAP5: DO D6 FAA3 FAP9: DO 08 FAA3 FAP9: BC F2 03 FAP9: BC F2 03 FAA3 FAA3 FAA3 FAA3 F4 00 D FAA2: AC F2 03 FAA3 FAA3: AC F2 F3 FAA3 FAA4: AC F2 F3 F4A9 FAA3: AC F4 F4A9 FAA9 FAA4: AC F4A3 F4A9 F4A9 FAA8: AC F4 F4A9 F4A9 FAA8: BD FC F4 FAA8 FAB2: DD F7 FAA8 FAB2: AP C8 FAA9 C8	570 571 572 573 574 575 576 577 578 577 578 80 577 800 601 602 602 603 604 605 606 607	BNE LDA CMP IXSEV LDY JMP OFIX JMP F************************************	NDFIX #\$E0 SDFTEV+1 NDFIX \$SDFTEV BASIC (SDFTEV) *** APPLEII * * BRKV-1,X SETPLP \$\$C8	 YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC YES SO REENTER SYSTEM NO SO POINT AT WARM START FOR NEXT RESET AND DO THE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS WITH CNTRL B ADRS OF CURRENT BASIC LOAD HI SLOT +1
FABF:AD F2 03 FA92:D0 OF FAA3 FA94:A9 EO FA93:CD FA33 FA94:CD F3 FA33 FA33 FA95:CD FA3 FA33 FA33 FA95:A0 03 FA33:C F2 03 FA47:A0 60 EO FAA3 FAA3:C F2 03 FAA3:AC FA 00 G0 FA FAA3:C F2 03 FAA3:AC FA 00 G0 F3 FAA4 FAA4 FA FAA4:A2 C6 F2 03 FAA7 FAA9	570 571 572 573 574 575 576 577 578 577 578 600 601 577 600 601 560 602 603 604 605 604 605 606 607 608	BNE LDA CMP BNE IXSEV LDY OFIX JMP OFIX JMP OFIX JMP ETPG3 EQU LDX ETPLP LDA STA BNE LDA STX	NOFIX #*E0 SOFTEV+1 NOFIX #3 SOFTEV BASIC (SOFTEV) ** APPLEII * #5 PWRCON-1,X BRKV-1,X SETPLP #\$C8 LDC0	<pre> ; YES SEE IF COLD START ; HAS BEEN DONE YEY? ; DOES SOFT ENTRY VECTOR POINT AT BASIC ; YES SO REENTER SYSTEM ; NO SO POINT AT WARN START ; FOR NEXT RESET ; AND DO THE COLD START ; SOFT ENTRY VECTOR ; SET PAGE 3 VECTORS ; WITH CNTRL B ADRS ; OF CURRENT BASIC ; LOAD HI SLOT +1 ; SETPG3 MUST RETURN X=0 </pre>
FABF: AD F2 03 FAP2: D0 OF FAA3 FAP4: AP E0 FAA3 FAP5: D0 OB FAA3 FAA7: D0 OB FAA3 FAA4: 20 60 FB FAA4: 20 50 FAA7 FAA7: A2 05 FAA5 FAA8: BD FC FA FAA5: CA FA OB FAA5: A7 CB FA FA56: B5 O1	570 571 572 573 574 575 576 577 578 577 578 600 FL 601 577 400 FL 601 507 800 602 604 604 605 606 607 608 609	BNE LDA CMP IXSEV LDY STY STY MPD DFIX MPP TPF03 EQU ETPC3 EQU LDA STA BNE DEX BNE STA STA	NOFIX #*EO SOFTEV+1 NOFIX *3 SOFTEV BASIC (SOFTEV) *** APPLEII * FWRCON-1,X BRKV-1,X SETPLP **CB LOCC LOC1	 YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC YES SO REENTER SYSTEM ND SO POINT AT WARM START FOR NEXT RESET AND DO THE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS WITH CNTRL B ADRS OF CURRENT BASIC LOAD HI SLOT +1 SETPG3 MUST RETURN X=0 SET PA H
FABF: AD F2 03 FA92: DO OF FAA3 FA94: A9 EO FA93: CD FFA33 FA94: A9 EO FA34: CD F3 FA94: CD F3 G3 FA43: A0 FA97: CD CB FAA3 FAA3 FA98: AO C3 FAA3 FAA3 FAA9: CC C2 03 FAA46: FAA6: AC 00 EO FAA47: FAA7: A2 C5 FAA9: BD FC FAA8: AD FC FA FAA9: C3 FAA8: DD FC FA FAA9: C4 FAA8: DD FC FA FAA9: C4 FAA8: DD FC FA FAA9: C4 FAA8: C0 FA FAA9 FA FAA8: C4 FA FA FAA9 FAA9 FA FA FA FAA9 FA FA FA FAA9 FA FA FA FAA9 <td>570 571 572 573 574 575 575 576 577 400 577 400 577 401 56 602 403 502 403 502 404 405 606 407 608 607 608</td> <td>BNE LDA CMP IXSEV LDY STY OFIX JMP OFIX JMP ETPG3 EQU ETPG3 EQU ETPC3 EQU ETPLP LDA STA BNE LDA STX STA STA</td> <td>NDFIX #*EO SOFTEV+1 NDFIX *3 SOFTEV BASIC (SOFTEV) ** APPLEII * * * * BRKV-1,X SETPLP * *CCB LOCC LOC1 *7</td> <td><pre> ; YES SEE IF COLD START ; HAS BEEN DONE YEY? ; DOES SOFT ENTRY VECTOR POINT AT BASIC ; YES SO REENTER SYSTEM ; NO SO POINT AT WARN START ; FOR NEXT RESET ; AND DO THE COLD START ; SOFT ENTRY VECTOR ; SET PAGE 3 VECTORS ; WITH CNTRL B ADRS ; OF CURRENT BASIC ; LOAD HI SLOT +1 ; SETPG3 MUST RETURN X=0 </pre></td>	570 571 572 573 574 575 575 576 577 400 577 400 577 401 56 602 403 502 403 502 404 405 606 407 608 607 608	BNE LDA CMP IXSEV LDY STY OFIX JMP OFIX JMP ETPG3 EQU ETPG3 EQU ETPC3 EQU ETPLP LDA STA BNE LDA STX STA STA	NDFIX #*EO SOFTEV+1 NDFIX *3 SOFTEV BASIC (SOFTEV) ** APPLEII * * * * BRKV-1,X SETPLP * *CCB LOCC LOC1 *7	<pre> ; YES SEE IF COLD START ; HAS BEEN DONE YEY? ; DOES SOFT ENTRY VECTOR POINT AT BASIC ; YES SO REENTER SYSTEM ; NO SO POINT AT WARN START ; FOR NEXT RESET ; AND DO THE COLD START ; SOFT ENTRY VECTOR ; SET PAGE 3 VECTORS ; WITH CNTRL B ADRS ; OF CURRENT BASIC ; LOAD HI SLOT +1 ; SETPG3 MUST RETURN X=0 </pre>
FABF: AD F2 03 FA92: D0 OF FAA3 FA94: A9 E0 FAA3 FA94: A9 E0 FAA3 FA97: D0 08 FAA3 FA97: D0 08 FAA3 FA97: BC F2 03 FAA7: BC F2 03 FAA3: A0: 4C 00 E0 FAA4: A0: 7 FAA9 FAA9 FAA8: A0: 7 FAA9 FAA9 FAA2: A0: 7 FAA8 FAA9 FAA2: A0: 70 EF A5 FAA8: A0: CA FAB8: A0 FAB8: A0 FAB8: A0: 07 FAB8: A0 FAB8: A0 FAB2: A0: 07 FAB2: CA FAB2: CA	570 571 572 573 574 575 575 575 577 400 577 400 577 400 507 401 505 4005 4005 4005 4005 4005 4005 4	BNE LDA CMP IXSEV LDY JMP OFIX JMP ************************************	NDFIX #*EO SOFTEV+1 NDFIX *3 SOFTEV BASIC (SOFTEV) *** APPLEII * * PWRCON-1,X BRKV-1,X SETPLP #*CB LOCO LOC1 *7 LOC1	 YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC YES SO REENTER SYSTEM ND SO POINT AT WARM START FOR NEXT RESET AND DO THE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS WITH CNTRL B ADRS OF CURRENT BASIC LOAD HI SLOT +1 SETPG3 MUST RETURN X=0 SET PA H
FABF: AD F2 03 FA92: DO OF FAA3 FA94: A9 EO FA93 FA94: CD FF3 FA33 FA97: DD 08 FAA3 FA98: CD F3 FAA3 FA97: CD FAA3 FAA3 FA78: A0 03 FAA3 FAA7 FAA3 FAA5 FAA4 C0 EO FAA5: AC 04 FA FAA42: AC 05 FAA7 FAA7: AC FA FAA7 FAA2: AD FA FAA7 FAA2: DD FA FAA7 FAA2: AD C5 FAA8 FAA2: DD FA FAA8 FAB2: CD FA FAB2 FAB2: CD FA FAB2 FAB2: SD FA FAB2 FAB2: SD FA FAB2 FAB3: SD T FAB2 FAB2: C5 O1 FAB2 FAB2: C5 O1 <td>570 571 572 573 574 575 575 576 577 577 400 577 401 577 402 602 577 402 603 504 603 504 603 604 605 607 608 607 608 607 610 512</td> <td>BNE LDA CMP IXSEV LDY STY STY JMP DFIX JMP DFIX JMP ETPC3 EQU ETPC3 EQU ETPC3 EQU ETPC9 LDA STA BNE LDA STX STA LCJP LDY DEC LDA</td> <td>NOFIX #*E0 SOFTEV+1 NOFIX #3 SOFTEV BASIC (SOFTEV) (SOFTEV) ** APPLEII * #5 PWRCON-1,X BRKV-1,X SETPLP #\$C8 LOC1 C1 T LOC1 LOC1</td> <td> YES SEE IF COLD START HAS BEEN DONE YEY? DDES SOFT ENTRY VECTOR POINT AT BASIC' YES SO REENTER SYSTEM NO SO POINT AT WARN START FOR NEXT RESET AND DO THE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS WITH CNTRL B ADRS OF CURRENT BASIC LOAD HI SLOT +1 SETPG3 MUST RETURN X=0 SET PTR H Y IS BYTE PTR </td>	570 571 572 573 574 575 575 576 577 577 400 577 401 577 402 602 577 402 603 504 603 504 603 604 605 607 608 607 608 607 610 512	BNE LDA CMP IXSEV LDY STY STY JMP DFIX JMP DFIX JMP ETPC3 EQU ETPC3 EQU ETPC3 EQU ETPC9 LDA STA BNE LDA STX STA LCJP LDY DEC LDA	NOFIX #*E0 SOFTEV+1 NOFIX #3 SOFTEV BASIC (SOFTEV) (SOFTEV) ** APPLEII * #5 PWRCON-1,X BRKV-1,X SETPLP #\$C8 LOC1 C1 T LOC1 LOC1	 YES SEE IF COLD START HAS BEEN DONE YEY? DDES SOFT ENTRY VECTOR POINT AT BASIC' YES SO REENTER SYSTEM NO SO POINT AT WARN START FOR NEXT RESET AND DO THE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS WITH CNTRL B ADRS OF CURRENT BASIC LOAD HI SLOT +1 SETPG3 MUST RETURN X=0 SET PTR H Y IS BYTE PTR
FABF: AD F2 03 FA92: D0 OF FAA3 FA94: A9 E0 FAA3 FA94: A9 E0 FAA3 FA97: D0 D8 FAA3 FA97: D0 D8 FAA3 FA97: BC F2 03 FAA7: AC 00 E0 FAA3: AC F2 03 FAA6: C F2 03 FAA6: AC 00 E0 FAA3: AC 00 E7 FAA6: C F2 03 FAA7: C FAA7 FAA7 FAA7: AC 05 FAA7 FAA2: AC 05 FAA3: CA FAA2: A7 CB FA FAB1: CA FAB2: CA FAB3: CA FAB2: A7 CB FAA8 FAB2: CA FAB3 FAB3 FAB2: CA FAB3 FAB3 FAB3: CA FAB3 FAB3 FAB3: CA FAB3 FAB3 FAB3: CA FAB3	570 572 573 574 575 F: 575 F: 576 S77 578 NC 607 S1 603 SE 604 605 606 607 SL 608 607 SL 6113	BNE LDA CMP IXSEV LDY JMP OFIX JMP ************************************	NDFIX ##EO SOFTEV+1 NDFIX #3 SOFTEV BASIC (SOFTEV) *** APPLEII * * FWRCON-1,X BRKV-1,X SETPLP #\$CB LOC0 LOC1 47 LOC1 LOC1 LOC1 LOC1	<pre> ; YES SEE IF COLD START ; HAS BEEN DONE YEY? ; DOES SOFT ENTRY VECTOR POINT AT BASIC ; YES SO REENTER SYSTEM ; ND SO POINT AT WARM START ; FOR NEXT RESET ; AND DO THE COLD START ; SOFT ENTRY VECTOR ; SET PAGE 3 VECTORS ; WITH CNTRL B ADRS ; OF CURRENT BASIC ; LDAD HI SLOT +1 ; SETPG3 MUST RETURN X=0 ; SET PT H ; Y IS BYTE PTR ; AT LAST SLOT YET? </pre>
FABF: AD F2 03 FA92: DO OF FAA3 FA94: A9 EO FA93 FA94: CD FF3 FA33 FA94: CD F3 33 FA94: CD F3 33 FA94: CD F3 33 FA97: CD FAA3 FAA3 FA78: CD F3 33 FAA9: A0 C3 FAA3 FAA7: CD F2 03 FAA6: A0 C3 FAA7 FAA7 FAA7 FAA7 FAA7 A2 C5 FAA8: A2 C5 FAA7 FAA2: A7 CE FAA7 FAA2: DF FA FAA8 FAA8: ED F7 FAA8 FAB2: CA FA FAA8 FAB2: CA FA FA FAB2: CA FA FA FAB2: CA FA FA FAB2: CA FA FA FAB2: CA FA F	570 571 572 573 574 575 575 577 577 578 601 601 577 600 601 602 603 604 605 605 604 605 606 607 608 608 606 607 612 612 612 614	BNE LDA CMP IXSEV LDY JMP OFIX JMP OFIX JMP DFIX JMP ETPG3 EQU ETPC3 EQU ETPC9 LDA STA BNE LDA STX STA STA BEQ BEQ BEQ BEQ	NDFIX #\$E0 SDFTEV+1 NDFIX BASIC (SDFTEV) BASIC (SDFTEV) ** APPLEII * * BRKV-1,X SETPLP H&CON-1,X SETPLP #\$C8 LDC0 LDC1 LDC1 LDC1 BFIXSEV	 YES SEE IF COLD START HAS BEEN DONE YEY? DDES SOFT ENTRY VECTOR POINT AT BASIC' YES SO REENTER SYSTEM NO SO POINT AT WARN START FOR NEXT RESET AND DO THE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS WITH CNTRL B ADRS OF CURRENT BASIC LOAD HI SLOT +1 SETPG3 MUST RETURN X=0 SET PTR H Y IS BYTE PTR
FABF: AD F2 03 FA92: D0 OF FAA3 FA94: AP E0 FAA3 FA94: AP E0 FAA3 FA97: D0 08 FAA3 FA97: D0 08 FAA3 FA97: BC 03 FAA3 FAA7: C F2 03 FAA3: AC 00 E0 FAA3: AC 00 E0 FAA3: AC 00 E0 FAA3: AC 00 E0 FAA3: AC 00 FAA4: FAA7: C FAA7 FAA7 FAA2: AC 05 FAA4: FAA2: AC 05 FAA5 FAA2: AC 05 FAA5 FAA2: AC 05 FAA5 FAA2: AC 05 FAA5 FAA2: AC C FAA5 FAA3: AC C FAA5 FAB4: AC <	570 572 572 574 575 574 575 575 577 578 600 601 577 608 602 603 605 604 605 604 605 607 608 607 608 607 608 607 608 612 613 615	BNE LDA CMP IXSEV LDY JMP OFIX JMP STY STY STY STY ETPO3 EGU LDX ETPLP LDA STA BNE LDA STA LDJP LDA STA STA STA STA STA STA	NDFIX #*EO SOFTEV+1 NDFIX #3 SOFTEV BASIC (SOFTEV) *** APPLEII *5 PWRCON-1,X BRKV-1,X SETPLP #*CC LOCO LOC1 UCC1 #*CO FIXSEV MSLOT	<pre> YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC YES SO REENTER SYSTEM NO SO POINT AT WARM START FOR NEXT RESET AND DO THE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS WITH CNTRL B ADRS OF CURRENT BASIC LOAD HI SLOT +1 SETPG3 MUST RETURN X=0 SET PTR H Y IS BYTE PTR AT LAST SLOT YET? YES AND IT CAN'T BE A DISK </pre>
FABF: AD F2 03 FA92: D0 OF FAA3 FA94: AP E0 FAA3 FA94: AP E0 FAA3 FA97: D0 08 FAA3 FA97: D0 08 FAA3 FA97: BC 03 FAA3 FAA7: C F2 03 FAA3: AC 00 E0 FAA3: AC 00 E0 FAA3: AC 00 E0 FAA3: AC 00 E0 FAA3: AC 00 FAA4: FAA7: C FAA7 FAA7 FAA2: AC 05 FAA4: FAA2: AC 05 FAA5 FAA2: AC 05 FAA5 FAA2: AC 05 FAA5 FAA2: AC 05 FAA5 FAA2: AC C FAA5 FAA3: AC C FAA5 FAB4: AC <	570 572 573 574 575 F: 575 F: 577 NC 577 NC 602 SE 602 SE 604 605 606 607 SL 606 607 SL 606 607 SL 606 607 SL 611 612 614 N)	BNE LDA CMP IXSEV LDY STY JMP OFIX JMP FIPG3 EQU ETPC3 EQU ETPC3 EQU DEX BNE LDA STA LDDP LDY DEX DEX BNE STA STA STA STA STA STA STA STA STA STA	NOFIX ##E0 SOFTEV+1 NOFIX BASIC (SOFTEV) BASIC (SOFTEV) ** APPLEII * #5 BRKV-1,X SETPLP H%CON-1,X SETPLP H%CC LOCO LOCO LOCO LOCO CI LOCO CI LOCO FIXSEV MSLOT (LOCO),Y	<pre> YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC YES SO REENTER SYSTEM NO SO POINT AT WARN START FOR NEXT RESET AND DO THE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS HITH CNTRL B ADRS OF CURRENT BASIC LOAD HI SLOT +1 SETPG3 MUST RETURN X=0 SET PTR H Y IS BYTE PTR AT LAST SLOT YET? YES AND IT CAN'T BE A DISK FETCH A SLOT BYTE </pre>
FABF: AD F2 03 FA92: D0 OF FAA3 FA94: A7 E0 FA94: A7 E0 FA94: CD F3 FA97: D0 08 FA97: D0 03 FA97: B0 C3 FA97: B0 C3 FA47: B0 F2 FAA3: CF F2 FAA3: CF F3 FAA6: CF F3 FAA7: CF FAA7 FAA7: A2 C5 FAA8: BD FC FAA8: CA FA FAB1: CA FAB1: CA FAB2: A7 C5 FAB3: B0 F7 FAB4: A7 C8 FAB2: CA FA FAB2: CA FA FAB2: CA C1 FAB2: CA FA FAB2: CA C1 FAC2: F0 C7	570 572 572 574 575 575 575 575 575 577 578 577 578 577 600 577 600 500 602 602 602 600 600 600 600 600 600 6	BNE LDA CMP IXSEV LDY JMP OFIX JMP ************************************	NDFIX #*EO SDFTEV+1 NDFIX #3 SDFTEV BASIC (SDFTEV) *** APPLEII *5 PWRCON-1,X BRKV-1,X SETPLP #*CC LOCC LOC1 UCC1 #*CC LOC1 UCC1 UCC1 UCC1 UCC1 UCC1 UCC1 UCC1 U	<pre> YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC YES SO REENTER SYSTEM NO SO POINT AT WARM START FOR NEXT RESET AND DO THE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS WITH CNTRL B ADRS OF CURRENT BASIC LOAD HI SLOT +1 SETPG3 MUST RETURN X=0 SET PIR H Y IS BYTE PTR AT LAST SLOT YET? YES AND IT CAN'T BE A DISK FETCH A SLOT BYTE IS IT A DISK ?? </pre>
FABF: AD F2 03 FA92: DO OF FAA3 FA94: A9 EO FAA3 FA94: CD F3 03 FA97: DD 08 FAA3 FA96: CD F3 03 FA97: DD 08 FAA3 FA97: CD 08 FAA3 FA97: A0 03 FA78 FAA0: 4C 00 EO FAA2: AC 05 FAA3 FAA2: AC 00 FA FAA2: DD F7 FAA8 FAB2: BD F7 FAA8 FAB2: AC CB F8 FAB2: AD C7 FAA8 FAB2: AD C7 FAA8 FAB2: AD C7 FAA8 FAB2: AD C7 FAA8 FAB2: AD C7	570 571 572 573 574 575 575 575 602 603 604 601 577 602 603 604 602 603 604 605 604 605 604 605 607 611 611 6118	BNE LDA CMP IXSEV LDY STY JMP OFIX JMP THENE ETPG3 EQU ETPC3 EQU ETPLP LDA STA BNE LDA STA LDOP LDY DEX STA STA STA STA STA BEQ BEQ STA CMP STA BEQ BEQ STA STA STA STA STA STA STA STA STA STA	NOFIX ##E0 SOFTEV+1 NOFIX BASIC (SOFTEV) BASIC (SOFTEV) ** APPLEII * #5 BRKV-1,X SETPLP H%CON-1,X SETPLP H%CC LOCO LOCO LOCO LOCO CI LOCO CI LOCO FIXSEV MSLOT (LOCO),Y	<pre> YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC YES SO REENTER SYSTEM NO SO POINT AT WARN START FOR NEXT RESET AND DO THE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS HITH CNTRL B ADRS OF CURRENT BASIC LOAD HI SLOT +1 SETPG3 MUST RETURN X=0 SET PTR H Y IS BYTE PTR AT LAST SLOT YET? YES AND IT CAN'T BE A DISK FETCH A SLOT BYTE </pre>
FABF: AD F2 03 FA92: D0 OF FAA3 FA94: A7 E0 FAA3 FA94: CD F3 FAA3 FA95: CD F3 G3 FA97: D0 08 FAA3 FA97: BC F2 G3 FAA3 FAA3 FAA3 FAA3 EC F2 G3 FAA3 BC F2 G3 FAA3 CC F2 G3 FAA3 CC F2 G3 FAA3 CC F2 G3 FAA42 C FAA7 FAA7 FAA7 A2 C5 FAA7 FAA2 A2 C5 FAA8 FAA2 A2 C5 FAA8 FAB2 A2 D F7 FAB2 A47 CE FA FAB2 A2 D F7 FAB2 A5 DE F FAB2 A0 07 F FAB2 A0 07 F	5701 5725 5725 5725 5725 5725 5725 5725 572	BNE LDA CMP IXSEV LDY JMP OFIX JMP ************************************	NDFIX #*EO SDFTEV+1 NDFIX #3 SDFTEV BASIC (SDFTEV) *** APPLEII *5 PWRCON-1,X BRKV-1,X SETPLP #*CC LOCC LOC1 UCC1 #*CC LOC1 UCC1 UCC1 UCC1 UCC1 UCC1 UCC1 UCC1 U	<pre> YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC YES SO REENTER SYSTEM NO SO POINT AT WARM START FOR NEXT RESET AND DO THE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS WITH CNTRL B ADRS OF CURRENT BASIC LOAD HI SLOT +1 SETPG3 MUST RETURN X=0 SET PTR H Y IS BYTE PTR AT LAST SLOT YET? YES AND IT CAN'T BE A DISK FETCH A SLOT BYTE NO, SO NEXT SLOT DOWN </pre>
FABF: AD F2 03 FA92: DO OF FAA3 FA94: A9 EO FAA3 FA94: CD F3 G3 FA97: DD 08 FAA3 FA96: CD F3 03 FA97: DD 08 FAA3 FAA9: AC 02 G3 FAA9: AC 02 G4 FAA3: AC F2 03 FAA6: AC 00 E0 FAA3: AC F2 03 FAA4: A2 06 F3 FAA8: AD FC FAA9 FAA8: AD FC FA FAB2: AD F7 FAA8 FAB2: AD F7 FAA8 FAB2: AD F7 FAA8 FAB2: AD F7 FAA8 FAB2: AD CF FAA8 FAB2: AD </td <td>5701 5771 5773 5773 5775 5775 5775 5775 577</td> <td>BNE LDA CMP IXSEV LDY JMP JMP OFFIX JMP F************************************</td> <td>NOFIX #*E0 SOFTEV+1 NOFIX SOFTEV BASIC (SOFTEV) ** APPLEII * * SETPLP HRCON-1,X BRKV-1,X SETPLP LOC1 LOC1 LOC1 COC1 * COC1 * COC1 SLOC0 Y SLOOP SLOOP SLOOP</td> <td><pre> YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC YES SO REENTER SYSTEM NO SO POINT AT WARM START FOR NEXT RESET AND DO THE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS WITH CNTRL B ADRS OF CURRENT BASIC LOAD HI SLOT +1 SETPG3 MUST RETURN X=0 SET PTR H Y IS BYTE PTR AT LAST SLOT YET? YES AND IT CAN'T BE A DISK FETCH A SLOT BYTE NO, SO NEXT SLOT DOWN </pre></td>	5701 5771 5773 5773 5775 5775 5775 5775 577	BNE LDA CMP IXSEV LDY JMP JMP OFFIX JMP F************************************	NOFIX #*E0 SOFTEV+1 NOFIX SOFTEV BASIC (SOFTEV) ** APPLEII * * SETPLP HRCON-1,X BRKV-1,X SETPLP LOC1 LOC1 LOC1 COC1 * COC1 * COC1 SLOC0 Y SLOOP SLOOP SLOOP	<pre> YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC YES SO REENTER SYSTEM NO SO POINT AT WARM START FOR NEXT RESET AND DO THE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS WITH CNTRL B ADRS OF CURRENT BASIC LOAD HI SLOT +1 SETPG3 MUST RETURN X=0 SET PTR H Y IS BYTE PTR AT LAST SLOT YET? YES AND IT CAN'T BE A DISK FETCH A SLOT BYTE NO, SO NEXT SLOT DOWN </pre>
FABF: AD F2 03 FA92: DO OF FAA3 FA94: A9 EO FAA3 FA94: CD F3 G3 FA97: DD 08 FAA3 FA96: CD F3 03 FA97: DD 08 FAA3 FAA9: AC 02 G3 FAA9: AC 02 G4 FAA3: AC F2 03 FAA6: AC 00 E0 FAA3: AC F2 03 FAA4: A2 06 F3 FAA8: AD FC FAA9 FAA8: AD FC FA FAB2: AD F7 FAA8 FAB2: AD F7 FAA8 FAB2: AD F7 FAA8 FAB2: AD F7 FAA8 FAB2: AD CF FAA8 FAB2: AD </td <td>570 571 572 573 574 575 574 575 577 577 577 577 577 577</td> <td>BNE LDA CMP STY STY DFIX JMP DFIX JMP DFIX JMP DFIX JMP ETPG3 EQU ETPG3 EQU ETPC3 EQU ETPC9 LDA STA BNE LDA STX STA DEX DEX STX STA CMP BNE BEQ BEQ BEQ BEQ BEQ BEQ BEQ BPL BPL</td> <td>NDFIX #*EO SDFTEV+1 NDFIX #3 SDFTEV BASIC (SDFTEV) *** APPLEII *5 PWRCON-1, X BRKV-1, X BRKV-1, X SETPLP #5 LOCC1 LOCC1 LOCC1 LOCC1 LOCC1 LOCC1 LOCC1 LOCC1 H*CO LDCC1 LOCC1 LOCC1 LOCC1 LOCC1 VSLOOP NXTBYT</td> <td><pre> YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC YES SO REENTER SYSTEM NO SO POINT AT WARN START FOR NEXT RESET AND DO IHE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS WITH CNTRL B ADRS OF CURRENT BASIC LOAD HI SLOT +1 SETP3 MUST RETURN X=0 SET PIR H Y IS BYTE PTR AT LAST SLOT YET? YES AND IT CAN'T BE A DISK FETCH A SLOT BYTE IS IT A DISK ?? NO, SO NEXT SLOT DOWN YES, SO CHECK NEXT BYTE UNTLL 4 BYTES CHECKED </pre></td>	570 571 572 573 574 575 574 575 577 577 577 577 577 577	BNE LDA CMP STY STY DFIX JMP DFIX JMP DFIX JMP DFIX JMP ETPG3 EQU ETPG3 EQU ETPC3 EQU ETPC9 LDA STA BNE LDA STX STA DEX DEX STX STA CMP BNE BEQ BEQ BEQ BEQ BEQ BEQ BEQ BPL BPL	NDFIX #*EO SDFTEV+1 NDFIX #3 SDFTEV BASIC (SDFTEV) *** APPLEII *5 PWRCON-1, X BRKV-1, X BRKV-1, X SETPLP #5 LOCC1 LOCC1 LOCC1 LOCC1 LOCC1 LOCC1 LOCC1 LOCC1 H*CO LDCC1 LOCC1 LOCC1 LOCC1 LOCC1 VSLOOP NXTBYT	<pre> YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC YES SO REENTER SYSTEM NO SO POINT AT WARN START FOR NEXT RESET AND DO IHE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS WITH CNTRL B ADRS OF CURRENT BASIC LOAD HI SLOT +1 SETP3 MUST RETURN X=0 SET PIR H Y IS BYTE PTR AT LAST SLOT YET? YES AND IT CAN'T BE A DISK FETCH A SLOT BYTE IS IT A DISK ?? NO, SO NEXT SLOT DOWN YES, SO CHECK NEXT BYTE UNTLL 4 BYTES CHECKED </pre>
FABF: AD F2 03 FA92: D0 OF FAA3 FA94: A9 E0 FAA3 FA94: A9 E0 FAA3 FA97: D0 08 FAA3 FA97: B0 08 FAA3 FA97: B0 08 FAA3 FA47: BC F2 03 FAA3: A0 FAA3: A0 03 FAA3 FAA3: A0 03 FAA3 FAA3: A0 04 FAA3 FAA3: A0 C0 FB FAA4: A2 05 FAA4 FAA2: A2 D FAA5 FAA2: A2 D FAA5 FAA2: A7 CB FAA4 FAB2: CA FAA5 FAA5 FAB2: A7 CB FA FAB2: A5 01 FAB2: A5 FAB2: A5 01 FAA5 FAC: B FB FA7 FAC: B FB FA7 FAC: B FB FA7 FAB2: A5 D1 FAB2 FAB2: A5 FA7 FA9 FAC2: FD<	5701 5771 5773 5773 5775 5775 5775 5775 577	BNE LDA CMP IXSEV LDY JMP OFIX JMP F************************************	NOFIX #*E0 SOFTEV+1 NOFIX SOFTEV BASIC (SOFTEV) ** APPLEII * * SETPLP HRCON-1,X BRKV-1,X SETPLP LOC1 LOC1 LOC1 COC1 * COC1 * COC1 SLOC0 Y SLOOP SLOOP SLOOP	<pre> YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC YES SO REENTER SYSTEM NO SO POINT AT WARM START FOR NEXT RESET AND DO THE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS WITH CNTRL B ADRS OF CURRENT BASIC LOAD HI SLOT +1 SETPG3 MUST RETURN X=0 SET PTR H Y IS BYTE PTR AT LAST SLOT YET? YES AND IT CAN'T BE A DISK FETCH A SLOT BYTE NO, SO NEXT SLOT DOWN </pre>
FABF: AD F2 03 FA92: D0 OF FAA3 FA94: A9 E0 FAA3 FA94: CD F3 FAA3 FA97: D0 OB FAA3 FAA9: CD O3 FAA3 FAA3: CC F2 03 FAA4: A0 O6 FAA5 FAA2: A0 O6 FB FAA4: A0 O7 FAA5 FAA2: D0 F7 FAA8 FAB2: D0 F7 FAA8 FAB2: D0 F7 FAA8 FAB2: A0 O7 FAB5 FAC2: C0 D7 FA98 FAC2: C0 D7 FA98 FAC2: C0 D7 FA98 FAC2: C0 C FA98 FAC7: B1 O0 FA98 FAC5: CB C FA08 FAC5: CB FAC7 FA04 <t< td=""><td>570 571 572 573 574 575 575 575 575 575 575 575 575 575</td><td>BNE LDA CMP IXSEV LDY JMP DFIX JMP DFIX JMP DFIX JMP ETPG3 EQU ETPC3 EQU ETPC9 LDA STA DEX BNE LDA STA DEX STA STA DEX DEX DEX DEX DEX DEX DEX DEX DEX DEX</td><td>NDFIX #*EO SDFTEV+1 NDFIX #3 SDFTEV BASIC (SDFTEV) *** APPLEII *5 PWRCON-1, X BRKV-1, X BRKV-1, X SETPLP #5 LOCC1 LOCC1 LOCC1 LOCC1 LOCC1 LOCC1 LOCC1 LOCC1 H*CO LDCC1 LOCC1 LOCC1 LOCC1 LOCC1 VSLOOP NXTBYT</td><td><pre> YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC YES SO REENTER SYSTEM NO SO POINT AT WARN START FOR NEXT RESET AND DO IHE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS WITH CNTRL B ADRS OF CURRENT BASIC LOAD HI SLOT +1 SETP3 MUST RETURN X=0 SET PIR H Y IS BYTE PTR AT LAST SLOT YET? YES AND IT CAN'T BE A DISK FETCH A SLOT BYTE IS IT A DISK ?? NO, SO NEXT SLOT DOWN YES, SO CHECK NEXT BYTE UNTLL 4 BYTES CHECKED </pre></td></t<>	570 571 572 573 574 575 575 575 575 575 575 575 575 575	BNE LDA CMP IXSEV LDY JMP DFIX JMP DFIX JMP DFIX JMP ETPG3 EQU ETPC3 EQU ETPC9 LDA STA DEX BNE LDA STA DEX STA STA DEX DEX DEX DEX DEX DEX DEX DEX DEX DEX	NDFIX #*EO SDFTEV+1 NDFIX #3 SDFTEV BASIC (SDFTEV) *** APPLEII *5 PWRCON-1, X BRKV-1, X BRKV-1, X SETPLP #5 LOCC1 LOCC1 LOCC1 LOCC1 LOCC1 LOCC1 LOCC1 LOCC1 H*CO LDCC1 LOCC1 LOCC1 LOCC1 LOCC1 VSLOOP NXTBYT	<pre> YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC YES SO REENTER SYSTEM NO SO POINT AT WARN START FOR NEXT RESET AND DO IHE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS WITH CNTRL B ADRS OF CURRENT BASIC LOAD HI SLOT +1 SETP3 MUST RETURN X=0 SET PIR H Y IS BYTE PTR AT LAST SLOT YET? YES AND IT CAN'T BE A DISK FETCH A SLOT BYTE IS IT A DISK ?? NO, SO NEXT SLOT DOWN YES, SO CHECK NEXT BYTE UNTLL 4 BYTES CHECKED </pre>
FABF: AD F2 03 FA92: D0 OF FAA3 FA94: A9 E0 FAA3 FA97: D0 OB FAA3 FA97: D0 OB FAA3 FA97: D0 OB FAA3 FA97: D0 OB FAA3 FA47: EC F2 03 FAA3: C0 FAA3: A0 C3 FAA3 FAA3: A0 C3 FAA3 FAA3: A0 C4 FAA3 FAA3: A0 C5 FAA4 FAA4: A2 C5 FAA9 FAA2: A2 D5 FAA4 FAA2: A2 C5 FAA8 FAB2: A4 C8 FAB4: A7 FAB2: A9 DEF O3 FAB2: A4 C8 FAB5 FAB2: A5 O1 FAB2: A5 FAB2: A5 D1 FAB2: A5 FAC: B F80 FAC FAC: B F80 FAC FAC: B F80 FAC FAC: B FAC FAC FAC: B FAD5 FAC7 <	570 571 572 573 574 575 574 575 577 577 577 600 577 600 577 600 500 601 602 603 604 6005 6005 6005 6005 6005 6005 6005	BNE LDA CMP IXSEV LDY JMP OF IX JMP F************************************	NOFIX #*E0 SOFTEV+1 NOFIX ** APPLEII * ** APPLEII * * * SETPLP H%C0N-1,X BRKV-1,X SETPLP L0C1 L0C1 L0C1 L0C1 L0C1 L0C1 L0C1 L0C1	<pre> YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC YES SO REENTER SYSTEM NO SO POINT AT WARN START FOR NEXT RESET AND DO IHE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS WITH CNTRL B ADRS OF CURRENT BASIC LOAD HI SLOT +1 SETP3 MUST RETURN X=0 SET PIR H Y IS BYTE PTR AT LAST SLOT YET? YES AND IT CAN'T BE A DISK FETCH A SLOT BYTE IS IT A DISK ?? NO, SO NEXT SLOT DOWN YES, SO CHECK NEXT BYTE UNTLL 4 BYTES CHECKED </pre>
FABF: AD F2 03 FA92: DO OF FAA3 FA94: A9 EO FAA3 FA94: CD F3 FAA3 FA95: CD F3 G3 FA97: DO 08 FAA3 FA97: CD F2 03 FA97: DO 08 FAA3 FA97: CD 60 EO FAA3: AC F2 03 FAA3: AC F2 03 FAA4: AC 60 EO FAA4: A2 05 FAA5 FAA2: DD F7 FAA8 FAA2: DD F7 FAA8 FAB2: DD F7 FAA8 FAB2: DD F7 FAA8 FAB2: AD CF FAA8 FAB2: AD F7 FAA8 FAB2: AD F7 FAA8 FAB2: AD F7 FAA8 FAB2: AD CF FAA8 FAB2: AD CF FAB8 FAB2: AD CF FAA9 FAB2: AD CF FAF FAC2:	5701 5715 5725 573575 5745 F: 5745 F: 5745 F: 5745 F: 5745 F: 5747 F:	BNE LDA CMP IXSEV LDY JMP OFIX JMP OFIX JMP OFIX JMP ETPG3 EQU ETPC3 EQU ETPC9 LDA STA DEX BNE LDA STA DEX DEX DEX DEX DEX DEX DEX DEX DEX DEX	NDFIX #*E0 SDFTEV+1 NDFIX #3 SDFTEV BASIC (SDFTEV) *** APPLEII *5 PWRCON-1, X BRKV-1, X BRKV-1, X SETPLP #5 LDCC1 LDC1 LDC1 LDC1 LDC1 LDC1 LDC1 LDC	<pre> YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC YES SO REENTER SYSTEM NO SO POINT AT WARN START FOR NEXT RESET AND DO IHE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS WITH CNTRL B ADRS OF CURRENT BASIC LOAD HI SLOT +1 SETPG3 MUST RETURN X=0 SET PTR H Y IS BYTE PTR AT LAST SLOT YET? YES AND IT CAN'T BE A DISK FETCH A SLOT BYTE IS IT A DISK ?? NO, SO NEXT SLOT DOWN YES, SO CHECK NEXT BYTE UNTIL 4 BYTES CHECKED YOB BOOT </pre>
FABF: AD F2 03 FA92: D0 OF FAA3 FA94: A9 EO FAA3 FA97: D0 OB FAA3 FA97: D0 OB FAA3 FA97: D0 OB FAA3 FA97: D0 CB FAA3 FA47: BC F2 O3 FAA2: AC OD CD FAA3: AC OD FAA3 FAA3: AC OD FAA3 FAA3: AC OD FAA4 FAA4: C0 FAA7 FAA7 FAA7: AC OD FAA7 FAA2: AC CD FAA7 FAA2: AC CD FAA7 FAA2: AC CD FAA7 FAA2: AC CD FAA8 FAA2: AC<	570 571 572 573 574 574 574 575 574 575 574 575 574 600 FL 600 500 600 500 600 500 600 500 600 500 600 500 5	BNE LDA CMP IXSEV LDY JMP OF IX JMP OF IX JMP TETPG3 EGU ETPC3 EGU ETPLP LDA STA BNE LDA STA LDDP LDY BEQ STA CMP BEG STA CMP BEG DEY BNE DEY DEY DEY DEY DEY DEY DEY DEY DEY DE	NOFIX #*E0 SOFTEV+1 NOFIX ** SOFTEV BASIC (SOFTEV) ** APPLEII * * * SETPLP H*C0 LOC1 LOC1 LOC1 LOC1 LOC1 LOC1 LOC1 LOC1	<pre> YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC YES SO REENTER SYSTEM ND SO POINT AT WARN START FOR NEXT RESET AND DO THE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS WITH CNTRL B ADRS OF CURRENT BASIC LOAD HI SLOT +1 SETPG3 MUST RETURN X=0 SET PAR AT LAST SLOT YET? YES AND IT CAN'T BE A DISK FETCH A SLOT BYTE IS IT A DISK ?? NO, SO NEXT SLOT DOWN YES, SO CHECK NEXT BYTE UNTIL 4 BYTES CHECKED GO BODT / DISPLAY USER REG CONTENTS </pre>
FABF: AD F2 03 FA92: D0 OF FAA3 FA94: A9 E0 FAA3 FA94: C0 F3 03 FAA3 FA95: C0 F3 03 FAA3 FA97: BC F2 03 FAA3 FAA9: C0 60 E0 FAA3: C6 F2 03 FAA3 FAA3: AC F2 03 FAA3 FAA4: A0 06 FAA3 FAA4: A0 G0 FAA9 FAA4: A0 FAA9 FAA9 FAA2: D0 F7 FAA8 FAA2: D0 F7 FAA8 FAA2: A9 C6 FA FAB2: D0 F7 FAA8 FAB2: C0 F7 FAA8 FAB2: A0 C7 FAB2 FAB2: A0 C7 FAB2 FAB2: A0 C7 FAB3	5701 5771 5772 5773 5775 5775 5775 5775 5775 5775	BNE LDA CMP IXSEV LDY JMP OFIX JMP OFIX JMP DFIX JMP CFIX JMP CFIX LDY LDY ETPC3 EQU ETPC9 LDA STA BNE LDA STA LDJP LDY DEX LDA CMP DEX LDA STA STA STA CMP DEX LDA DEX DEX DEX DEX DEX DEX DEX DEX DEX DEX	NOFIX #*E0 SOFTEV+1 NOFIX #3 SOFTEV BASIC (SOFTEV) ** APPLEII * * * PWRCON-1, X BRKV-1, X SETPLP #*C8 LOC0 LOC1 LOC1 LOC1 LOC1 C1 * * * * * * * * * * * * * * * * *	<pre> YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC YES SO REENTER SYSTEM NO SO POINT AT WARN START FOR NEXT RESET AND DO IHE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS WITH CNTRL B ADRS OF CURRENT BASIC LOAD HI SLOT +1 SETPG3 MUST RETURN X=0 SET PTR H Y IS BYTE PTR AT LAST SLOT YET? YES AND IT CAN'T BE A DISK FETCH A SLOT BYTE IS IT A DISK ?? NO, SO NEXT SLOT DOWN YES, SO CHECK NEXT BYTE UNTIL 4 BYTES CHECKED YOB BOOT </pre>
FABF: AD F2 03 FA92: D0 OF FAA3 FA94: A9 EO FAA3 FA97: D0 OB FAA3 FA97: D0 OB FAA3 FA97: D0 OB FAA3 FA97: D0 CB FAA3 FA47: BC F2 O3 FAA2: AC OD CD FAA3: AC OD FAA3 FAA3: AC OD FAA3 FAA3: AC OD FAA4 FAA4: C0 FAA4 FAA7 FAA7: AC OD FAA7 FAA2: AC OD FAA4 FAA2: AC OD FAA4 FAA2: AC OD FAA7 FAA2: AC CD FAA7 FAA2: AC CD FAA7 FAA2: AC CD FAA7 FAA2: AC CD FAA8 FAA2: AC<	570 571 572 573 574 574 574 575 574 575 574 575 574 600 FL 600 500 600 500 600 500 600 500 600 500 600 500 5	BNE LDA CMP IXSEV LDY JMP OF IX JMP OF IX JMP TETPG3 EGU ETPC3 EGU ETPLP LDA STA BNE LDA STA LDDP LDY BEQ STA CMP BEG STA CMP BEG DEY BNE DEY DEY DEY DEY DEY DEY DEY DEY DEY DE	NOFIX #*E0 SOFTEV+1 NOFIX ** SOFTEV BASIC (SOFTEV) ** APPLEII * * * SETPLP H*C0 LOC1 LOC1 LOC1 LOC1 LOC1 LOC1 LOC1 LOC1	<pre> YES SEE IF COLD START HAS BEEN DONE YEY? DOES SOFT ENTRY VECTOR POINT AT BASIC YES SO REENTER SYSTEM ND SO POINT AT WARN START FOR NEXT RESET AND DO THE COLD START SOFT ENTRY VECTOR SET PAGE 3 VECTORS WITH CNTRL B ADRS OF CURRENT BASIC LOAD HI SLOT +1 SETPG3 MUST RETURN X=0 SET PAR AT LAST SLOT YET? YES AND IT CAN'T BE A DISK FETCH A SLOT BYTE IS IT A DISK ?? NO, SO NEXT SLOT DOWN YES, SO CHECK NEXT BYTE UNTIL 4 BYTES CHECKED GO BODT / DISPLAY USER REG CONTENTS </pre>

FADE: A9 00	629		LDA	#\$00	
FAE0: 85 41	630		STA	АЗН	
FAE2: A2 FB	631		LDX	#\$FB	
FAE4: A9 A0		RDSP1	LDA	#\$A0	
FAE6:20 ED FD	633		JSR	COUT	
FAE9: BD 1E FA	634		LDA	RTBL-251, X	
FAEC: 20 ED FD	635		JSR	COUT	
FAEF: A9 BD	636		LDA	#\$BD	
FAF1:20 ED FD	637		JSR	COUT	
FAF4:		* LDA ACC+5,			
FAF4: 85 4A	639		DFB	\$B5,\$4A	
FAF6: 20 DA FD	640		JSR	PRBYTE	
FAF9: EB	641		INX		
FAFA: 30 E8 FAE4	642		BM1	RDSP1	
FAFC: 60	643		RTS		
FAFD: 59 FA	644	PWRCON	DW	OLDBRK	
FAFF: 00 E0 45	645		DFB	\$00, \$E0, \$45	
FB02: 20 FF 00 FF	646	DISKID	DFB	\$20, \$FF, \$00, \$	FF
FB06:03 FF 3C	647		DFB	\$03, \$FF, \$3C	
FB09:C1 F0 F0 EC	648	TITLE	ASC	'Apple]	IE'
FB11: FB11	649	XLTBL	EQU	*	
FB11:C4 C2 C1	650		DFB	\$C4, \$C2, \$C1	
EB14: EE C3	651		DFB	\$FF, \$C3	
FB16: FF FF FF	652		DFB	\$FF, \$FF, \$FF	
FB19:		* MUST DRG	\$FB19		
FB19:C1 D8 D9	654	RTBL	DFB	\$C1, \$D8, \$D9	REGISTER NAMES FOR REGDSP:
FB1C: DO D3	655	N. T.D.C.	DFB	\$DO, \$D3	; 'AXYPS'
FB1E: AD 70 CO	654	PREAD	LDA	PTRIG	TRIGGER PADDLES
F821: A0 00	657		LDY		INIT COUNT
FB23: EA	658		NOP		COMPENSATE FOR 1ST COUNT
FB24: EA	659		NOP		Your Erente For for oom
FB24:EA FB25:BD 64 CO		PREAD2	LDA	PADDLO, X	COUNT Y-REG EVERY 12 USEC.
	661	TREMUE	BPI	RTS2D	COULT I THEY EVENT IE COLO.
FB28:10 04 FB2E	661		INY	N (38)	
FB2A: CB				PREAD2	EXIT AT 255 MAX
FB2B: DO F8 FB25	663		BNE	PREAUZ	JEXII AL 255 MAX
FB2D: 88	664		DEY		
FB2E: 60		RTS2D	RTS		
FB2F:	666		CHN	BJS. SRC2	
FB2F:	1	*			
FB2F: A9 00	2	INIT	LDA	#\$00	CLR STATUS FOR DEBUG SOFTWARE
FB31:85 48	З		STA	STATUS	
FB33: AD 56 CO	4		LDA	LORES	
FB36: AD 54 CO	5		LDA		; INIT VIDEO MODE
FB39: AD 51 CO	6	SETTXT	LDA	TXTSET	SET FOR TEXT MODE
FB3C: A9 00	7		LDA	#\$00	FULL SCREEN WINDOW
FB3E: FO OB FB4B	8		BEQ	SETWND	
FB40: AD 50 CO	9	SETGR	LDA	TXTCLR	SET FOR GRAPHICS MODE
FB43: AD 53 CO	10		LDA	MIXSET	LOWER 4 LINES AS TEXT WINDOW
FB46:20 36 FB	11		JSR	CLRTOP	
FB49: A9 14	12		LDA	#\$14	
FB4B: 85 22	13	SETWND	STA	WNDTOP	SET FOR 40 COL WINDOW
FB4D: A9 00	14		LDA	#\$00	; TOP IN A-REG,
FB4F: 85 20	15		STA	WNDLFT	; BOTTOM AT LINE \$24
FB51: 0001	16		DO	APPLE2E	; /RRA0981
FB51: A0 08	17		LDY	#8	CODE=SETWND /RRA0981
FB53: DO 5F FBB4	18		BNE		D0 40/80 /RRA0981
FB55:	19		ELSE	001001	; /RRA0981
S	20		LDA	#\$28	
s	21		STA	WNDWDTH	
FB55:	22		FIN	HILDING III	/RRA0981
FB55: A9 18	23		LDA	#\$18	//////////
FB57:85 23	24		STA	WNDBTM	
FB59: A9 17	25		LDA	#\$17	VTAB TO ROW 23
FB5B:85 25		TABV	STA	##1/ CV	VTABS TO ROW IN A-REG
FB5D: 4C 22 FC	27		JMP	VTAB	A THE TO NOT AN EINED
FB60:20 58 FC		APPLEII	JSR		
		EN CHEAT			CLEAR THE SCRN
CD12. 10 00			INV		CLEAR THE SCRN
FB63: A0 08	29	STITLE	LDY	#8	
FB65: B9 08 FB	29 30	STITLE	LDA	#8 TITLE-1, Y	GET A CHAR
FB65:89 08 FB FB68:99 0E 04	29 30 31	STITLE	LDA STA	#8	
FB65:89 08 FB FB68:99 0E 04 FB6B:88	29 30 31 32	STITLE	LDA STA DEY	#8 TITLE-1, Y LINE1+14, Y	GET A CHAR
FB65: B9 08 FB FB68: 99 0E 04 FB6B: 88 FB6C: D0 F7 FB65	29 30 31 32 33	STITLE	LDA STA DEY BNE	#8 TITLE-1, Y	GET A CHAR
FB65: B9 08 FB FB68: 99 0E 04 FB6B: 88 FB6C: D0 F7 FB65 FB6E: 60	29 30 31 32 33 34		LDA STA DEY BNE RTS	#8 TITLE-1, Y LINE1+14, Y STITLE	;GET A CHAR ;PUT IT AT TOP CENTER OF SCREEN
FB65: B9 08 FB FB6B: 99 0E 04 FB6B: 88 FB6C: D0 F7 FB65 FB6E: 60 FB6F: AD F3 03	29 30 31 32 33 34 35	STITLE	LDA STA DEY BNE RTS LDA	#8 TITLE-1, Y LINE1+14, Y STITLE	;GET A CHAR ;PUT IT AT TOP CENTER OF SCREEN
FB65: B9 08 FB FB66: 99 0E 04 FB6B: 88 FB66: D0 F7 FB65 FB6E: 60 FB6F: AD F3 03 FB72: 49 A5	29 30 31 32 33 34 35 36		LDA STA DEY BNE RTS LDA EOR	#8 TITLE-1, Y LINE1+14, Y STITLE SOFTEV+1 #\$A5	GET A CHAR
FB65: B9 08 FB FB68: 99 0E 04 FB68: 88 FB66: D0 F7 FB65 FB6E: 60 FB6F: AD F3 03 FB72: 49 A5 FB74: 8D F4 03	29 30 31 32 33 34 35 36 37		LDA STA DEY BNE RTS LDA EOR STA	#8 TITLE-1, Y LINE1+14, Y STITLE	;GET A CHAR ;PUT IT AT TOP CENTER OF SCREEN
FB65: B9 OB FB FB68: 99 OE 04 FB68: 88 FB6C: D0 F7 FB65 FB66: 60 FB67: AD F3 03 FB72: 49 A5 FB74: 6D F4 03 FB77: 60	29 30 31 32 33 34 35 36 37 38	SETPWRC	LDA STA DEY BNE RTS LDA EOR STA RTS	#8 TITLE-1, Y LINE1+14, Y STITLE SOFTEV+1 #\$A5	;GET A CHAR ;PUT IT AT TOP CENTER OF SCREEN ;ROUTINE TO CALCULATE THE 'FUNNY ;COMPLEMENT' FOR THE RESET VECTOR
FB65:B9 08 FB FB68:99 06 04 FB68:88 FB65:60 F7 FB65 FB65:60 F7 FB65 FB65 FB72:47 A5 FB74:49 A5 FB74:8D F4 03 FB77:40 F8 FB75:77:60 F878 F878 F878 F878	29 30 31 32 33 34 35 36 37 38 37		LDA STA DEY BNE RTS LDA EOR STA RTS EQU	#8 TITLE-1,Y LINE1+14,Y STITLE SOFTEV+1 #\$A5 PWREDUP *	; GET A CHAR ; PUT IT AT TOP CENTER OF SCREEN ; ROUTINE TO CALCULATE THE 'FUNNY ; COMPLEMENT' FOR THE RESET VECTOR ; CHECK FOR A PAUSE (CONTROL-S).
FB65: BP OB FB FB68: 9P OE O4 FB68: 8B FB62: D0 F7 FB65 FB62: 6D F7 FB65 FB72: 4P A5 FB72: 4P A5 FB77: 60 FB77: 60 FB78: FB78 FB78: C9 BD	29 30 31 32 33 34 35 36 37 38 37 38 39 40	SETPWRC	LDA STA DEY BNE RTS LDA EOR STA RTS EQU CMP	#8 TITLE-1,Y LINE1+14,Y STITLE SOFTEV+1 #\$A5 PWREDUP * *	; GET A CHAR ; PUT IT AT TOP CENTER OF SCREEN ; ROUTINE TO CALCULATE THE 'FUNNY ; COMPLEMENT' FOR THE RESET VECTOR ; CHECK FOR A PAUSE (CONTROL-S). ; ONLY WHEN I HAVE A CR
FB65:B7 08 FB FB68:97 00 04 FB66:B0 F7 FB65 FB65:60 F7 FB65 FB65:60 03 FB74:90 F3 03 FB74:90 F4 03 FB77:60 FB78: FB78 FB78:C7 80 FB74:B0 F8 FB74:B0 F8 FB74:B0 F8 FB74:B0 F8 FB74:B0 F8 FB74:B0 F8 F874:B0 F8 F874:B0 F8 F874:B0 F8 F874:B0 F8 F874:B0 F8 F874:B0 F8 F874:B0 F8 F874:B0 F8 F874:F874:F874 F874:F874 F8	29 30 31 32 33 34 35 36 37 38 37 38 39 40	SETPWRC	LDA STA DEY BNE RTS LDA EOR STA RTS EQU CMP BNE	#8 TITLE-1,Y LINE1+14,Y STITLE SOFTEV+1 #\$A5 PWREDUP * #\$BD NOWAIT	; GET A CHAR ; PUT IT AT TOP CENTER OF SCREEN ; ROUTINE TO CALCULATE THE 'FUNNY ; COMPLEMENT' FOR THE RESET VECTOR ; CHECK FOR A PAUSE (CONTROL-S). ; ONLSU, WHEN I HAVE A CR ; NOT SO, DO REGULAR
FB65:B7 08 FB FB68:97 00 04 FB66:B0 F7 FB65 FB65:60 F7 FB65 FB65:60 03 FB74:90 F3 03 FB74:90 F4 03 FB77:60 FB78: FB78 FB78:C7 80 FB74:B0 F8 FB74:B0 F8 FB74:B0 F8 FB74:B0 F8 FB74:B0 F8 FB74:B0 F8 F874:B0 F8 F874:B0 F8 F874:B0 F8 F874:B0 F8 F874:B0 F8 F874:B0 F8 F874:B0 F8 F874:B0 F8 F874:F874:F874 F874:F874 F8	29 30 31 32 33 34 35 36 37 38 37 38 39 40 41	SETPWRC	LDA STA DEY BNE RTS LDA EOR STA RTS EQU CMP BNE LDY	#8 TITLE-1.Y LINE1+14.Y STITLE SOFTEV+1 #\$A5 PWREDUP * #98D NOWAIT KBD	; GET A CHAR ; PUT IT AT TOP CENTER OF SCREEN ; ROUTINE TO CALCULATE THE 'FUNNY ; COMPLEMENT' FOR THE RESET VECTOR ; CHECK FOR A PAUSE (CONTROL-S). ; ONLY WHEN I HAVE A CR ; NOT SO, DO REGULAR ; IS KEY PRESSED?
FB65:B7 08 FB FB68:97 0E 04 FB68:98 FB66:00 F7 FB65 FB66:60 F3 03 FB764:40 F3 03 FB72:47 45 FB73:40 F4 03 FB77:0 03 FB78: F878 FB78:0 18 FB74 FB77:10 18 FB74 FB77:10 18 FB74 FB77:10 18 FB74	29 30 31 32 33 34 35 36 37 38 37 40 41 42 43	SETPWRC	LDA STA DEY BNE RTS LDA EDR STA RTS EQU CMP BNE LDY BPL	#8 TITLE-1.Y LINE1+14.Y STITLE SOFTEV+1 #\$A5 PWREDUP * #\$BD NOWAIT KBD NOWAIT	; GET A CHAR ; PUT IT AT TOP CENTER OF SCREEN ; ROUTINE TO CALCULATE THE 'FUNNY ; COMPLEMENT' FOR THE RESET VECTOR ; CHECK FOR A PAUSE (CONTROL-S). ; ONLY WHEN I HAVE A CR ; NOT SO. DO REGULAR ; IS KEY PRESSED? ; NO.
FB65:B7 08 FB FB68:70 0E 04 FB68:70 0E 04 FB68:08 FB67:00 F7 FB65 FB66:00 F7 FB65 FB67:40 F3 03 FB72:47 A5 FB77:60 F4 03 FB77:60 F878 FB78:07 B0 FB78:07 B0 FB78:07 B0 FB78:00 C0 FB77:10 13 F874 FB81:00 93	29 30 31 32 33 34 35 36 37 38 39 40 42 43 44	SETPWRC	LDA STA DEY BNE RTS LDA EOR STA RTS EQU CMP BPL CPY	#8 IITLE-1.Y LINE1+14.Y STITLE SOFTEV+1 #\$A5 PWREDUP * #\$8D NOWAIT KBD NOWAIT KBD NOWAIT	; GET A CHAR ; PUT IT AT TOP CENTER OF SCREEN ; ROUTINE TO CALCULATE THE 'FUNNY ; COMPLEMENT' FOR THE RESET VECTOR ; CHECK FOR A PAUSE (CONTROL-S). ; ONLY WHEN I HAVE A CR ; NOT SO, DO REGULAR ; IS KEY PRESSED? ; NO. ; YES == 15 IT CTEL=52
FB65:B7 08 FB FB68:97 0E 04 FB68:98 FB FB68:98 FB FB68:98 F7 FB68:90 F7 FB68:90 F7 FB68:90 F3 FB74:40 F3 FB74:50 F876 FB78:00 F876 FB78:00 F874 FB78:01 13 F874:F87:10 13 F874:F88:10 93 F883:100 F874	29 30 31 32 33 34 35 36 37 38 39 40 41 43 44 45	SETPWRC	LDA STA DEY BNE LDA EOR STA EOR STA EQU CMP BNE LDY BNE CPY BNE	#8 TITLE-1.Y LINE1+14.Y STITLE SOFTEV+1 #\$45 PWREDUP * #\$8D NOWAIT KBD NOWAIT #\$93 NOWAIT	; GET A CHAR ; PUT IT AT TOP CENTER OF SCREEN ; ROUTINE TO CALCULATE THE 'FUNNY ; COMPLEMENT' FOR THE RESET VECTOR ; CHECK FOR A PAUSE (CONTROL-S). ; DALLY WHEN I HAVE A CR ; NOT SO. DO REGULAR ; IS KEY PRESSED? ; NO. ; YES IS IT CTRL-S? ; NOPE IS IT CTRL-S? ; NOPE IS NOT CTRL-S? ; NOPE IS NOT CTRL-S?
FB65: B7 08 FB FB68: 97 0E 04 FB68: 88 FB62: D0 F7 FB65 FB62: D0 F7 FB65 FB72: 47 A5 FB74: 4D F3 03 FB77: 60 F8 FB77: 60 F8 FB77: 60 F8 FB78: C7 BD FB78: C7 BD FB78: C0 C0 FB77: 10 13 FB74 FB81: C0 73 FB88: 20 0F F874	29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 45	SETFWRC VIDWAIT	LDA STA DEY BNE LDA EOR STA RTS EQU CMP BNE LDY BNE LDY BNE BIT	#8 TITLE-1.Y LINE1+14.Y STITLE SOFTEV+1 #\$45 WREDUP * # #\$90 NOWAIT KBD NOWAIT KBD NOWAIT KBD NOWAIT KBDSTRB	; GET A CHAR ; PUT IT AT TOP CENTER OF SCREEN ; ROUTINE TO CALCULATE THE 'FUNNY ; COMPLEMENT' FOR THE RESET VECTOR ; CHECK FOR A PAUSE (CONTROL-S). ; ONLY WHEN I HAVE A CR ; NOT SO, DO REQULAR ; IS KEY PRESSED? ; NO. ; YES IS IT CTRL-S? ; NOPE IGNORE ; CLEAR STROBE
FB65: B7 08 FB FB68: 70 0E 04 FB68: 88 FB68: 70 0E 04 FB FB65: 60 FB68: 80 F7 FB65 FB68: 60 F7 FB73 FB77: 40 F878 FB77: 40 F878 FB77: 40 F878 FB77: 40 F878 FB78: C0 C0 F876: AC 00 C0 F876: AC 00 F874 F876: AC 00 C0 F878: C0 73 F883: C0 C0 F883: AC 00 C0 F883: C0 00	29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 44 44 44 44 44 44 44 44 47	SETPWRC	LDA STA DEYE RTS LDA EDRA EDRA EDRA EQU CMPE LDY BNE BIT LDY	#8 TITLE-1.Y LINE1+14.Y STITLE SOFTEV+1 #\$45 PWREDUP * #9 NUWAIT KBD NUWAIT KBDSTRB KBD	; GET A CHAR ; PUT IT AT TOP CENTER OF SCREEN ; ROUTINE TO CALCULATE THE 'FUNNY ; COMPLEMENT' FOR THE RESET VECTOR ; CHECK FOR A PAUSE (CONTROL-S). ; DNLT SU, DO REGULAR ; IS KEY PRESSED? ; NO. ; YES IS IT CTRL-S? ; NOPE IS NT CTRL-S? ; NOPE NTROTE
FB65:B7 08 FB FF66:T7 08 06 04 FB68:T0 F7 FB65 FB67:AD F3 03 FB72:A7 A5 FB77:AD F3 03 FB77:AD F878 F878 FB77:AD C0 C0 FB78:AC 00 C0 FB76:AC 00 C0 FB76:AC 00 C0 FB76:AC 00 C0 FB76:AC 00 F874 FB76:AC 00 C0 FB76:AC 00 F874 FB76:AC 00 F874 FB76:AC 00 C0 F876:C0 C0 F874 F883:D0 00 F874 F883:AC 00 C0	29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 45	SETFWRC VIDWAIT	LDA STA DEY RTS LDA EOR RTS EOR EOR EOR BNE BNE BPL BPL BPL	#8 TITLE-1.Y LINE1+14.Y STITLE SOFTEV+1 #\$45 WREDUP * # #\$90 NOWAIT KBD NOWAIT KBD NOWAIT KBD NOWAIT KBDSTRB	; GET A CHAR ; PUT IT AT TOP CENTER OF SCREEN ; ROUTINE TO CALCULATE THE 'FUNNY ; COMPLEMENT' FOR THE RESET VECTOR ; CHECK FOR A PAUSE (CONTROL-S). ; ONLY WHEN I HAVE A CR ; NOT SO, DO REGULAR ; IS KEY PRESSED? ; NO. ; YES IS IT CTRL-S? ; NOPE IGNORE ; CLEAR STROBE ; WAIT TOL NEXT KEY TO RESUME ; WAIT TOK KEYPRESS
FB65:B7 08 FB FB68:70 06 04 FB68:70 06 04 FB68:70 07 FB65 FB768:70 03 FB778 FB772:47 45 FB74 FB772:47 45 FB78 FB78: FB78 FB78: FB78 FB78: C FB78: C FB78: C FB78: C FB78: D FB78: C FB78: C FB83: D FB83: C FB88: C FB88: C FB88: C	29 30 31 32 34 35 36 37 38 40 41 42 43 445 445 447 48 47	SETFWRC VIDWAIT	LDA STA DEY RTS LDA EOR RTS EQUP STA CPY BPL LDY BPL LDY BPL CPY	#8 TITLE-1.Y LINE1+14.Y STITLE SOFTEV+1 #\$45 PWREDUP * #9 NUWAIT KBD NUWAIT KBD KBDWAIT KBD KBDWAIT	; GET A CHAR ; PUT IT AT TOP CENTER OF SCREEN ; ROUTINE TO CALCULATE THE 'FUNNY ; COMPLEMENT' FOR THE RESET VECTOR ; CHECK FOR A PAUSE (CONTROL-S). ; ONLY WHEN I HAVE A CR ; NOT SO. DO REGULAR ; IS KEY PRESSED? ; NO. ; YES IS IT CTRL-S? ; NOPE IGNORE ; CLEAR STROBE ; WAIT TIL NEXT KEY TO RESUME ; WAIT FOR KEYPRESS ; IS IT CONTROL-C?
FB65:B7 08 FB FF68:70 00 F0 FB68:70 00 F7 FB45:100 F7 FB45:100 F3 FB45:100 F3 FB45:100 F3 FB47:40 F3 FB77:40 F878 FB78:0 F878 F878:0 18 F876:0 C0 F877:10 13 F876:0 C0 F877:10 13 F876:0 C0 F877:10 13 F874:20 F894 F812:0 73 F843:00 OF F883:00 C0 F888:10 F894 F888:10 C0 F888:10 C8	29 30 31 32 33 34 5 36 37 38 39 40 42 43 445 445 447 48	SETFWRC VIDWAIT	LDA STA DEY RTS LDA EOR RTS EOR EOR EOR BNE BNE BPL BPL BPL	#8 TITLE-1.Y LINE1+14.Y STITLE SOFTEV+1 #\$45 PWREDUP * #\$90 NOWAIT KBD NOWAIT KBD NOWAIT KBDSTRB KBD KBDWAIT	; GET A CHAR ; PUT IT AT TOP CENTER OF SCREEN ; ROUTINE TO CALCULATE THE 'FUNNY ; COMPLEMENT' FOR THE RESET VECTOR ; CHECK FOR A PAUSE (CONTROL-S). ; ONLY WHEN I HAVE A CR ; IS KEY PRESSED? ; NOT SO, DO REGULACR ; NOT SO, DO REGULACR ; NOT SO, DO REGULACR ; NOT SO, TO RESURE ; NOT SO, TO RESURE ; NOT SO, TANDRE ; CLEAR STROBE ; WAIT TOL NEXT KEY TO RESUME ; WAIT TOK KEYPRESS ; IS IT CONTROL-C? ; YES, SO LEAVE IT
FB65:B7 08 FB FB68:70 00 F7 FB68:70 07 FB62:70 F7 FB67:40 F3 03 FB77:40 F3 03 FB77:40 F4 03 FB77:40 F4 03 FB77:40 F4 03 FB77:40 F4 03 FB77:40 F8 00 F7 FB78:00 F8 F87 FB77:10 13 F874 F817:10 13 F874 F883:00 07 F894 F883:00 07 F894 F888:10 F8 F888 F888:10 F8 F888 F888:10 C3 F888:10 C3 F888:10 F8 F888 F888:10 C3 F888:10 C3 F888:10 C3 F888:10 C4 F888:10 C3 F888:10 C4 F888:10 C5 F888:10 C5 F888:10 C5 F888:10 C4 F888:10 C5 F888:10 C5 F888:10 C5 F888:10 C5 F888:10 C5 F894:12 C10 C0	29 30 31 32 34 35 36 37 38 40 41 42 43 445 445 447 48 47	SETFWRC VIDWAIT	LDA STA DEY RTS LDA EOR RTS EQUP STA CPY BPL LDY BPL LDY BPL CPY	#8 TITLE-1.Y LINE1+14.Y STITLE SOFTEV+1 #\$45 PWREDUP * #9 NUWAIT KBD NUWAIT KBD KBDWAIT KBD KBDWAIT	; GET A CHAR ; PUT IT AT TOP CENTER OF SCREEN ; ROUTINE TO CALCULATE THE 'FUNNY ; COMPLEMENT' FOR THE RESET VECTOR ; CHECK FOR A PAUSE (CONTROL-S). ; DNLT SU, DO REGULAR ; IS KEY PRESSED? ; NO. ; YES IS IT CTRL-S? ; NOPE IGNORE ; CLEAR STROBE ; WAIT TIL NEXT KEY TO RESUME ; WAIT FOR KEYPRESS ; IS IT CONTROL-C? ; YES, SO LEAVE IT ; CLR STROBE
FB65:B7 08 FB FB68:70 00 F7 FB68:70 07 FB62:70 F7 FB67:40 F3 03 FB77:40 F3 03 FB77:40 F4 03 FB77:40 F4 03 FB77:40 F4 03 FB77:40 F4 03 FB77:40 F8 00 F7 FB78:00 F8 F87 FB77:10 13 F874 F817:10 13 F874 F883:00 07 F894 F883:00 07 F894 F888:10 F8 F888 F888:10 F8 F888 F888:10 C3 F888:10 C3 F888:10 F8 F888 F888:10 C3 F888:10 C3 F888:10 C3 F888:10 C4 F888:10 C3 F888:10 C4 F888:10 C5 F888:10 C5 F888:10 C5 F888:10 C4 F888:10 C5 F888:10 C5 F888:10 C5 F888:10 C5 F888:10 C5 F894:12 C10 C0	29 30 31 323 34 35 36 37 38 39 40 412 43 445 445 447 489 51	SETFWRC VIDWAIT	LDA STA DEYE RTS LDA RTS EQR BNE EDR BNE BNE BPL BPL BPL BPLY BPL BPLY BEG	#8 TITLE-1.Y LINE1+14.Y STITLE SOFTEV+1 #\$45 PWREDUP * #\$8D NOWAIT NOWAIT #\$93 NOWAIT #\$93 NOWAIT \$8DSTRB KBD KBDWAIT #\$83 NOWAIT	; GET A CHAR ; PUT IT AT TOP CENTER OF SCREEN ; ROUTINE TO CALCULATE THE 'FUNNY ; COMPLEMENT' FOR THE RESET VECTOR ; CHECK FOR A PAUSE (CONTROL-S). ; ONLY WHEN I HAVE A CR ; IS KEY PRESSED? ; NOT SO, DO REGULACR ; NOT SO, DO REGULACR ; NOPE - IS IT CTRL-S? ; NOPE - IS IT CTRL-S? ; NOPE - IS NORE ; LEAR STROBE ; WAIT TILL NEXT KEY TO RESUME ; WAIT TOR KEYPRESS ; IS IT CONTROL-C? ; YES, SO LEAVE IT ; CLR STROBE ; DO AS BEFORE
FB65:B7 08 FB FB68:70 0E 04 FB68:70 0E 04 FB68:70 0F7 FB65 FB66:100 F7 FB65 FB67:40 F3 03 FB72:60 F7 FB72:60 F7 FB73:60 F8 FB72:60 F8 FB72:60 F8 FB73:60 C0 FB75:60 C0 FB75:61 03 FB75:00 C0 FB75:10 13 FB83:20 0F FB83:20 0F FB88:10 0C FB88:20 0C	29 30 31 32 33 34 35 37 39 40 411 425 46 47 49 50 51 52	SETPWRC VIDWAIT KBDWAIT	LDA STA DEYE RTS LDA RTS LDA RTS EQMP BDT BDA BDY BDY BDY BDY BDY BDY BDY BDY BDY BDY	#8 TITLE-1.Y LINE1+14.Y STITLE SOFTEV+1 #\$A5 PWREDUP * #98D NOWAIT KBD NOWAIT KBD NOWAIT KBDSTRB KBDWAIT #\$83 NOWAIT KBDAIT KBDAIT	; GET A CHAR ; PUT IT AT TOP CENTER OF SCREEN ; ROUTINE TO CALCULATE THE 'FUNNY ; COMPLEMENT' FOR THE RESET VECTOR ; CHECK FOR A PAUSE (CONTROL-S). ; DNLT SU, DO REGULAR ; IS KEY PRESSED? ; NO. ; YES IS IT CTRL-S? ; NOPE IGNORE ; CLEAR STROBE ; WAIT TIL NEXT KEY TO RESUME ; WAIT FOR KEYPRESS ; IS IT CONTROL-C? ; YES, SO LEAVE IT ; CLR STROBE

F395 (A) C S4 JPP ESC1 ISSE CHAR AS INDEX F395 (A) S5 LDAK KLTRUCS,Y ITAMELATE ISAK TO GAD F382 (D) OOD1 S5 DAR APHLERE ITAMELATE ISAK TO GAD F382 (D) COD1 S7 USR ITAMELATE ISAK TO GAD F383 (D) CE F38 S6 CATA F383 (D) CE F38 S6 CATA F383 (D) CE F39 S6 CATA F383 (D) CE F397 S6 CC F38 F383 (D) CE F397 S6 CC F397 S6 CC F383 (D) CO F397 S6 CC F397 S0 CATA F383 COLD JAK F383 (D) CO F397 S0 CATA F380 CATA<							
PERC 10 PERC 10 <t< td=""><td>F898:4C 2C</td><td>FC</td><td></td><td></td><td>JMP</td><td>ESC1</td><td></td></t<>	F898:4C 2C	FC			JMP	ESC1	
FBF:20 20 7FB 37 JBR ESCULD ID THE CURSOR MODION FBA:20 20 1001 30 DJR APPLE2E IDT THE CURSOR MODION FBA:20 21 FD 30 DJR APPLE2E IDT THE CURSOR MODION FBA:20 21 FD 30 DJR APPLE2E INT FACADES FBA:20 20 25 CPP APPLE2E INT FACADES INT FACADES FBA:20 20 25 CCPP 46(7) ILESS THAN 11/7 INT FACADES FBA:20 20 77 ESCLD IDD NDEMAL IOD NDEMAL FBA:20 2007 73 ETRNTCKOM EGU 4001 INT FACADES FBB:20 20 77 TOCKROM EGU 4001 INT FACADES FBB:20 20 73 TOCKROM EGU 4000 INT FACADES FBB:20 2007 70 ETN TKTKOM EGU FACADES INT FACADES FBB:20 2007 70 ETN TKTKOM<				ESCNOW			
FBAC: 0001 58 DD APPLEZE IFOR FLLE ESCAPES/REA0981 FBAC: 021 FD SC IOR INAX IOR INAX IOR INAX FBAC: 021 FD SC IOR INAX IOR INAX IOR INAX FBAC: 021 FD SC ISR INAX INAX FBAC: 021 FE SC SEC IST INAX INAX FBAC: 021 FE SC SC FE INAX INAX FBAC: 021 FE SC FE SC INAX INAX FBAC: 021 FE SC SC SC IST INAX INAX FBAC: 020 IST INAX SE SC IST INAX INAX INAX FBB3: GO 73 SETEL INAX INAX INAX INAX FBB3: GO 73 SETEL INAX INAX INAX FBB3: <t< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td>TRANSLATE IJKM TO CBAD</td></t<>							TRANSLATE IJKM TO CBAD
FBAS: 20 SPR RDESC ICAL ICAL <thical< th=""> ICAL ICAL I</thical<>							FOR FULL ESCAPES/RRA0981
S 61 JSR RDKPY IABO GET NEXT FBA3: CC 6250LD IIS IVT IVT IVT FBA3: S0 EE FBA7: S0 FA7 FBA7: S0 FA7 FBA7: S0 FA7 FBA7: S0 FA7 FBA7: FBA7: FBA7: FBA7: FBA7: FBA7: FBA7: FFB1: FFFB1: FFFB1: FFFB1:	FBA2: 20 21		59		JSR		;GET IJKM, ijkm, ARROWS/RRA0981
PBAS: C C2 FIN I/PRA0981 PBAS: C C Sected FIN I/PRA0981 PBAS: C C Sected IST HIS AN "N" D PBAS: C C FIN IST HIS AN "N" D D PBAS: C C FIN VES. SD DDLD MAY P PBAS: C C FIN VES. SD DDLD MAY P PBAS: D C FIN D DD DDLD MAY PBAS: D C FIN D DD DDLD MAY PBAS: D C FIN D DD DDLD MAY PBAS: C C COS I/PRA0981 DD DDLD MAY PBAS: C COS 7 SETINTCROM EQU COS //PRA0981 PBAS: C COS 7 SETINTCROM EQU COS //PRA0981 PBAS: C COS 7 SETINTCROM EQU COS //PRA0981 PBAS: C C C COS //PRA0981 //PRA0981 PBAS: C							
FBAS: 00 CE 43 ESCNEW CMP #*CE .15 THIS AN 'M''' FBAX: 80 CF FBAY 64 CC ESCLUD 1/4 'DD MEATER - DD IT' FBAX: 90 CA FFF 64 CC ESCLUD 1/5 IS NA 'M''' FBAX: 50 CC 67 CPP #*CC IS IT AN 'M''' FBAX: 50 CC 67 CPP #*CC IS IT AN 'M''' FBAX: 50 CC 67 CPP #*CC IS IT AN 'M''' FBAX: 50 CC 67 CPP #*CC IS IT AN 'M'' FBAX: 50 CC 67 CPP #*CC IF AN 'M''' FBAX: 60 CC 73 BCTINCTXORD EQU #COOL //FRAOP81 FBAX: 60 CC 73 BCTINCTXORD EQU #COOL //FRAOP81 FBAX: 60 CC 73 BCTINCTXORD EQU #COOL //FRAOP81 FBAX: 60 CC 73 BCTINCTXORD EQU #COOL BAXENTCH/RRAOP81 FBAX: 60 CC 10 PP ISAXE MARAOP81 BAXENTCH/RRAOP81 FBAX: 60 CC 10 PP ISAXE MARAOP81 BAXENTCH/RRAOP81 F						RDKEY	
FBA9: 00 EG FEAS: 00 EG FESCULD ILESS THAN '1'? FBA8: 70 EG FFPT 65 CPP WEC9 ILESS DD DLD MAY FBA8: 70 EG FFPT 66 GEG FSCULD IND NORMAL FBB3: DC EG FSCULD IDD NORMAL FSCULD IND NORMAL FBB3: DC C006 71 SECULD IND NORMAL FBB3: DC C015 73 RECKTORMON EQU *CO15 //RAA0981 FBB3: DC C015 74 VERSION DFB *O6 //RAA0981 FBB3: DC C015 70 BIT RECKTORMON GET *RECKTORA0981 FTATE/RRA0981 FBB3: DC C015 FD14 X/OUTOCX EGT //RAA0981 FTATE/RRA0981 FBB4: DC 7C0 B1 STATE/RRA0981 STATE/RRA0981 FTATE/RRA0981 FBB4: DC 7C0 B1 RECL0 //RAA0981 STATE/RRA0981 <				FSCNEW		#50F	IS THIS AN INTO
FBA9: 00 EG FEAS: 00 EG FESCULD ILESS THAN '1'? FBA8: 70 EG FFPT 65 CPP WEC9 ILESS DD DLD MAY FBA8: 70 EG FFPT 66 GEG FSCULD IND NORMAL FBB3: DC EG FSCULD IDD NORMAL FSCULD IND NORMAL FBB3: DC C006 71 SECULD IND NORMAL FBB3: DC C015 73 RECKTORMON EQU *CO15 //RAA0981 FBB3: DC C015 74 VERSION DFB *O6 //RAA0981 FBB3: DC C015 70 BIT RECKTORMON GET *RECKTORA0981 FTATE/RRA0981 FBB3: DC C015 FD14 X/OUTOCX EGT //RAA0981 FTATE/RRA0981 FBB4: DC 7C0 B1 STATE/RRA0981 STATE/RRA0981 FTATE/RRA0981 FBB4: DC 7C0 B1 RECL0 //RAA0981 STATE/RRA0981 <	FBA7: BO EE	FB97		2001121			; 'N' OR GREATER - DO IT!
FBAD: CP CC 67 CMP MEC 15 IT AN 'L'? FBAD: FD CF FPFT 20 BEE ESCLUD D0 MORAL FBBS: COOP FPTT 20 BEE ESCLUD D0 MORAL FBBS: COOP 77 SETLUTCRMON EQU ¥CCOP //RAA0981 FBBS: COOP 72 SETLUTCRMON EQU ¥CCOP //RAA0981 FBBS: COOP 73 VERSION FFB YARA0981 FBBS: COOP 73 VERSION FFB YARA0981 FBBS: 78 78 VERSION FFB YARA0981 YARA0981 FBBS: 78 70 CO FFB YARA0981 YARA0981 FBBS: 78 70 CO FFB YARA0981 YARA0981 FBBS: 78 70 CO FFB YARA0981 YARA0981 FBB: 78 70 CO FFB YARA0981 YARA0981 FBB: 78 70 NOP YARA0981 YAFA0981 F	FBA9: C9 C9		65				LESS THAN 'I'?
FBA: FO E6 FEPS 68 DEG ESCULD IDD NORMAL FBB: DO E7 000 70 ESCULD IDD NORMAL FBB: DO E6 000 70 SETLUTCRND 400 YRA0981 FBB: DO E6 000 73 RECKROM EQU YRA0981 FBB: DO E6 73 RECKROM EQU YRA0981 FBB: DO E6 73 VERSION PFB YA* FBB: DO E6 FBB YA PFB YA* YA* FBB: DO E6 FBB YA		FB97					YES, SO DO OLD WAY
FBB1: DD EB FBB2: COD D1 T FBB3: COD01 71 SETSLUTCROM EGU 4COD4 //RAO981 FBB3: COD07 71 SETSLUTCROM EGU 4COD4 //RAO981 FBB3: COD17 74 * FCD15 //RAO981 FBB3: COD17 74 * FCD15 //RAO981 FBB3: COD17 74 * FCD15 //RAO981 FBB3: GOT07 73 SET FCD2 //RAO981 FBB3: GOT0 RD PHP :SAVE SET FCM201 FBB3: GOT0 RD PHP :SAVE SET SET <td< td=""><td></td><td>6807</td><td></td><td></td><td></td><td></td><td></td></td<>		6807					
FBB3: 0001 70 D0 APPLE2E //RRA0951 FBB3: C007 72 SETINTCRNOM EQU ¥C007 //RRA0951 FBB3: C015 73 NECKNOM EQU ¥C007 //RRA0951 FBB3: C015 73 VERSION PFB YARA0951 FBB4: FBB3 73 VERSION PFB YARA0951 FBB4: YARA0951 JARA0951 JARA0951 JARA0951 FBB4: YARA0951 JARA0951 JARA0951 JARA0951 FBB4: YARA0951 YARA0951 JARA0951 JARA0951 FBB0: YARA0951 JARA0951							
FBB3: C007 72 SETINTCXRDH EQU +C07 //RA0981 FBB3: C015 77 SETINTCXRDH EPB //RA0981 FBB3: C015 77 SETINTCXRDH //RA0981 FBB4: FBB4: FBB4: 70 C010 //RA0981 FBB4: FBB4: 70 C010 //RA0981 //RA0981 FBB4: FBB4: 70 C010 //RA0981 //RRA0981 FBB4: C000 70 BIT PDCXRDH //RA0981 FBB4: C000 C1 FB1 PDCXRDH //RA0981 FBB2: C000 C1 FB1 PDCXRDH //RA0981 FBB2: C000 C1 FB1 //RA0981 //RA0981 FBC0: FB2 //RA0981 //RA0981 //RA0981 FB2: FB2: S S //RA0981 FB0: A NDP //RA0981 //RA0981 FB0: S 90 NDP //RA0981 //RA0981 S S 90 NDP /	FBB3:						
FBB3: C015 73 RDCXRDM EQU +C015 //RRAOPB1 FBB3: C6 73 VERSIDN DFB \$06 FDD.TDC:CK/RAOPB1 FBB3: C6 73 VERSIDN DFB \$06 FDD.TDC:CK/RAOPB1 FBB3: C00TOCX TAME FDD.TDC:CK/RAOPB1 INHIBIT DURING BANKBITCH/RAOPB1 FBB4: C0 TA FED.TDC:CV/REAT STAFE/RRAOPB1 ISAVE DURING BANKBITCH/RAOPB1 FBB4: D0 C1 B1 STA SETINCCARD ISAVE ROMBANK STAFE/RRAOPB1 FBC0: EA MP STA SETINCCARD ISAVE ROMBANK STAFE/RRAOPB1 FBC0: EA MP VECXO //RRAOPB1 //RRAOPB1 FBC0: EA MDP //RRAOPB1 //RRAOPB1 S B2 MDP //RRAOPB1 //RRAOPB1				SETSLOTCXRON	1 EQU	\$C006	; /RRA0981
FBB2: 74 * /RRA0981 FBB3: FBB4 75 0CTCX EQU * //RRA0981 FBB4: FBB4 75 0CTCX EQU * //RRA0981 FBB5: 76 CTCX EQU * //RRA0981 FBB5: 76 DTCX EQU * //RRA0981 FBB5: 76 DTC EI DTCC DTCC DTCC FBB5: 00 PHP DTCC DTCC DTCC DTCC DTCC FBB5: 00 PHP DTCC DTCC DTCC DTCC DTCC FBB5: DTCC BTC DTCC DTCC DTCC DTCC DTCC FBB5: DTCC BTC DTCC DTCC DTCC DTCC DTCC FBB5: DTCC BTC DTCC				SETINTCXROM	EQU		
FBB2: C6 FOR TDC/ECK//RAO981 FB84: FB8 FB8 FB8 FB8 FB8 FB85: FB8 FB8 FB8 FB8 FB8 FB8 FB85: FB8		0015			EGO	\$0010	
FB3:08 77 PHP ;SAVE USER IRG STATE/RAC991 FB3:08 2C 13 C0 79 BIT RDCXR0H ;GAVE CURRENT STATE/RAC991 FB3:08 00 94 STI RDCXR0H ;GAVE CURRENT STATE/RAC991 FB3:08 00 00 100 RAVE CURRENT STATE/RAC991 ;GAVE CURRENT STATE/RAC991 FB3:08 00 01 12 JAPP #: CURRENT STATE/RAC991 ;GAVE CURRENT STATE/RAC991 FB3:08 00 12 JAPP #: CURRENT STATE/RAC991 ;GAVE CURRENT STATE/RAC991 FB3:00 00 12 JAPP #: CURRENT STATE/RAC991 ;GAVE CURRENT STATE/RAC991 FB3:00 00 12 JAPP #: CURRENT STATE/RAC991 ;GAVE CURRENT STATE/RAC991 FB3:00 00 12 JAPP #: CURRENT STATE/RAC991 ;GAVE CURRENT STATE/RAC991 FB3:00 73 NDP ://RAC991 ;RAC991 S:00 91 NDP ://RAC991 ;RAC991 S:00 91 NDP ://RAC991 ;GAVE CURRENT STATE/RAC991 S:00 92 NDP ://RAC991 ;GAVE CURRENT STATE/RAC991 S:00 <t< td=""><td></td><td></td><td></td><td>VERSION</td><td>DFB</td><td>\$06</td><td>FOR IDCHECK/RRA0981</td></t<>				VERSION	DFB	\$06	FOR IDCHECK/RRA0981
FB81: 78 78 SEI ; INHIBIT DURING BANKBUTCH/RRAO981 FB85: 08 60 PHP DCT CURRENT STATE/RRAO981 FB85: 08 60 PHP SET ROMS ANK STATE/RRAO981 FB85: 08 60 JMP SET ROMS ANK STATE/RRAO981 FB00: 64 60 H VCRARA0981 FB00: 64 65 H NOP FF00: 64 84 NOP //RRA0981 FF00: 64 84 NOP //RRA0981 FF00: 64 84 NOP //RRA0981 S 85 87 NOP S 86 NOP //RRA0981 S 97 NOP //RRA0981 S 97 NOP //RRA0981 S 97 NOP //RRA0981 S 97 NOP ///RRA0981 S 97 NOP ////////////////////////////////////		FBB4		GOTOCX		*	
FBB2:2C 15 C0 79 BIT RDCXRDM JCET CURRENT STATE/RRA0951 FBB3:4C 00 C1 81 STA SET NTCXRDM JSET ROUS ON/RRA0951 FBB3:4C 00 C1 82 JPP FCION JSET ROUS ON/RRA0951 FBB3:4C 00 C1 83 JPP FCION JSET ROUS ON/RRA0951 FBC1: 84 NOP J/RA0951 FSC1: 85 ELSE JPP J/RA0951 S 87 NOP J/RA0951 S 87 NOP JPP S 87 NOP JPP S 90 NOP JPP S 91 NOP JPP S 92 NOP JPP S 93 NOP JPP S							SAVE USER IRQ STATE/RRA0981
FB3F:00 00 PHP :SAVE.ROMBANK.STATE_FRADOP01 FB3F:00 00 CO STA SETINTCXNON :=SOFF TO CXSPACE/RRAOP01 FB00: 00 CO STA SETINTCXNON :=SOFF TO CXSPACE/RRAOP01 FB00: 00 CO STA SET ROME ON/RRAOP01 ::FAVE.RAOP01 FB00: 00 STA SET ROME ON/RRAOP01 :/RRAOP01 ST 85 NOP :/RRAOP01 ST 85 NOP :/RRAOP01 ST 85 NOP :/RRAOP01 ST 97 NOP :/RRAOP01 ST 97 NOP :/RRAOP01 ST 97 NOP :/RRAOP01 ST 97 NOP :/RRAOP01 ST 90 NOP :/RRAOP01 ST 90 NOP :/RRAOP01 ST 90 NOP :/RRAOP01 ST 90 NOP :/RRAOP01 ST 101 #UST RKS #FEC1 :/RRAOP01 FB01 101 MUST RKS #FEC1 <td< td=""><td>FBB5:78</td><td>c0</td><td></td><td></td><td></td><td>PDCYDOM</td><td>; INHIBIT DURING BANKSWITCH/RRA0981</td></td<>	FBB5:78	c0				PDCYDOM	; INHIBIT DURING BANKSWITCH/RRA0981
FB3D:4C 00 C1 81 STA SET INTCRMP JET ROMS INVERAOPSI FB3D:4C 00 C1 82 JMP *CION >>>DFT DC CXPACE/RA09SI FB3D:4C 00 C1 83 NOP /RA09SI FB3D:4C 00 C1 85 84 NOP /RA09SI FBC0:EA 84 NOP /RA09SI FB1 85 ELDE /RA09SI S 95 97 NOP S 97 NOP //RA09SI FS 97 NOP //RA09SI S 97 NOP //RA09SI FS 97 NOP //RA09SI FS 97 NOP //RA09SI FS 97 NOP //RA09SI FS 97	FB89:08	0				RDCARDIT	SAVE ROMBANK STATE/RRA0981
FBC0: B3 * /RRA0981 FBC0:EA B4 NOP :/RRA0981 FBC0:EA B4 NOP :"I GOT PLENTY OF NOTHING!" S B7 NOP :"I GOT PLENTY OF NOTHING!" S 97 NOP :"I GOT CALCEASE ADDR IN BASL.H FEC1: 100 # MUST OFG #FEC1 : /RRA0981 FEC2: :00 :#I GOT #FEC1 : CALCEASE ADDR IN BASL.H FEC2: :00 :#I GOT #FEC1 : CALCEASE ADDR IN BASL.H FEC2: :00 :#I GOT #FEC1 : CALCEASE ADDR IN BASL.H FEC2: :00 <		ço				SETINTCXROM	SET ROMS ON/RRA0981
FBC0: PR //RRA0981 FSC1: 85 ELSE //RRA0981 S 85 NOP /"I GOT PLENTY OF NOTHING!" S 85 NOP /"I GOT PLENTY OF NOTHING!" S 85 NOP /"I GOT PLENTY OF NOTHING!" S 85 97 NOP S 97 NOP S 92 NUP S 93 NOP S 93 NOP S 93 NOP S 97 NOP S	FBBD: 4C 00	C 1			JMP	\$C100	
FBC1: B5 ELSE ;/RRA09B1 S B7 NOP ;"I GOT PLENTY OF NOTHING!" S B7 NOP ; S 97 NOP ; S 92 NOP ; S 93 NOP ; S 93 NOP ; S 95 NOP ; S 97 NOP ; ;/RRA09B1 FBC1: 100 FIN ;/RA09B1 FBC1: 101 * MUS ORG *FBC1 ;/REC9B1 FBC2:44 103 LSR A ; FOR GIVEN LINE ND. FBC2:44 103 LSR A ; ARG000000000000000000000000000000000000				*	NOP		
S B6 NOP ,"I GOT PLENTY OF NOTHING!" S B8 NOP ,"I GOT PLENTY OF NOTHING!" S B8 NOP S B7 NOP S 90 NOP S 91 NOP S 92 NOP S 92 NOP S 93 NOP S 93 NOP S 94 NOP S 97 NOP S 97 <t< td=""><td>FBCU: EA</td><td></td><td></td><td></td><td></td><td></td><td></td></t<>	FBCU: EA						
S 67 NOP S 89 NOP S 97 NOP S 91 NOP S 91 NOP S 92 NOP S 93 NOP FEC1: 100 FIN //RA0991 FEC2: 101 * MUE ORG \$FEC1 FEC1: 101 * MUE ORG \$FEC1 FEC2: 103 LSR A / CALCBASE ADDR IN BASL, H FEC2: 103 LSR A / CALCBASE ADDR IN BASL, H FEC2: 103 LSR A / CALCBASE ADDR IN BASL, H FEC2: 03 104 AND **03 / OCLINE NO. <=517			86				
S 90 NOP S 91 NOP S 91 NOP S 92 NOP S 93 NOP S 93 NOP S 93 NOP S 93 NOP S 97 NOP FBC1: 101 * MUST ORG *FBC1 CALCBASE ADDR IN BASL, H FBC2: 203 104 AND #031 / OCLUE NO.<<=#17			87				
S 90 NOP S 91 NOP S 92 NOP S 93 NOP S 94 NOP S 94 NOP S 97 NOP S 97 NOP S 97 NOP S 97 NOP FBC1: 100 FIN //RA0981 FBC2:44 103 LSR //CALCBASE ADDR IN BASL, H FBC2:44 103 LSR //CALCBASE ADDR IN BASL, H FBC2:60 04 103 LSR //CALCBASE ADDR IN BASL, H FBC2:78 03 104 AND #503 /CALCBASE ADDR IN BASL, H FBC2:80 03 104 AND #304 /AR00000ABCE, GENERATE FBC2:80 04 105 GRA #304 /AR000ABCBE, GENERATE FBC2:90 04 105 GRA #304 /AR000 /AR00 FBC2:91 105 GRA AAD #418 / BASL=CABCBC //AR00 <t< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td></t<>							
S 91 NOP S 92 NOP S 93 NOP S 93 NOP S 93 NOP S 93 NOP S 97 NOP FBC1: 101 * MUST DRG *FBC1 /ARA0981 FBC2: 102 BASCALC FHA /CALCBASE ADDR IN BASL, H FBC3: 703 103 LSR /A FBC3: 703 104 AND *001 /CRENCOCALCE FBC3: 703 103 LSR /A /CRCBASE ADDR IN BASL, H FBC3: 703 104 AND *001 /CRCBASE ADDR IN BASL, H FBC3: 703 104 AND *503 /CRCBASE ADDR IN BASL, H FBC3: 703 104 AND *503 /CRCBASE ADDR IN BASL, H FBC3: 75 105 STA ASSL /S							
5 92 NDP 5 93 NDP 5 93 NDP 5 95 NDP 5 97 NDP 5 97 NDP 5 97 NDP 5 97 NDP 7 101 * MUST ORG *FBC1 FBC1: 100 * FIN //RRA0981 FBC2:44 103 * LSR //CALCBASE ADDR IN BASL, H FBC2:44 103 * LSR //CALCBASE ADDR IN BASL, H FBC2:44 103 * LSR //CALCBASE ADDR IN BASL, H FBC2:44 103 * LSR //CALCBASE ADDR IN BASL, H FBC2:44 103 * LSR //CALCBASE ADDR IN BASL, H FBC3:90 04 105 CRACCC2 ENERATE FBC3:90 04 105 CRACCC2 ENERATE FBC3:90 04 107 PLA //ARA000001CD FBC3:90 04 107 PLA //ARA000001CD FBC3:90 04 113 ASL ASL FBD2:50 111 BASL ASL ASL <td>5</td> <td></td> <td>91</td> <td></td> <td></td> <td></td> <td></td>	5		91				
S 74 NDP S 75 NDP S 76 NDP S 77 NDP S 78 NDP FEC1: 100 FIN ;/RRA0981 FEC2: 101 * MUST DRG %FBC1 ;/ARA0981 FEC2:44 103 LSR A ; GALCBASE ADDR IN BASL, H FEC2:44 103 LSR A ; GALCBASE ADDR IN BASL, H FEC2:44 103 LSR A ; GALCBASE ADDR IN BASL, H FEC2:70 04 105 DRA H\$03 ; OC=LINE NO.C=\$17 FEC3:60 04 105 DRA H\$03 ; AND FEC3:60 04 105 DRA H\$03 ; AND FEC3:60 04 105 DRA H\$18 ; AND FED3:82 28 111 BASL A FBD2:04 112 ASL A A FBD3:05 28 114 DRA BASL BASL FBD4:05 28 114 RTA BASL F FBD5:06 07 117 BLL1 CHAR BASL FBD8:05 28 115 STA BASL F			92				
S 95 NDP S 97 NDP S 97 NDP FBC1: 100 FIN /RRA0981 FEC1: 101 * NUST DRG \$FBC1 /CALCBASE ADDR IN BASL,H FEC1: 101 * NUST DRG \$FBC1 /CALCBASE ADDR IN BASL,H FEC2: 102 BASCALC PHA /CALCBASE ADDR IN BASL,H FEC3: 00 FIN //RG=000ADCDC GENERATE FEC3: 02 04 ND<#000							
S 96 NOP S 97 NOP S 97 NOP FBC1: 100 FIN ;/RRA0781 FBC1: 101 * WUST OR 4FBC1 ;/CALCBASE ADR IN BASL,H FBC2:44 103 LSR A ;GALCBASE ADR IN BASL,H FBC3:27 03 103 LSR A ;GALCBASE ADR IN BASL,H FBC2:44 103 LSR A ;GALCBASE ADR IN BASL,H FBC3:07 04 AND #603 ;ACLBASE ADR IN BASL,H FBC3:07 04 AND #603 ;ACLBASE ADR IN BASL,H FBC5:07 105 DRA #603 ;ACLBASE ADR IN BASL,H FBC5:07 105 DRA #603 ;ACLBASE ADR IN BASL,H FBC6:07 105 DRA #603 ;ACLBASE ADR IN BASL,H FBC6:07 FBD 107 PLA #18 ;AR FBD2:00 FBD BCC BASL A FBD FBD2:00 112 BASL A FBD FBD FBD FBD2:00 <t< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td></t<>							
S 97 NUP S 97 NUP FBC1: 100 FIN ;/RA0981 FBC1: 100 FIN ;CALGBASE ADDR IN BASL, H FBC2:44 102 BASCALC PHA ;CALGBASE ADDR IN BASL, H FBC2:44 103 LSR A ;FDR GIVEN LINE NO. FBC3:27 03 104 AND #903 ;OCLENC LINE NO. FBC3:29 04 105 DRA #304 ;ARG-000ABCDE, GENERATE FBC7:68 107 PLA ;AND #ADD FBC6:90 02 FBD0 FBD0 PASELE AADD FBC6:90 02 FBD0 ASL A FBD3 FBD2:95 111 BASCLC2 STA BASL FBD3 FBD2:95 113 ASL A FBD3 FBD3 FBC7 FBD3 FBD3:95 113 ASL A FBD3 FBD3 FBC7 FBC1 FBC7 FBC1 FBC7 FBC1 FBC7 FBC1 FBC7 FBC1 FBC7 FBC1							
S 98 NOP FBC1: 100 FIN ;/RRA0981 FBC1: 101 * WIST GR 4FBC1 ;/GALCBASE ADR IN BASL,H FBC2:44 103 LSR A ; GALCBASE ADR IN BASL,H FBC3:27 03 104 AND #903 ; OC=LINE ND.<=\$17			97				
FBC1: 100 FIN ;/RA0981 FBC1: 101 * MUST 0G *FBC1 ;CALCBASE ADR IN BASL,H FBC2:44 103 LSR A ;CALCBASE ADR IN BASL,H FBC2:44 103 LSR A ;CALCBASE ADR IN BASL,H FBC2:63 03 104 AND #603 ;CALCBASE ADR IN BASL,H FBC2:68 107 PLA ;ARG=000ABCDE, GENERATE FBC2:67:85 29 106 STA BASH ;BASH=000001CD FBC3:67:96 107 PLA ;AND #518 ;JBASH=0000001CD FBC6:67:7F 110 ADC #857 FBD2:00 FBC5:67 FBD2:00 112 ASL A FBD2:00 112 ASL A FBD2:00 112 ASL A FBD3:03 113 ASL A FBD3:05 116 RTS FBD4:05:28 114 ORA BASL FBD3:06 116 RTS FBD5:09 12 FBF ISB NR RTSB FBD ISC FBD5:00 12 BEL1 CMP <			98		NOP		
FBC1: 101 * MUST ORG #FBC1 FBC1: 102 BASCALC PHA ; CALCBASE ADDR IN BASL, H FBC2:4A 103 AND *FOR GIVEN LINE ND. FBC3:29 03 104 AND *FOR GIVEN LINE ND. FBC3:29 03 104 AND *FOR GIVEN LINE ND. FBC3:29 03 104 STA BASH ; ARG=000ABCE, GENERATE FBC3:29 03 106 STA BASH ; BASH=000001CD FBC4:29 18 108 AND #502 AND FBC4:29 18 109 BASL AND FBD2:82 111 BASLC2 STA BASL FBD3:82 28 111 BASL A FBD3:03 28 113 ASL A FBD3:04 113 ASL A FBD5:85 28 115 STA BASL FBD5:80 29 117 BELL CHAR? (CONTROL-6) FBD8:80 20 12 FBEF 118 BNE RTS2B ; NO, RETURN. FBD5:49 40 119 LDA #4807 ; BELL CHAR? (CONTROL-6) ; FBEF </td <td></td> <td></td> <td>99</td> <td></td> <td>MAP</td> <td></td> <td></td>			99		MAP		
FBC1:48 102 BASCALC PHA ;CALCBASE ADR IN BASL,H FBC2:4A 103 LSR A ;FDR GIVEN LINE ND. FBC3:29 03 104 AND #603 ; OC=LINE ND. FBC3:69 04 105 DRA #604 ; ARG=000ABCDE, GENERATE FBC7:85 29 106 STA BASH ; BASH=0000001CD FBC3:67 107 PLA ; AND #604 ; AND FBC6:67 7F 110 ADC #851 ; BASL=EABAB000 FBC5:67 7F 110 ADC #87F FBD0:85 28 111 BASL A FBD2:0A 112 ASL A FBD3:0A 113 ASL A FBD8:05 28 114 ORA BASL FBD8:05 12 FBF ; BELL CONTROL-G) FBD8:00 12 FBF ; BASL ; CONTROL-G) FBD8:00 12 FBF ; AA ; FBF FBD8:00 12 FBF ; AA ; FBF <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>							
FBC2: 4A 103 LSR A ; FDR GIVEN LINE ND. FBC3: 29 03 04 AND #003 ; O<=LINE ND.<==17				* MUST OF	FIN		; /RRA0981
FBC2:85:29 106 DRA #104 ;ARG=000ABCDE, GENERATE FBC2:85:29 106 STA BASH ; BASH=000001CD FBC2:85:29 107 PLA ; AND FBC4:29:18 108 AND #118 ; BASH=000001CD FBC5:67:68 107 PLA ; AND FBC5:67:79 100 ADC #87C FBC5:67:70 110 ADC #87F FBD2:0A 112 ASL A FBD3:0A 112 ASL A FBD4:05:28 114 ORA BASL FBD5:85:28 117 STA BASL FBD6:85:28 117 BELL CHAR? (CONTROL-G) FBD8:00 12 FBEF 118 BNE RTS28 ; NO, RETURN. FBD8:20 12 FBET IS STA WAT ; DELAY .01 SECONDS FBE5:20 AB CC 123 JSR WAT ; DELAY .01 SECONDS FBE5:40 CC 122 BL2 LDA #46CO ; DELAY .01 SECONDS	FBC1:		101		FIN G \$FBC	1	
FBC7:85 29 106 STA BASH ; BASH-000001CD FBC7:65 107 PLA ; AND FBC7:67 108 AND #18 ; BASL=EABAB000 FBC2:60 02 FBD0 109 BCC BASLC2 FBC2:60 02 FBD0 109 BCC BASL FBD2:0A 112 ASL A FBD3:0A 113 ASL A FBD3:05 28 114 ORA BASL FBD3:05 28 114 ORA BASL FBD8:00 116 RT5 FBD8:00 116 RT5 FBD8:00 116 RT5 FBD9:012 FBE1 118 BNE RTS28 ; NO, RETURN. FBD8:00 12 FBE1 B NE RTS28 ; DAGLEAY (CONTROL-G) FBD9:00 12 FBE1 LDA #400 ; DELAY (O SECONDS FBE2:00 122 BEL12 LDA #400 ; DGLES SPEAKER AT 1 KHZ FBE4:49 02 JSTR <td>FBC1: FBC1: 48 FBC2: 4A</td> <td></td> <td>101 102 103</td> <td></td> <td>FIN G \$FBC PHA LSR</td> <td>A</td> <td>CALCBASE ADDR IN BASL, H</td>	FBC1: FBC1: 48 FBC2: 4A		101 102 103		FIN G \$FBC PHA LSR	A	CALCBASE ADDR IN BASL, H
FBC2:68 107 PLA ; AND FBCA:29 108 AND #\$18 ; BASL=EABAB000 FBCC:67 109 BCC BASCLC2 ; BASL=EABAB000 FBCE:67 FE 110 ADC #\$7F FBD0:85 28 111 BASCLC2 STA BASL FBD2:0A 112 ASL A A FBD3:65 28 114 ORA BASL FBD6:65 28 114 ORA BASL FBD8:65 28 117 BELL CMAR? (CONTROL-G) FBD8:65 28 117 BELL CHAR? (CONTROL-G) FBD8:67 FDF 117 BELL MAT ; DELAY .01 SECONDS FBD9:20 AB FC 120 JSR WAIT ; DELAY .01 SECONDS FBE2:40 CO 121 LDY #\$CC ; FBE2; AD GG 127 FBE5:40 CO 122 BELL LDA STA BASL ; FBE2; AD GG 144 LDA STA FASC FBE6:20	FBC1: FBC1:48 FBC2:4A FBC3:29 03		101 102 103 104		FIN PHA LSR AND	A #\$03	;CALCBASE ADDR IN BASL,H ; FOR GIVEN LINE NO. ; O<=LINE NO. <=\$17
FBCC: 90 62 FBD0 100 ACC BASLLC2 FBCE: 67 110 ACC #ASCLC2 FL FBCE: 67 110 ACC #ASCL FBD2: 0A 112 ASL A FBD3: 0A 112 ASL A FBD4: 05 28 114 ORA BASL FBD5: 65 28 114 ORA BASL FBD6: 65 28 117 BELL ORA FBD7: 07 117 BELL CMP #487 FBD8: 60 116 RTS FBEAR NO. RETURN. FBD9: 67 117 BELL M440 ; VES FBD7: 67 117 BELL M440 ; VES FBD8: 60 121 LD4 #440 ; VES FBD8: 60 121 LD4 #400 ; VES FBE7: 60 00 121 LD4 #400 ; SEC. FBE6: 60 123 JSR WAIT ; DELAY .01 SECONDS FBE6: 60 127 RTS2 DEY<	FBC1: FBC1:48 FBC2:4A FBC3:29 03 FBC5:09 04		101 102 103 104 105		FIN CG \$FBC PHA LSR AND ORA	A #\$03 #\$04	; CALCBASE ADDR IN BASL,H ; FOR GIVEN LINE NO. ; O<=LINE NO. <=\$17 ; ARG=0000ABCDE, GENERATE
FBCE:69 7F 110 ADC ##7F FBD0:85 28 111 BASCL STA BASL FBD3:0A 112 ASL A FBD3:0A 113 ASL A FBD3:0A 113 ASL A FBD3:0A 113 ASL A FBD3:0A 113 ASL A FBD8:60 116 RTS BASL FBD8:60 116 RTS ; FBD8:60 117 BELL1 CMP ##87 ; BELL CHAR? (CONTROL-G) FBD8:60 12 FBE IIA MA RTSR ; SECURAT FBD8:60 12 FBE IIA HAT ; DELLY CONTROL-G) FFE FBD7:20 AB FC 120 JSR WAIT ; DECV SECONDS FBE5:40 CO 122 BLA #400 ; TGGQLE SPEAKER AT 1 KHZ FBE5:40 CO 124 LDA #400 ; CURSOR H INDEX Y=RE FBE5:40 CO 127 RTS FGE	FBC1: FBC1: 48 FBC2: 4A FBC3: 29 03 FBC5: 09 04 FBC7: 85 29		101 102 103 104 105 106		FIN CG \$FBC PHA LSR AND ORA STA	A #\$03 #\$04)CALCBASE ADDR IN BASL,H ;FOR GIVEN LINE NO. ; O<=LINE NO.<=\$17 ;ARG=000ABCDE, GENERATE ; BASH=000001CD
FBD2:05:28 111 BASL STA BASL FBD2:0A 112 ASL A FBD3:0A 113 ASL A FBD3:0A 113 ASL A FBD4:05:28 114 ORA BASL FBD3:60 115 STA BASL FBD3:05:28 117 BELL CMP #887 ; BELL CHAR? (CONTROL-G) FBD8:05:28 117 BELL CMP #887 ; BELL CHAR? (CONTROL-G) FBD8:07 12 FBEF 118 BNE RTS28 ; NO, RETURN. FBD8:07 12 FBEF 119 LDA #440 ; YES FBD8:07 12 BEL LDA #440 ; YES FBE7:08 0 121 LDY #400 FBE3:00 12 FBE4 LDA #490 ; SCONDS FBE5:00 65 FBE4 LDA SPKR FBE7:91 STA (BASL); ; STORE CHAR IN LINE FBE7:128 127 STA (BASL); ; STORE CHAR IN LINE FEF2:91 STA <td>FBC1: FBC1: 48 FBC2: 4A FBC3: 27 03 FBC5: 07 04 FBC7: 85 27 FBC7: 68 FBCA: 27 18</td> <td></td> <td>101 102 103 104 105 106 107 108</td> <td></td> <td>FIN RG \$FBC PHA LSR AND ORA STA PLA AND</td> <td>A #\$03 #\$04 BASH #\$18</td> <td>; CALCBASE ADDR IN BASL; H ; FOR GIVEN LINE ND. ; O<=LINE ND.<=\$17 ; ARG=000ABCDE, GENERATE ; BASH=000001CD ; AND</td>	FBC1: FBC1: 48 FBC2: 4A FBC3: 27 03 FBC5: 07 04 FBC7: 85 27 FBC7: 68 FBCA: 27 18		101 102 103 104 105 106 107 108		FIN RG \$FBC PHA LSR AND ORA STA PLA AND	A #\$03 #\$04 BASH #\$18	; CALCBASE ADDR IN BASL; H ; FOR GIVEN LINE ND. ; O<=LINE ND.<=\$17 ; ARG=000ABCDE, GENERATE ; BASH=000001CD ; AND
FBD2:0A 112 ASL A FBD3:0A 113 ASL A FBD4:05:28 114 ORA BASL FBD3:60 115 STA BASL FBD3:60 116 RTS FBD3:60 117 BELL CMTP FBD5:60 117 BELL1 CMP ##87 FBD5:60 12 FBET BNR RTS28 ; NO, RETURN. FBD5:20 AB FC 120 JSR WAIT ; DELAY .01 SECONDS FBE2:A0 CO 121 LD4 ##400 ; VES FDS:20 AB FC FBE5:A0 CO 122 BELL2 LD4 ##0C ; TDGGLE SPEAKER AT 1 KHZ FBE5:60 127 RTS28 RTS FBE5:60 127 RTS28 FBF2:9128 127 STA (BASL),Y ; STORE CHAR IN LINE YES.CR FBF2:128 127 STA (BASL),Y ; STORE CHAR IN LINE YES.CR FBF2:9128 127 STA (BASL),Y ; STORE CHAR IN LINE YES.CR FBF2:04 <	FBC1: FBC1:48 FBC2:4A FBC3:27 03 FBC5:07 04 FBC7:85 27 FBC7:48 FBC4:27 18 FBC4:27 18 FBC4:27 18	FBDO	101 102 103 104 105 106 107 108 109		FIN RG \$FBC PHA LSR AND ORA STA PLA AND BCC	A #\$03 #\$04 BASH #\$18 BASCLC2	; CALCBASE ADDR IN BASL; H ; FOR GIVEN LINE ND. ; O<=LINE ND.<=\$17 ; ARG=000ABCDE, GENERATE ; BASH=000001CD ; AND
FBD3:0A 113 ASL A FBD4:05 28 114 ORA BASL FBD8:05 28 115 STA BASL FBD8:05 28 115 STA BASL FBD8:00 116 RTS FBD8:00 12 FBF 118 DNE FDD1:47 40 179 LDA #\$40 ; YES FBD2:20 A8 FC 120 JSR WAIT ; DGLAY OI SECONDS FBE2:A0 CO 121 LDY #\$40 ; YES FBE3:A0 OC 122 BELL2 LDA #\$40 ; YES FBE5:20 A8 FC 120 JSR WAIT ; DGGLE SPEAKER AT 1 KHZ FBE5:30 GO 122 BLL2 LDA #\$00 ; TOGGLE SPEAKER AT 1 KHZ FBE5:40 30 CO 127 RTS2 FBF FBF FBF STGRADV LDA STGRADV STGRADV INDEX D Y-REG FBE7:40 127 RTS28 RTS FBF8:524 131 <t< td=""><td>FBC1: FBC1:48 FBC2:4A FBC3:29 03 FBC5:09 04 FBC7:85 29 FBC9:68 FBCA:29 18 FBCC:90 02 FBCE:69 7F</td><td>FBDO</td><td>101 102 103 104 105 106 107 108 109 110</td><td>BASCALC</td><td>FIN CG \$FBC PHA LSR AND ORA STA PLA AND BCC ADC</td><td>A #\$03 #\$04 BASH #\$18 BASCLC2 #\$7F</td><td>; CALCBASE ADDR IN BASL; H ; FOR GIVEN LINE ND. ; O<=LINE ND.<=\$17 ; ARG=000ABCDE, GENERATE ; BASH=000001CD ; AND</td></t<>	FBC1: FBC1:48 FBC2:4A FBC3:29 03 FBC5:09 04 FBC7:85 29 FBC9:68 FBCA:29 18 FBCC:90 02 FBCE:69 7F	FBDO	101 102 103 104 105 106 107 108 109 110	BASCALC	FIN CG \$FBC PHA LSR AND ORA STA PLA AND BCC ADC	A #\$03 #\$04 BASH #\$18 BASCLC2 #\$7F	; CALCBASE ADDR IN BASL; H ; FOR GIVEN LINE ND. ; O<=LINE ND.<=\$17 ; ARG=000ABCDE, GENERATE ; BASH=000001CD ; AND
FBD2:65 28 115 STA BASL FBD8:60 116 RTS FBD8:60 117 BELL CMP ##87 ; BELL CHAR? (CONTROL-G) FDD8:00 12 FBEF 118 BNE RTS28 ; NO, RETURN. FDD8:00 12 FBEF 118 BNE RTS28 ; NO, RETURN. FBD0:47 40 117 LDA ##40 ; YES FBE2:A0 CO 121 LDY ##0C ; TOGGLE SPEAKER AT 1 KHZ FBE4:A7 CO 122 BELL2 LDA ##0C ; TOGGLE SPEAKER AT 1 KHZ FBE5:20 AB FC 123 JSR WAIT ; FOR .1 SEC. FBE5:40 30 CO 124 LDA SPKR FBE5:60 127 RTS2 RTS FBF5:61 STORADV LDY CH ; CURSOR H INDEX TO Y-REG FBF7:128 127 STS2 RTS FBF6:62 131 LDA CH ; INCREMENT CURSOR H INDEX FBF5:05 21 132 CMP WNOWDTH ; BEYO	FBC1: FBC1: 48 FBC2: 4A FBC3: 27 03 FBC5: 07 04 FBC7: 85 29 FBC7: 68 FBCA: 27 18 FBCA: 27 18 FBCC: 70 02 FBCE: 67 7F FBD0: 85 28	FBDO	101 102 103 104 105 106 107 108 107 110 110	BASCALC	FIN CG \$FBC PHA LSR AND ORA STA PLA AND BCC ADC STA	A #\$03 #\$04 BASH #\$18 BASCLC2 #\$7F BASL	; CALCBASE ADDR IN BASL; H ; FOR GIVEN LINE ND. ; O<=LINE ND.<=\$17 ; ARG=000ABCDE, GENERATE ; BASH=000001CD ; AND
FBDB:60 116 RTS FBDB:60 117 BELL CMP #\$877 ; BELL CHAR? (CONTROL-G) FBDB:00 12 FBEF 118 BNE RTS28 ; NO. RETURN. FBDD:04 40 119 LDA #\$40 ; YES FBDD:20 AB FC 120 JSR WAIT ; DELAY. 01 SECONDS FBE2:A0 CO 121 LDA #\$400 ; TOGGLE SPEAKER AT 1 KHZ FBE3:A7 OC 122 BELL LDA #\$400 ; TOGGLE SPEAKER AT 1 KHZ FBE3:A0 GCO 124 LDA SFWR ; FBE5:A0 GCO 124 LDA SFWR FBE5:A0 GCO 127 RTS28 RTS FBE5:A0 127 TTS28 RTS FBE7:40 127 RTS28 RTS FBF2:91 28 129 STORADV LDY (HASL),Y ; STORE CHAR IN LINE FBF3:80 24 130 ADVANCE INC CH ; INCREMENT CURSOR H INDEX FBF3:80 66 F66	FBC1: FBC1: 48 FBC2: 4A FBC3: 29 03 FBC5: 09 04 FBC7: 65 29 FBC9: 68 FBCA: 29 18 FBCA: 29 18 FBCA: 29 18 FBC6: 69 7F FBD0: 65 28 FBD2: 0A	FBDO	101 102 103 104 105 106 107 108 109 110 111 112 113	BASCALC	FIN CG \$FBC PHA LSR AND DRA STA PLA AND BCC STA ASL ASL	A #\$03 #\$04 BASH #\$18 BASLC2 #\$7F BASL A A	; CALCBASE ADDR IN BASL; H ; FOR GIVEN LINE ND. ; O<=LINE ND.<=\$17 ; ARG=000ABCDE, GENERATE ; BASH=000001CD ; AND
FBD9:C9 67 117 BELL1 CMP ##87 ;BELL CHAR? (CONTROL-G) FBDB:D0 12 FBEF 118 BNE RTS28 ; NO, RETURN. FBDD:D4 40 119 LDA ##40 ; YES FBDF:D2 A8 FC 120 JSR WAIT ; DELAY.01 SECONDS FBE2:A9 CO 121 LDY ##6C ; TOGGLE SPEAKER AT 1 KHZ FBE4:A9 CO 122 BELL2 LDA ##0C ; TOGGLE SPEAKER AT 1 KHZ FBE5:A0 30 CO 124 LDA SFWR FEC:BE FEC:BE FBE5:A0 30 CO 127 RTS28 RTS ; CURSOR H INDEX TO Y-REG FBF2:91<28	FBC1: FBC1: FBC2: FBC3: FBC3: FBC3: FBC7: FBC7: FBC7: FBC7: FBC7: FBC7: FBC3: FBC4: FBC4: FBC5: FBD2: FBD2: FBD3: FBD4: FBD5: FBD4: FBD4: FB5 <td>FBDO</td> <td>101 102 103 104 105 106 107 108 107 110 110 111 112 113 114</td> <td>BASCALC</td> <td>FIN CG \$FBC PHA LSR AND ORA STA PLA AND BCC ADC STA ASL ASL ORA</td> <td>A #\$03 #\$04 BASH #\$18 BASCLC2 #\$7F BASL A A BASL</td> <td>; CALCBASE ADDR IN BASL; H ; FOR GIVEN LINE ND. ; O<=LINE ND.<=\$17 ; ARG=000ABCDE, GENERATE ; BASH=000001CD ; AND</td>	FBDO	101 102 103 104 105 106 107 108 107 110 110 111 112 113 114	BASCALC	FIN CG \$FBC PHA LSR AND ORA STA PLA AND BCC ADC STA ASL ASL ORA	A #\$03 #\$04 BASH #\$18 BASCLC2 #\$7F BASL A A BASL	; CALCBASE ADDR IN BASL; H ; FOR GIVEN LINE ND. ; O<=LINE ND.<=\$17 ; ARG=000ABCDE, GENERATE ; BASH=000001CD ; AND
FBDB:D0 12 FPEF 118 BNE RTS28 ; NO, RETURN. FBDD:A9 40 119 LDA #\$40 ; YES FBDD:A9 40 119 LDA #\$40 ; YES FBDE:A9 40 121 LDY #\$00 ; DGLAY.01 SECONDS FBE2:A0 CO 121 LDY #\$00 ; TOGQLE SPEAKER AT 1 KHZ FBE4:A0 0C 122 BELL2 LDA #\$00 ; TOGQLE SPEAKER AT 1 KHZ FBE5:A0 30 CO 124 LDA SPKR FOR .1 SEC. FBE6:B0 F5 FBE4 L26 BNE BELL2 FBE7:A0 127 RTS28 RTS FTS FBE7:A0 127 RTS28 RTS FTS FBE7:A0 ADVANCE INC CH ; CURSOR H INDEX TO Y-REG FBF3:S0 ADVANCE INC CH ; INCREMENT CURSOR H INDEX FBF6:S2 131 LDA CH ; MOVERT CURSOR H INDEX FBF6:S2 132 CHP WNDMDTH ; EVOND WINDOW WIDTH? FBF6:S2 133 RTS3 ; NO. NETURN. FBF5:S0<	FBC1: FBC1: FBC2: FBC3: FBC5: FBC5: FBC7: FBC7: FBC6: FBC7: FBC7: FBC7: FBC6: FBC7: FBC6: FBC6: FBC6: FBD0: FBD0: FBD2: CA FBD4: FBD5: FBD4: FBD5: FBD5: FBD5: FBD5: FBD5: FBD5:	FBDO	101 102 103 104 105 106 107 108 107 110 111 112 113 114 115	BASCALC	FIN CG \$FBC PHA LSR AND ORA STA AND BCC ADC STA ASL ORA STA	A #\$03 #\$04 BASH #\$18 BASCLC2 #\$7F BASL A A BASL	; CALCBASE ADDR IN BASL; H ; FOR GIVEN LINE ND. ; O<=LINE ND.<=\$17 ; ARG=000ABCDE, GENERATE ; BASH=000001CD ; AND
FBDD:A9 40 119 LDA ##40 ; YES FEDF:20 AB FC 120 LDA ##40 ; JDELAY.01 SECONDS FFBE:A0 CO 121 LDY ##5CO ; JDELAY.01 SECONDS FFBE:A70 C 122 BELL2 LDA ##0CC ; TOGOLE SPEAKER AT 1 KHZ FBE:A10 A0 CO 124 LDA ##0CC ; TOGOLE SPEAKER AT 1 KHZ FBE:A10 A0 CO 124 LDA SPKR FBE SPER FBE:A1 125 DEV FBEF:A0 127 RTS2B FDF STORE CHAR IN LINE FDF FBF:A128 127 RTS2B RTS ; CURSOR H INDEX TO Y-REG FBF3:A2 131 LDA CH ; INCREMENT CURSOR H INDEX FEG FBF3:A5 24 131 LDA CH ; INCREMENT CURSOR H INDEX FBF3:A3 STORADV STORE CHAR IN LINE FBF3:A3 STORADV ; STORE CHAR IN LINE FBF3:A3 STORADV ; STORE CHAR IN LINE FBF3:A3 STORADV ; STORE CHAR	FBC1: FBC2:40 FBC3:27 03 FBC5:07 04 FBC7:85 27 FBC7:68 FBC7:68 FBC2:70 02 FBC6:27 18 FBC6:27 18 FBC6:27 18 FBC6:47 FBD3:05 28 FBD3:0A FBD4:05 28 FBD4:05 28 FBD4:05 28 FBD8:40	FBDO	101 102 103 104 105 106 107 108 107 110 111 112 113 114 115 116	BASCALC BASCLC2	FIN CG \$FBC PHA LSR AND ORA STA PLA AND BCC STA ASL ORA STA STA STA RTS	A #\$03 #\$04 BASH #\$18 BASCLC2 #\$7F BASL A BASL BASL BASL BASL	;CALCBASE ADDR IN BASL,H ;FOR GIVEN LINE ND. ; O<=LINE ND.<=\$17 ;RAG=000ABCDE, GENERATE ; BASH=000001CD ; AND ; BASL=EABAB000
FFBE2:A0 C0 121 LDY ##CO FFBE4:A7 9C 122 BELL2 LDA ##CO FFBE4:A7 9C 122 BELL2 LDA ##CO FFBE4:A7 9C 122 BELL2 LDA ##CO FBE5:A0 A8 FC 122 BELL2 LDA SPKR FBE5:A0 A5 F6 124 LDA SPKR FBE5:A0 F5 FBE4 126 DEV FBE5:A0 F5 FBE4 127 RTS28 RTS FBF2:91 28 127 RTS28 RTS ; CURSOR H INDEX TO Y-REG FBF3:A1 28 127 RTS ; STORE CHAR IN LINE FBF3:A1 28 127 RTS (BASL).Y ; STORE CHAR IN LINE FBF3:A5 24 131 LDA CH ; (MOVE RIGHT) FBF3:A5 24 133 BCS CR ; NO. RETURN. FBF5:A5 21 132 CMP WNDWDTH ; BEYOND WIDTH? FBF3:B0 64 64 FC42 133 BCS FBF5:A0 134 NT33 RTS ; NO. RETURN. FBF5:B0 F 135 VIDOUT CMP #A00 ; CONTROL CHAR? FBF5:B0 EF FBF0 136 BCS	FBC1: FBC1:40 FBC2:40 FBC3:29 03 FBC3:09 04 FBC7:85 29 FBC3:68 FBC3:08 FBC3:08 FBC3:08 FBC3:08 FBD3:05 FBD3:05 FBD4:05 FBD4:05 FBD4:05 FBD4:05 FBD4:05 FBD4:05 FBD4:05 FBD4:05 FBD5:00		101 102 103 104 105 106 107 108 107 110 111 112 113 114 115 116 117 118	BASCALC BASCLC2	FIN C SFBC PHA LSR AND ORA STA PLA AND BCC STA ADC STA ADC STA ADC STA STA CRA STA STA STA STA STA STA STA ST	A #\$03 #\$04 BASH #\$18 BASLC2 #\$7F BASL A A BASL BASL BASL BASL BASL BASL BASL BASL BASL	<pre>;CALCBASE ADDR IN BASL,H ;FOR GIVEN LINE ND. ; O<=LINE ND.<=\$17 ;ARG=000ABCDE, GENERATE ; BASH=000001CD ; AND ; BASL=EABAB000 ;BELL CHAR? (CONTROL-G) ;ND, RETURN.</pre>
FBER:A9 OC 122 BELL2 DA #BOC ;TOGGLE SPEAKER AT 1 KHZ FBEG:20 AB FC 123 JSR WAIT ; FOR .1 SEC. ; FBEG:20 SPEAKER AT 1 KHZ FBED:20 GC 124 LDA SPKR ; FOR .1 SEC. FBED:20 GF FBE4 125 DEY ; FBED:20 FS FBE4 ; CURSOR H INDEX TO Y-REG FBEF:40 127 RTS2 RTS ; STORE CHARIN LINE ; FBF6:42 130 ADVANCE INC CH ; INCREMENT CURSOR H INDEX FBF6:42 130 ADVANCE INC CH ; INCREMENT CURSOR H INDEX FBF6:42 130 ADVANCE INC CH ; INCREMENT CURSOR H INDEX FBF6:42 130 ADVANCE INC CH ; INCREMENT CURSOR H INDEX FBF6:43 24 131 LDA CH ; INCENSOR H INDEX FDA FBF6:43 24 130 ADVANCE INDEX	FBC1: FBC2:4A FBC2:4A FBC3:27 03 FBC3:07 04 FBC7:65 27 FBC7:65 27 FBC7:68 FBC2:07 27 FBC2:00 22 FBC2:00 22 FBC2:00 25 FBD2:04 FBD3:00 FBD4:05 28 FBD8:00 12 FBD8:00 12 FBD8:00 47 FBD8:00 47 FBD8:00 47	FBEF	101 102 103 104 105 106 107 108 107 110 111 112 113 114 115 116 117 118	BASCALC BASCLC2	FIN C \$FBC PHA LSR AND ORA STA PLA AND BCC ADC STA ASL ASL ORA STA RTS CMP BNE LDA	* 4 * 803 * 804 BASH * 818 BASL * 877 BASL A A BASL BASL # 887 RS28 * 840	<pre>;CALCBASE ADDR IN BASL,H ;FOR GIVEN LINE ND. ;O<=LINE ND.<=\$17 ;ARG=000ABCDE, GENERATE ; BASH=000001CD ; AND ; BASL=EABAB000 ;BASL=EABAB000</pre>
FBE6:20 A8 FC 123 JSR WAIT ; FDR .1 SEC. FBE9:A0 30 C0 124 LDA SPKR FBE0:D0 F5 FBE4 125 DEY FBE0:D0 F5 FBE4 126 BNE BELL2 FBE7:A0 127 TTS2B RTS FBF FBF0:A4 24 128 STORADV LDY CH ; CURSOR H INDEX TO Y-REG FBF2:91 25 127 STA (BASL),Y ; STORE CHAR IN LINE FBF4:26 24 130 ADVANCE INC CH ; INCREMENT CURSOR H INDEX FBF6:A5 21 132 CMP WADWDTH ; BEYOND WIDTH? FBF6:A5 24 133 BCS CR ; VES. CR TO NEXT LINE. FBF6:B0 64 F626 133 RTS ; NO. NETVEN. FBF7:B0 64 FC0 135 VIDUT CMP #A00 ; CONTROL CHAR? FBF7:B0 FFF0 136 BCS STORADV ; NO. OUTPUT IT. ; NOC2:10 FC02:10 EC FBF0 138 BPL S	FBC1: FBC2:4A FBC2:4A FBC3:27 03 FBC3:07 04 FBC7:85 27 FBC7:65 FBC3:64 FBC3:07 02 FBC2:07 02 FBC2:07 02 FBC2:07 02 FBD2:05 28 FBD2:0A FBD4:05 28 FBD4:05 28 FBD4:05 28 FBD4:05 28 FBD4:05 28 FBD4:05 28 FBD4:05 28 FBD5:00 12 FBD5:20 48	FBEF	101 102 103 104 105 106 107 108 107 110 111 112 113 114 115 116 117 118 119 120	BASCALC BASCLC2 BELL1	FIN G \$FBC PHA LSR AND ORA STA PLA AND STA AND AND STA AND AND AND AND AND AND AND AN	* 403 #\$03 BASH #\$18 BASCL2 #\$77 BASL A A A BASL BASL BASL BASL BASL BAS	<pre>;CALCBASE ADDR IN BASL,H ;FOR GIVEN LINE ND. ;O<=LINE ND.<=\$17 ;ARG=000ABCDE, GENERATE ; BASH=000001CD ; AND ; BASL=EABAB000 ;BASL=EABAB000</pre>
FBEC:80 124 LDA SPKR FBEC:80 125 DEY FBED:D0 F5 FBE4 126 BNE BELL2 FBED:20 F5 FBE4 127 RTS2 RTS FBEF:60 127 RTS2 RTS FBF:61 129 STA (BASL), y ; STORE CHAR IN LINE FBF:42 128 STORADV LDY CH ; INCREMENT CURSOR H INDEX FBF:42 129 STA (BASL), y ; STORE CHAR IN LINE FBF:42 131 LDA CH ; (MOVE RIGHT) FBF:82 131 LDA CH ; (MOVE RIGHT) FBF:80 64 FC42 133 RTS ; NG, RETURN, FBF:80 64 FC42 134 RTS3 RTS ; NG, RETURN, FBF:80 64 FC42 134 RTS3 RTS ; NG, OUTPUT IN, FBF:80 65 FTBF 134 RTS3 RTS ; NG, OUTPUT IT, FBF:80 F6 FBF0 136 BPL STORADV ; NVERSE VIDEO? FC02:10 F0 FB7 138 BPL STORADV ; VES, OUTPUT IT, FC03:60 5A FC42 <	FBC1: FBC2:40 FBC2:40 FBC3:27 03 FBC3:27 03 FBC3:27 03 FBC3:27 03 FBC3:27 18 FBC3:27 18 FBC3:27 18 FBC3:27 18 FBC3:27 18 FBD3:05 28 FBD3:00 FBD4:55 28 FBD3:00 FBD5:20 47 FBD5:20 47 FBD5:20 47	FBEF	$\begin{array}{c} 101\\ 102\\ 103\\ 104\\ 105\\ 106\\ 107\\ 108\\ 109\\ 110\\ 1112\\ 112\\ 113\\ 114\\ 115\\ 114\\ 117\\ 118\\ 119\\ 120\\ 121 \end{array}$	BASCALC BASCLC2 BELL1	FIN G \$FBC PHA LSR AND BCC ADC ADC ADC ADC ASL ORA STA ASL ORA STA CMP BNE LDA JSR LDY	A 3 #\$04 #\$04 BASH #\$18 BASLC2 #\$77 BASL BASL BASL BASL BASL #\$87 RTS28 #\$40 WAIT #\$00	<pre>;CALCBASE ADDR IN BASL,H ;FUR GIVEN LINE ND. ;O<=LINE ND. <=\$17 ;ARG=000ABCDE, GENERATE ; BASH=000001CD ; AND ; BASL=EABAB000 ;BASL=EABAB000 ;BELL CHAR? (CONTROL-G) ;ND. RETURN. ;YES ;DELAY.01 SECONDS</pre>
FBED:D0 FS FBE4 126 BNE BELL2 FBEF:60 127 RTS RTS FBF:60 127 RTS STORADV LDY CH ; CURSOR H INDEX TO Y-REG FBF:60 127 RTS (BASL), y ; STORE CHAR IN LINE FBF:42 128 D29 STA (BASL), y ; STORE CHAR IN LINE FBF:42 130 ADVANCE INC CH ; INCREMENT CURSOR H INDEX FBF:54 131 LDA CH ; IMOVE RIGHT) FBEYOND HINDOH WIDTH? FBF:66 131 RTS ; NG. RETURN. FBFS FBF:66 133 RTS3 RTS ; NG. NETURN. FBF:66 134 RTS3 RTS ; NG. OUTPUT IN. FBF:06 134 RTS3 RTS ; INVERSE VIDEO? FC02:10 EC FBF BF0 136 BCS STORADV ; INVERSE VIDEO? FC02:10 EC FBF 138 CMP ##BD ; INV	FBC1: FBC21:40 FBC2:40 FBC3:29 03 FBC3:09 04 FBC7:85 29 FBC7:65 FBC3:09 02 FBC4:07 02 FBC2:00 02 FBC2:00 02 FBD2:0A FBD3:0A FBD4:05 28 FBD2:0A FBD8:50 FBD9:09 07 FBD9:09 07 FBD9:20 07 FBD	FBEF	$\begin{array}{c} 101\\ 102\\ 103\\ 104\\ 105\\ 106\\ 107\\ 108\\ 107\\ 110\\ 110\\ 111\\ 112\\ 1113\\ 114\\ 115\\ 1116\\ 117\\ 118\\ 117\\ 121\\ 121\\ 122 \end{array}$	BASCALC BASCLC2 BELL1	FIN RG \$FBC PHA LSR AND ORA STA PLA AND BCC ADC STA ASL ORA STA ASL ORA STA LDA JSR LDA	* 403 #\$03 BASH #\$18 BASCLC2 #\$77 BASL AA A BASL BASL BASL BASL BASL BASL	<pre>;CALCBASE ADDR IN BASL,H ;FOR GIVEN LINE ND. ; O<=LINE ND.<=\$17 ;ARG=000ABCDE, GENERATE ; BASH=000001CD ; AND ; BASL=EABAB000 ;BASL=EABAB000 ;BASL=EABAB000 ;BASL=EABAB000 ;SELL CHAR? (CONTROL-G) ;ND, RETURN. ;YES ;DELAY.01 SECONDS ;TOGGLE SPEAKER AT 1 KHZ</pre>
FBEF: 60 127 RT528 RT5 FBF0: A4 24 128 STORADV LDY CH ; CURSOR H INDEX TO Y-REG FBF2: 91 28 128 STORADV LDY CH ; STORE CHAR IN LINE FBF2: 91 28 129 ADVANCE INC CH ; STORE CHAR IN LINE FBF4: 82 24 130 ADVANCE INC CH ; INCREMENT CURSOR H INDEX FBF6: 85 24 131 LDA CH ; (MOVE RIGHT) FBF6: 85 24 132 CMP WADWDTH ; BEYOND WIDTH? FBF6: 80 66 F626 133 BCS CR ; YES. CT TO NEXT LINE. FBF7: 80 66 F646 133 RTS3 ; NO. RETURN. ; FBF5: 60 FBF7: 80 EF FFF0 135 VIDUT CMP #\$A0 ; CONTROL CHAR? FBF5: 80 EF FFF0 138 BPL STORADV ; NO. OUTPUT IT. FC02: 10 EC FBF0 139 CMP #\$A0 ; CR? FC04: C9 3A FC62 140 BQ CR ; YES. FC06: C9 A A 141 CMP #\$A40 ; LINE FEED?	FBC1: FBC21:48 FBC2:4A FBC3:29 03 FBC3:09 04 FBC7:65 29 FBC7:65 FBC3:60 02 FBC4:07 02 FBC2:00 02 FBC2:00 02 FBD2:05 28 FBD2:05 FBD3:00 FBD4:05 28 FBD2:65 28 FBD2:50 49 FBD5:20 48 FBD2:20	FBEF FC FC	101 102 103 104 105 106 107 108 107 110 110 111 112 113 114 115 116 117 118 119 120 121 122 123	BASCALC BASCLC2 BELL1	FIN G \$FBC PHA LSR ORA STA AND BCC STA AND BCC STA ASL ORA STA ASL ORA STA JSR LDA JSP LDA	4 0 # 904 # 904 BASH # 918 BASL BASL BASL BASL BASL BASL BASL BASL BASL WAIT # 900 #	<pre>;CALCBASE ADDR IN BASL,H ;FOR GIVEN LINE ND. ; O<=LINE ND.<=\$17 ;ARG=000ABCDE, GENERATE ; BASH=000001CD ; AND ; BASL=EABAB000 ;BASL=EABAB000 ;BASL=EABAB000 ;BASL=EABAB000 ;SELL CHAR? (CONTROL-G) ;ND, RETURN. ;YES ;DELAY.01 SECONDS ;TOGGLE SPEAKER AT 1 KHZ</pre>
FBF0:A4 24 128 STORADV LDY CH ;CURSDR H INDEX TO Y-REG FBF2:91:28 129 STA (BASL); ;STORE CHAR IN LINE FBF4:46:24 130 ADVANCE INC CH ;INCREMENT CURSOR H INDEX FBF6:45:24 131 LDA CH ;INCREMENT CURSOR H INDEX FBF6:55:24 132 CMP WINDWDTH ;BEYOND WIDTH? FBF6:66:66 133 RTS RTS ;ND, RETURN. FBF5:06:60 134 RTS3 RTS ;ND, OUTPUT IT. FBF7:06:6F FBF0 136 BCS STORADV FBF7:06:6F 136 BCS STORADV ;NURERSE VIDEO? FC02:10:EC FB7 139 CMP ##BD FC04:C9 139 STORADV ;VES. CUTPUT IT. FC04:C9 5A C44 EQ CMP ##BC FC04:C9 5A C44 EQ F FSD OD TT. FC04:C9 FA 143 CMP ##BE ;BACK SPACE? (CONTROL-H) FC06:C9 FB6 143 CMP ##BE ;BACK SPACE? (CONTROL-H)	FBC1: FBC2:40 FBC2:40 FBC3:27 03 FBC3:27 03 FBC3:27 03 FBC3:27 03 FBC3:27 18 FBC3:27 18 FBC3:27 18 FBC3:27 18 FBC3:27 18 FBD3:04 28 FBD3:05 28 FBD3:05 28 FBD3:50 12 FBD5:50 49 FBD5:20 48 FBD5:20 48 FBD5:20 48 FBC3:20 48	FBEF FC FC CO	101 102 103 104 105 106 107 108 107 110 110 111 112 113 114 115 116 117 118 117 120 121 122 123 124 125	BASCALC BASCLC2 BELL1	FIN G \$FBC PHA LSR AND STA AND BCC STA ASL ORA STA ASL ORA STA STA JSR LDA JSR LDA JSR LDA JSR	A G #903 #804 BASH #\$18 BASLC2 #\$77 BASL BASL BASL BASL BASL BASL BASL BASL	<pre>;CALCBASE ADDR IN BASL,H ;FOR GIVEN LINE ND. ; O<=LINE ND <=%17 ;ARG=000ABCDE, GENERATE ;BASH=000001CD ;AND ;BASL=EABAB000 ;BASL=EABAB000 ;BASL=EABAB000 ;BASL=EABAB000 ;JOLAY.01 SECONDS ;JOLAY.01 SECONDS ;JOGGLE SPEAKER AT 1 KHZ</pre>
FBF2:91 28 129 STA (BASL),Y ;STORE CHAR IN LINE FBF4:62 130 ADVANCE INC CH ;INCREMENT CURSOR H INDEX FBF6:63 24 130 ADVANCE INC CH ;INCREMENT CURSOR H INDEX FBF6:63 24 131 LDA CH ;INCREMENT CURSOR H INDEX FBF6:63 24 132 CMP WADWDTH ;BEYOND WIDTH? FBF8:60 64 F642 133 BCS CR ;YES. CR TO NEXT LINE. FBF5:60 134 RTS3 RTS ; NO. RETURN. FBF5:70 64 F642 BCS STORADV ; NO. OUTPUT IT. FC02:10 EC FBF0 138 BPL STORADV ; VES. FC02:10 EC FBF0 CMP #\$8D ; CR? FC04:C9 BD 139 CMP #\$8D ; INFEFED? FC04:C9 SA F642 H0 BCQ CR ; YES. FC04:C9 SA F43 CMP #\$8A ; INFEFED? FC04:C9 SA F642 BCQ INF ; ISOL FC04:C9 SA F43 CMP #\$8A ; INF	FBC1: FBC1:48 FBC2:4A FBC3:27 03 FBC3:07 04 FBC3:05 24 FBC3:08 FBC3:08 FBC3:08 FBC3:00 FBC4:07 02 FBD2:0A FBD4:05 28 FBD2:0A FBD4:05 28 FBD2:0A FBD4:05 28 FBD8:00 12 FBD8:00 12 FBD8:00 12 FBD8:20 A9 FBD8:20 A9 FBC3:20 A9	FBEF FC FC CO	101 102 103 104 105 106 107 108 107 110 111 112 113 114 115 116 117 1120 121 123 124 125 126	BASCALC BASCLC2 BELL1 BELL2	FIN G \$FBC PHA LSR ORA STA AND BCC STA ASL ORA STA ASL ORA STA ASL ORA STA ASL ORA DENE LDA LDA LDA DENE LDA DENE	A G #903 #804 BASH #\$18 BASLC2 #\$77 BASL BASL BASL BASL BASL BASL BASL BASL	<pre>;CALCBASE ADDR IN BASL,H ;FOR GIVEN LINE ND. ; O<=LINE ND <=%17 ;ARG=000ABCDE, GENERATE ;BASH=000001CD ;AND ;BASL=EABAB000 ;BASL=EABAB000 ;BASL=EABAB000 ;BASL=EABAB000 ;JOLAY.01 SECONDS ;JOLAY.01 SECONDS ;JOGGLE SPEAKER AT 1 KHZ</pre>
FBF4:E6 24 130 ADVANCE INC CH ; INCREMENT CURSOR H INDEX FBF6:A5 24 131 LDA CH ; (MOVE RIGHT) FBF8:C5 21 132 CMP WINDWDTH ; BEYOND WIDTH? FBF8:A5 24 133 BCS CR ; PES, CR TO NEXT LINE. FBF6:A5 26 133 BCS CR ; PES, CR TO NEXT LINE. FBF0:C5 00 135 VIDOUT CMP #\$400 ; CONTROL CHAR? FBFF:B0 EF FBF0 135 VIDOUT TAY ; INVERSE VIDEO? FC02:10 EC FBF0 138 BPL STORADV ; NUSCRE VIDEO? FC02:10 EC FBF0 139 CMP #\$40 ; CR? FC04:C9 BD 139 CMP #\$40 ; CR? FC04:C9 SA F40 EGQ CR ; YES. FC04:C9 SA F44 LMP #\$40 ; LINE FEED? FC04:C9 SB 143 CMP #\$48 ; BACK SPACE? (CONTROL-H) FC05:C9 C9 FB9 144 BNE ELL1 ; NO, CHCK FOR BELL. <td>FBC1: FBC2:40 FBC2:40 FBC3:27 03 FBC3:27 03 FBC3:27 03 FBC3:27 03 FBC3:27 18 FBC3:27 18 FBC3:27 18 FBC4:27 18 FBC4:27 18 FBD3:04 27 FBD3:05 28 FBD3:05 28 FBD3:05 28 FBD3:50 28 FBD3:50 28 FBD3:20 48 FBD3:20 48 FBD3:20 48 FBD3:20 48 FBD3:20 48 FBC3:20 48</td> <td>FBEF FC FC CO</td> <td>101 102 103 104 105 106 107 110 111 112 113 114 115 114 115 114 117 120 121 123 124 123 124 125</td> <td>BASCALC BASCLC2 BELL1 BELL2 RTS2B</td> <td>FIN G HA HA LSR AND ORA STA AND ORA STA AND AND AND AND AND AND AND AN</td> <td>A G #903 #804 BASH #\$18 BASLC22 #\$77 BASL A A A BASL BASL BASL BASL BASL #\$40 WAIT SPKR BELL2</td> <td><pre>CALCBASE ADDR IN BASL,H FOR GIVEN LINE ND. Collected of the collected of the collected of the collected BASH=0000ABCDE. GENERATE BASH=0000001CD AND BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB0000 BASL=EABAB000001CD BASL=EABAB000001CD BASL=EABAB000001CD BASL=EABAB000001CD BASL=EABAB000001CD BASL=EABAB0000001CD BASL=EABAB0000001CD BASL=EABAB0000001CD BASL=EABAB0000001CD BASL=EABAB0000001CD BASL=EABAB0000000000 BASL=EABAB00000000000 BASL=EABAB00000000000 BASL=EABAB000000000000 BASL=EABAB000000000000 BASL=EABAB00000000000000000000000 BASL=EABAB0000000000000000000000000000000000</pre></td>	FBC1: FBC2:40 FBC2:40 FBC3:27 03 FBC3:27 03 FBC3:27 03 FBC3:27 03 FBC3:27 18 FBC3:27 18 FBC3:27 18 FBC4:27 18 FBC4:27 18 FBD3:04 27 FBD3:05 28 FBD3:05 28 FBD3:05 28 FBD3:50 28 FBD3:50 28 FBD3:20 48 FBD3:20 48 FBD3:20 48 FBD3:20 48 FBD3:20 48 FBC3:20 48	FBEF FC FC CO	101 102 103 104 105 106 107 110 111 112 113 114 115 114 115 114 117 120 121 123 124 123 124 125	BASCALC BASCLC2 BELL1 BELL2 RTS2B	FIN G HA HA LSR AND ORA STA AND ORA STA AND AND AND AND AND AND AND AN	A G #903 #804 BASH #\$18 BASLC22 #\$77 BASL A A A BASL BASL BASL BASL BASL #\$40 WAIT SPKR BELL2	<pre>CALCBASE ADDR IN BASL,H FOR GIVEN LINE ND. Collected of the collected of the collected of the collected BASH=0000ABCDE. GENERATE BASH=0000001CD AND BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB000 BASL=EABAB0000 BASL=EABAB000001CD BASL=EABAB000001CD BASL=EABAB000001CD BASL=EABAB000001CD BASL=EABAB000001CD BASL=EABAB0000001CD BASL=EABAB0000001CD BASL=EABAB0000001CD BASL=EABAB0000001CD BASL=EABAB0000001CD BASL=EABAB0000000000 BASL=EABAB00000000000 BASL=EABAB00000000000 BASL=EABAB000000000000 BASL=EABAB000000000000 BASL=EABAB00000000000000000000000 BASL=EABAB0000000000000000000000000000000000</pre>
FBF8:C5 21 132 CMP WNDWDTH ;BEYOND WIDDWTP FBFA:B0 64 F642 133 BCS CR ;YES, CR TO NEXT LINE. FBFD:C9 A0 135 VIDDUT CMP ***A0 ;CONTROL CHAR? FBFD:C9 A0 135 VIDDUT CMP ***A0 ;CONTROL CHAR? FBFD:C9 A0 135 VIDDUT CMP ***A0 ;CONTROL CHAR? FBFD:C9 A0 135 VIDOUT CMP ***A0 ;CONTROL CHAR? FC01:A8 137 TAY ;INVERSE VIDEO? ; FC02:10 EC FBFD 138 BPL STURADV ;YES, OUTPUT IT. FC04:C9 BD 139 CMP ***A0 ;CR? ; FC04:C9 C5 AF C626 141 CMP ***B4 ;LINE FEED? FC04:C9 BB 143 CMP ***B8 ;BACK SPACE? (CONTROL-H) FC05:D0 C9 FBD9 144 BNE SELL1 ;NO, CHECK FOR BELL,	FBC1: FBC2:40 FBC2:40 FBC3:27 03 FBC5:07 04 FBC7:85 27 FBC7:85 27 FBC7:85 27 FBC7:85 27 FBC7:85 27 FBC3:07 FBD	FBEF FC FC CO	101 102 103 104 105 106 107 110 110 111 112 113 114 115 116 117 118 117 118 117 120 121 122 123 124 125 124	BASCALC BASCLC2 BELL1 BELL2 RTS2B STORADV	FIN G \$FBC PHA LSR LSR ORA STA AND CORA STA ANDC ACC STA ANDC STA ASL ORA STA ANDC STA ASL ORA STA ANDC STA ASL ORA STA ANDC STA ASL ORA STA ANDC STA ASL ORA STA ANDC STA ASL ORA STA ADDC STA ASL ORA STA ADDC STA ASL ORA STA ADDC STA ASL ORA STA ADDC STA ASL ORA STA ADDC STA ASL ORA STA ADDC STA ASL ORA STA ADDC STA ASL DORA STA ADDC STA ASL DORA STA ASL ORA STA ASL ORA STA ASL DORA STA ADDC STA ASL DORA STA ASL DORA STA ASL DORA STA ASL DORA STA ASL DORA STA ASL DORA STA ASL DORA STA ASL DORA STA LDA DA STA LDA STA STA STA STA STA STA STA ST	A #\$03 #\$04 BASH #\$18 BASLC2 #\$77 BASL A A A A A A A A A A A A A A A A A A	<pre>;CALCBASE ADDR IN BASL,H ;FOR GIVEN LINE ND. ; O<=LINE ND.(=\$17 ;ARG=000ABCDE, GENERATE ;BASH=000001CD ;AND ;BASL=EABAB000 ;BASL=EABAB000 ;BASL=EABAB000 ;BCURN. ;YES ;DELAY.01 SECNDS ;TOGQLE SPEAKER AT 1 KHZ ;FOR .1 SEC. ;CURSOR H INDEX TO Y-RE0 ;STORE CHAR IN LINE</pre>
FBFA:B0 66 FC62 133 BCS CR ; YES, CR YES, CR TO NEXT LINE. FBFC:60 134 RTS3 RTS ; NO, RETURN. FBFD:C9 A0 135 VIDUT CMP #\$40 ; CONTROL CHAR? FBFD:C9 A0 135 VIDUT CMP #\$40 ; CONTROL CHAR? FBFF:B0 EF FBF0 136 BCS STORADV ; NO, DUTPUT IT. FC02:10 EC FBF0 138 BPL STORADV ; VES, OUTPUT IT. FC04:C9 BD 137 CMP #\$80 ; CR? FC06:F0 5A FC62 140 BEG CR ; YES. FC08:C9 BA 141 CMP #\$80 ; LINE FEED? FC04:C9 5A FC62 140 BEG CR ; YES. FC08:C9 SA 141 CMP #\$80 ; LINE FEED? FC04:C9 5A FC64 142 BQ	FBC1: 49 FBC2: 44 FBC2: 27 03 FBC5: 07 04 FBC7: 85 29 FBC7: 85 29 FBC7: 85 29 FBC7: 85 29 FBC7: 60 FBC7: 60 FBC7: 60 FBD2: 65 28 FBD2: 0A FBD4: 65 28 FBD2: 0A FBD4: 65 28 FBD8: 00 FBC7: 60 FBC7: 60 FBC7: 40 CFBC2: 88 FBC2: 91 28 FBC2: 91 28 FBC4: 20 42 FBC7: 91 28 FBC7: 91 28 F	FBEF FC FC CO	101 102 103 104 105 106 107 108 110 110 111 112 113 114 115 114 117 118 117 118 117 122 123 124 123 124 127 128 127 128 127 128 127 128 127 130	BASCALC BASCLC2 BELL1 BELL2 RTS2B STORADV ADVANCE	FIN GC SFREC PHA LSR LSR AND ORA ABCC ADCC ADCC ACC ACC ACC ACC ACC	A #\$03 #\$04 BASH #\$18 BASLC2 #\$77 BASL A A BASL BASL BASL #\$87 RTS28 #\$40 WAIT SPKR BELL2 CH (BASL), Y CH	<pre>;CALCBASE ADDR IN BASL,H ;FDR GIVEN LINE ND. ; O<=LINE ND.<=%17 ;ARG=0000ABCDE, GENERATE ; BASH=000001CD ; AND ; BASL=EABAB000 ;BASL=EABAB000 ;BASL=EABAB000 ;BASL=EABAB000 ;BASL=EABAB000 ;DASL=EABAB0000 ;DASL=EABAB0000 ;DASL=EABAB0000 ;DASL=EABAB0000 ;DASL=EABAB0000 ;DASL=EABAB0000 ;DASL=EABAB0000 ;DASL=EABAB0000000000000000000000000000000000</pre>
FBFD:C9 A0 135 VIDUT CMP ##A0 ;CONTROL CHAR? FBFF:B0 EF FBF0 136 BCS STORADV ; NO, OUTPUT IT. FC01:A8 137 TAY ; INVERSE VIDEO? FC02:10 EC FBF0 138 BPL STORADV ; VES. OUTPUT IT. FC04:C9 BD 139 CMP ##80 ; CR? FC06:F0 5A FC62 140 BEQ CR ; YES. FC08:C9 6A 141 CMP #\$8A ; LINE FEED? FC04:F0 5A FC66 142 BEQ LF ; IF S0, D0 IT. FC04:F0 5A FC66 142 BEQ LF ; DO IT. FC04:F0 5A FC66 142 BEQ LF ; DO IT. FC04:F0 5A FC66 142 BEQ LF ; DO IT. FC04:F0 5A FC66 143 CMP #\$88 ; BACK SPACE? (CONTROL-H) FC06:F0 F08 143 DN DC MECK FOR BELL. ; NO, CHECK FOR BELL.	FBC1: FBC2:40 FBC2:40 FBC3:27 03 FBC3:27 03 FBC5:52 7 FBC7:85 27 FBC7:85 27 FBC7:85 27 FBC7:85 27 FBC3:27 18 FBC2:47 18 FBC2:47 18 FBC3:47 16 FBD3:0A 28 FBD3:0A 28 FBD3:40 FBD3:40 FBD3:40 FBD3:40 FBD3:40 FBD3:40 FBC3:40 FBC3:40 FBC3:40 FBC3:40 FBC3:45 24 FBC3:45 24 FBC3:45 24	FBEF FC FC CO	101 102 103 104 105 104 105 106 107 108 107 108 107 110 111 112 113 114 115 114 115 124 123 124 125 124 127 128 127 128 127 128 127 128 127 128 127 128 127 128 127 128 129 129 129 129 129 129 129 129 129 129	BASCALC BASCLC2 BELL1 BELL2 RTS2B STORADV ADVANCE	FIN SC SFFBC PHA LSR AND ORA PLSR AND ORA PLAD BCCC ASTA ASL ORA ASL JSR LDA JSR LDA JSRA LDA DEYE RTSY STA LDA	A 3 #903 #804 BASH #818 BASLC2 #877 BASL A A A BASL BASL BASL BASL #807 RT528 #840 WAIT #800 WAIT SPKR BELL2 CH (BASL),Y CH	<pre>;CALCBASE ADDR IN BASL,H ;FOR GIVEN LINE ND. ; O<=LINE ND.(=\$17 ;ARG=00ABCDE, GENERATE ;BASH=000001CD ;AND ;BASL=EABAB000 ;BASL=EABAB000 ;BASL=EABAB000 ;BASL=EABAB000 ;UD, RETURN. ;YES ;DELAY.01 SECONDS ;TOGQLE SPEAKER AT 1 KHZ ;FOR .1 SEC. ;CURSOR H INDEX TO Y-REG ;STORE CHAR IN LINE ;INCREMENT CURSOR H INDEX ; (MOVE RIGHT)</pre>
FBFD:C9 A0 135 VIDUT CMP ##A0 ;CONTROL CHAR? FBFF:B0 EF FBF0 136 BCS STORADV ; NO, OUTPUT IT. FC01:A8 137 TAY ; INVERSE VIDEO? FC02:10 EC FBF0 138 BPL STORADV ; VES. OUTPUT IT. FC04:C9 BD 139 CMP ##80 ; CR? FC06:F0 5A FC62 140 BEQ CR ; YES. FC08:C9 6A 141 CMP #\$8A ; LINE FEED? FC04:F0 5A FC66 142 BEQ LF ; IF S0, D0 IT. FC04:F0 5A FC66 142 BEQ LF ; DO IT. FC04:F0 5A FC66 142 BEQ LF ; DO IT. FC04:F0 5A FC66 142 BEQ LF ; DO IT. FC04:F0 5A FC66 143 CMP #\$88 ; BACK SPACE? (CONTROL-H) FC06:F0 F08 143 DN DC MECK FOR BELL. ; NO, CHECK FOR BELL.	FBC1: 49 FBC2: 40 FBC2: 27 03 FBC5: 07 04 FBC7: 55 29 FBC7: 55 29 FBC7: 56 29 FBC7: 50 02 FBC7: 50 02 FBC7: 50 02 FBC5: 50 05 FBD3: 00 55 FBD3: 00 FBD4: 55 28 FBD8: 20 45 FBD9: C7 97 FBD8: 20 45 FBD9: 20 45 FBD7: 20 48 FB22: 40 00 FBE4: 40 00 FBE4: 40 00 FBE4: 40 05 FBE7: 40 424 FBF2: 41 424 FBF4: 45 44 FBF4: 45 44 FBF6: 55 21	FBEF FC FC CO FBE4	101 102 103 104 105 106 107 108 107 108 107 110 111 113 114 115 113 114 115 120 121 123 124 125 124 127 128 127 128 129 130 131 132	BASCALC BASCLC2 BELL1 BELL2 RTS2B STURADV ADVANCE	FIN EC FIN EC PHA LSR BC ADRA	A #\$04 #\$04 BASH #\$18 BASL #\$77 BASL A A BASL BASL BASL BASL BASL BASL BASL BASL BASL BASL BASL BASL CL CL CH CH CH CH	<pre>;CALCBASE ADDR IN BASL,H ;FOR GIVEN LINE ND. ; O<=LINE ND. <=%17 ;ARG=000ABCDE, GENERATE ;BASH=000001CD ;ARD ;BASL=EABAB000 ;BASL=EABAB000 ;BASL=EABAB000 ;BASL=EABAB000 ;BASL=EABAB000 ;CURSOR H INDEX TO Y=REG ;STORG CHAR IN LINE ;INCREMENT CURSOR H INDEX ;(MOVE RIGHT)A; ;EXUMPLOW HIDTH? ;EXUMPLOW HIDTH? ;CALCBASE ADDR INDEX ;CALCBASE ADDR</pre>
FBFF:B0 EFF FBFO 136 BCS STORADV ; NO. DUTPUT IT. FC01:A8 137 TAY ; INVERSE VIDEO? FC02:10 EC FBFO 138 BPL STORADV ; YES, DUTPUT IT. FC04:C9 BD 139 CMP ##8D ; CR? FC08:C9 BA FC62 140 BEQ CR ; YES. FC08:C9 BA 141 CMP ##8A ; LINE FEED? FC04:C5 SA FC64 142 EQ LF ; INE FEED? FC04:C5 SA FC64 142 EQ LF ; INE FEED? FC04:C5 SA FC46 142 EQ LF ; INE FEED? FC04:F0 SA FC46 143 CMP ##88 ; BACK SPACK: PACK: PA	FBC1: FBC1: FBC1: FBC2:40 FBC3:29 FBC3:29 FBC3:29 FBC3:29 FBC3:29 FBC3:29 FBC3:29 FBC3:29 FBC3:29 FBC3:20 FBC2:00 FBD3:0A FBD4:05 FBD3:0A FBD5:0B FBD5:0A FBD5:0D FBD7:0A FBE7:0A FBE7:0A FBE7:0A FBF7:1A FBF6:A2 FBF7:1A FBF4:45 FBF4:45 FBF4:52 FBF4:52 FBF4:50 FBF4:50 FBF4:50 FBF4:	FBEF FC FC CO FBE4	101 102 103 104 105 106 107 110 111 112 113 114 115 116 117 120 123 124 125 124 127 128 127 128 127 128 127 130 131 133	BASCALC BASCLC2 BELL1 BELL2 RTS2B STORADV ADVANCE	FIN DEVELOPMENT PHA LSR AND AND AND ARA PLA BCCC STA ASSL JSR BNEA STA LDY LDY LDA RTS LDA RTS LDA ENT STA LDA RTS RTS RTS RTS RTS LDA RTS RTS RTS RTS RTS RTS RTS RTS RTS RTS	A #\$04 #\$04 BASH #\$18 BASL #\$77 BASL A A BASL BASL BASL BASL BASL BASL BASL BASL BASL BASL BASL BASL CL CL CH CH CH CH	<pre>;CALCBASE ADDR IN BASL,H ;FOR GIVEN LINE ND. ; O<=LINE ND. <=%17 ;ARG=000ABCDE, GENERATE ;BASH=000001CD ;ARD ;BASL=EABAB000 ;BASL=EABAB000 ;BASL=EABAB000 ;BASL=EABAB000 ;BASL=EABAB000 ;CURSOR H INDEX TO Y=REG ;STORG CHAR IN LINE ;INCREMENT CURSOR H INDEX ;(MOVE RIGHT)A; ;EXUMPLOW HIDTH? ;EXUMPLOW HIDTH? ;CALCBASE ADDR INDEX ;CALCBASE ADDR</pre>
FC02:10 EC FFFO 138 BPL STORADV ; YES, DUTPUT IT. FC04:C9 BD 139 CMP ##BD ; CR? FC04:C9 BD 139 CMP #BED ; CR? FC06:C9 BD 140 BEG CR ; YES. FC08:C9 BA 141 CMP #BEA ; LINE FEED? FC04:C9 SA C42 BG LF ; ISO.D0 IT. FC06:C9 SB 143 CMP #BEB ; BACK SPACK: 2 (CONTROL-H) FC06:D0 C7 FBD9 144 BNE BELL1 ; NO. CHECK FOR BELL.	FBC1: FBC1: FBC1: FBC2:4A FBC3:29 03 FBC3:29 03 FBC3:29 03 FBC3:29 02 FBC3:20 02 FBC5:09 02 FBC2:00 02 FBC5:528 FBD2:05 28 FBD3:00 FBD4:55 28 FBD5:00 12 FBD2:00 12 FBD2:00 12 FBD2:00 4 FBD2:00 75 FBE2:00 05 FBE2:00 55 FBE7:40 05 FBE7:40 5 FBE7:40 4 FBE7:40 5 FBE7:40 5 FBE7:40 5 FBE7:40 5 FBE7:40 5 FBF2:51 20 45 FBF2:41 26 FBF2:51 26 FBF2:52 31 26 FBF2:50 6 FBF2:50 6 FBF2:50 16 FBF2:60 5 FBF2:50 16 FBF2:50 16 FBF2:50 16 FBF2:50 7 FBF2:50 7 FBF3:50 5 FBF3:50 5 <tr< td=""><td>FBEF FC FC CO FBE4</td><td>101 102 103 104 105 104 107 108 107 108 107 110 111 112 113 114 115 113 114 115 121 122 123 124 127 128 127 130 131 133</td><td>BASCALC BASCLC2 BELL1 BELL2 RTS2B STORADV ADVANCE RTS3</td><td>FIN DESTA</td><td>A 3 #\$04 #\$04 BASH #\$18 BASLC22 #\$77 BASL A A A BASL BASL BASL BASL BASL BASL BASL BASL</td><td><pre>; CALCBASE ADDR IN BASL,H ; FOR GIVEN LINE ND. ; O<=LINE ND.<=\$17 ; ARG=000ABCDE, GENERATE ; BASH=000001CD ; AND ; BASL=EABAB000 ; BASL=EABAB000 ; BASL=EABAB000 ; BASL=EABAB000 ; DGRUP, CONTROL-G) ; ND, RETURN. ; YES ; DELAY. 01 SECONDS ; TOGGLE SPEAKER AT 1 KHZ ; FOR .1 SEC. ; CURSOR H INDEX TO Y-REG ; STORE CHAR IN LINE ; INCREMENT CURSOR H INDEX ; (MOVE RIGHT) ; YES. CT 0 NEXT LINE ; ND, RETURN. ; YES, CT 0 NEXT LINE ; ND, RETURN. ; YES, CT 0 NEXT LINE ; ND, RETURN. ; CONTROL CHAR?</pre></td></tr<>	FBEF FC FC CO FBE4	101 102 103 104 105 104 107 108 107 108 107 110 111 112 113 114 115 113 114 115 121 122 123 124 127 128 127 130 131 133	BASCALC BASCLC2 BELL1 BELL2 RTS2B STORADV ADVANCE RTS3	FIN DESTA	A 3 #\$04 #\$04 BASH #\$18 BASLC22 #\$77 BASL A A A BASL BASL BASL BASL BASL BASL BASL BASL	<pre>; CALCBASE ADDR IN BASL,H ; FOR GIVEN LINE ND. ; O<=LINE ND.<=\$17 ; ARG=000ABCDE, GENERATE ; BASH=000001CD ; AND ; BASL=EABAB000 ; BASL=EABAB000 ; BASL=EABAB000 ; BASL=EABAB000 ; DGRUP, CONTROL-G) ; ND, RETURN. ; YES ; DELAY. 01 SECONDS ; TOGGLE SPEAKER AT 1 KHZ ; FOR .1 SEC. ; CURSOR H INDEX TO Y-REG ; STORE CHAR IN LINE ; INCREMENT CURSOR H INDEX ; (MOVE RIGHT) ; YES. CT 0 NEXT LINE ; ND, RETURN. ; YES, CT 0 NEXT LINE ; ND, RETURN. ; YES, CT 0 NEXT LINE ; ND, RETURN. ; CONTROL CHAR?</pre>
FC04:C9 BD 139 CMP ##8D ;CR? FC04:F0 5A FC62 140 BEQ CR ; YES. FC08:F0 5A FC62 141 CMP ##8D ; LINE FEED? FC08:F0 5A FC66 142 BEQ LF ; IS FD D TT. FC00:C9 8B 143 CMP #\$88 ; BACK SPACE? (CONTROL-H) FC0E:D0 C9 FBD 144 BNE BELL1 ; NO, CHECK FOR BELL.	FBC1: FBC1: FBC1:48 FBC2:40 FBC3:27 FBD3:05 FBD3:00 FBD3:00 FBD3:01 FBD	FBEF FC FC FBE4 FC62	101 102 103 104 105 107 108 107 110 111 112 113 114 115 114 115 114 115 114 117 122 123 124 125 126 127 128 129 131 132 133 134 135	BASCALC BASCLC2 BELL1 BELL2 RTS2B STORADV ADVANCE RTS3 VIDDUT	FIN DESCRIPTION OF A DE	A #903 BASH #\$18 BASLC2 #\$77 BASL A A A BASL BASL BASL BASL BASL BAS	<pre>CALCBASE ADDR IN BASL,H ; FOR GIVEN LINE ND. ; O<=LINE ND. <=\$17 ; ARG=000ABCDE, GENERATE ; BASH=000001CD ; AND ; BASL=EABAB000 ; BASL=EABAB000 ; BASL=EABAB000 ; BASL=EABAB000 ; DELAY. 01 SECONDS ; TOGQLE SPEAKER AT 1 KHZ ; FOR.1 SEC. ; CURSOR H INDEX TO Y-REG ; STORE CHAR IN LINE ; INCREMENT CURSOR H INDEX ; (MOVE RIGHT) ; YES.CR TO NEXT LINE.; ND, RETURN. ; CONTROL CHAR? ; ND, OUTPUT IT.</pre>
FC06:F0 5A FC62 140 BEQ CR ; YES. FC08:C9 6A 141 CMP #\$8A ; LINE FEED? FC04:F0 5A FC66 142 BEQ LF ; IF 50, DO IT. FC06:C9 68 143 CMP #\$88 ; BACK SPACE? (CONTROL-H) FC06:C9 (57 FB09 144 BNE BELL1 ; NO. CHECK FOR BELL.	FBC1: FBC1: FBC1: FBC2:40 FBC3:27 FBC3:27 FBC3:27 FBC3:27 FBC3:27 FBC3:42 FBC4:27 FBC3:28 FBC2:00 FBC2:00 FBD2:05 FBD2:05 FBD4:05 FBD2:04 FB2:02	FBEF FC FC FBE4 FBE4 FBF0	101 102 103 104 105 106 107 108 110 111 112 113 114 115 114 117 118 117 112 121 123 124 125 128 129 130 131 132 134 135 136	BASCALC BASCLC2 BELL1 BELL2 RTS2B STORADV ADVANCE RTS3 VIDOUT	FIN DESCRIPTION OF THE SECONDARY SEC	A #03 #504 BASH #\$18 BASLC2 #\$77 BASL A A BASL BASL BASL BASL BASL BASL BASL BASL BASL BASL C2 BASL C2 BASL C2 BASL C2 BASL C2 BASL C2 C3 C3 C4 C4 C4 C4 C4 C7 C4 C7 C7 C7 C7 C7 C7 C7 C7 C7 C7	<pre>; CALCBASE ADDR IN BASL,H ; FOR GIVEN LINE ND. ; O<=LINE ND.<=\$17 ; ARG=000ABCDE, GENERATE ; BASH=000001CD ; AND ; BASL=EABAB000 ; BASL=EABAB000 ; BASL=EABAB000 ; BASL=EABAB000 ; DGLAY .01 SECONDS ; TOGGLE SPEAKER AT 1 KHZ ; FOR .1 SEC. ; CURSOR H INDEX TO Y-REG ; STORE CHAR IN LINE ; INCREMENT CURSOR H INDEX ; (MOVE RIGHT) ; BELAY .01 SEC. ; CURSOR H INDEX TO Y-REG ; STORE CHAR IN LINE ; INCREMENT CURSOR H INDEX ; (MOVE RIGHT) ; BELAY .01 NED ; CONTROL CHAR? ; NO. RETURN. ; NO. GUTPUT IT. ; INVERSE UIDED?</pre>
FC08:C9 BA 141 CMP #\$8A ; LINE FEED? FC04:F0 5A FC66 142 BEQ LF ; IF SD. DD IT. FC00:C9 8B 143 CMP #\$88 ; BACK SPACE? (CONTROL-H) FC06:D0 C9 FBD9 144 BNE BELL1 ; NO, CHECK FOR BELL.	FBC1: FBC1: FBC1:48 FBC2:40 FBC3:27 FBD3:05 FBD3:00 FBD3:00 FBD3:01 FBC3:20 FBE3:20 FB52:20 FB52:21 FB52:21 FB5	FBEF FC FC FBE4 FBE4 FBF0	101 102 103 104 105 106 107 110 112 113 114 115 114 115 114 117 120 121 123 124 125 124 127 128 127 128 127 128 129 131 132 134 135 136	BASCALC BASCLC2 BELL1 BELL2 RTS2B STORADV ADVANCE RTS3 VIDOUT	FIN DESCRIPTION OF A DE	A 4 4 4 4 4 4 4 4 5 4 5 5 5 5 5 5 5 5 5 5 5 5 5	<pre>CALCBASE ADDR IN BASL,H FOR GIVEN LINE ND. GALCBASE ADDR IN BASL,H FOR GIVEN LON C=517 ARG=000ABCDE, GENERATE BASH=000001CD AND BASL=EABAB000 BASL=EABA000 BASL=EABA00 BASL=EABA000 BASL=EABA000 BAS</pre>
FC0A:F0 5A FC66 142 BEG LF ; IF 5D DD IT. FC0C:C9 8B 143 CMP #\$8B ;BACK SPACE? (CONTRDL-H) FC0E:D0 C9 FBD9 144 BNE BELL1 ; NO, CHECK FOR BELL.	FBC1: FBC1:48 FBC2:4A FBC2:40 FBC3:27 03 FBC5:07 04 FBC7:65 27 FBC7:68 FBC7:07 02 FBC7:08 27 FBC7:08 27 FBC7:08 27 FBD2:04 FBD2:04 FBD2:04 FBD2:05 28 FBD2:05 28 FBD2:04 FBD2:07 97 FBD8:00 12 FBD2:07 97 FBD8:00 12 FBD2:07 97 FBD8:00 12 FBD2:04 79 FBD2:04 79 FBD2:20 48 FB22:40 20 FB22:40 20	FBEF FC CO FBE4 FC62 FBF0 FBF0	101 102 103 104 105 107 110 107 110 111 112 113 114 115 117 120 122 123 124 127 128 127 128 127 130 132 133 134 135	BASCALC BASCLC2 BELL1 BELL2 RTS2B STORADV ADVANCE RTS3 VIDOUT	FIN DESCRIPTION OF A DE	A # 4 # 5 # 504 # 504 # 504 # 504 # 507 # 502 # 507 # 502 #	<pre>CALCBASE ADDR IN BASL,H ; FOR GIVEN LINE ND. ; ARG=000ABCDE, GENERATE ; BASH=000001CD ; AND ; BASL=EABAB000 ; BASL=EABAB000 ; BASL=EABAB000 ; BASL=EABAB000 ; DASL=EABAB000 ; URSOR H INDEX TO Y-REG ; TOGOLE SPEAKER AT 1 KHZ ; FOR .1 SEC. ; CURSOR H INDEX TO Y-REG ; STORE CHAR IN LINE ; INCREMENT CURSOR H INDEX ; (MOVE RIGHT) ; BEYOND WIDTH? ; YES, CR TO NEXT LINE. ; ND, RCTURN. ; CONTROL CHAR? ; ND, OUTPUT IT. ; INVERSE VIDEO? ; YES</pre>
FCOE: DO C9 FBD9 144 BNE BELL1 > ND, CHECK FOR BELL.	FBC1: FBC1: FBC1: FBC2:4A FBC3:27 FBC3:27 FBC3:27 FBC3:27 FBC3:27 FBC3:27 FBC3:27 FBC3:27 FBC3:27 FBC3:28 FBC3:20 FBD2:05 FBD3:04 FBD4:05 FBD5:20 FBD5:20 FBD5:20 FBD5:20 FBD5:47 FBD5:20 FBD5:20 FBD5:20 FBD5:20 FBD5:20 FB2:20	FBEF FC CO FBE4 FC62 FBF0 FBF0 FC62	$\begin{array}{c} 101\\ 102\\ 103\\ 104\\ 105\\ 106\\ 107\\ 110\\ 107\\ 110\\ 111\\ 112\\ 113\\ 114\\ 115\\ 117\\ 122\\ 123\\ 124\\ 125\\ 127\\ 132\\ 124\\ 125\\ 127\\ 130\\ 131\\ 134\\ 135\\ 137\\ 134\\ 137\\ 134\\ 137\\ 140\\ 137\\ 141\\ 141\\ 141\\ 141\\ 141\\ 141\\ 141\\ 14$	BASCALC BASCLC2 BELL1 BELL2 RTS2B STORADV ADVANCE RTS3 VIDOUT	FIN DESCRIPTION OF A DE	A #\$03 #\$04 BASH #\$04 BASL BASL A A A BASL BASL BASL BASL BAS	<pre>;CALCBASE ADDR IN BASL,H ;FOR GIVEN LINE ND. ;OC=LINE ND.<=%17 ;ARG=0000ABCDE, GENERATE ;BASH=000001CD ;ARD ;BASL=EABAB000 ;BASL=EABAB000 ;BASL=EABAB000 ;BASL=EABAB000 ;DELAY.01 SECONDS ;TOGGLE SPEAKER AT 1 KHZ ;FOR.1 SEC. ;CURSDR H INDEX TO Y-RE0 ;STORE CHAR IN LINE ;INCERFENT CURSDR H INDEX ;(MOVE RIGHT) ;PES, CR TO NEXT LINE. ;ND, RETURN. ;CONTROL CHAR? ;ND, OUTPUT IT. ;INVERSE VIDED? ;YES. ;LINE FEED?</pre>
FC10:C6 24 145 BS DEC CH ; DECREMENT CURSOR H INDEX	FBC1: FBC1: FBC1: FBC2:40 FBC3:27 FBD3:05 FBD3:00 FBD3:01 FBC3:20 FBE3:20 FBE4:40 FB5:21 FB5:22 FB5:23<	FBEF FC FC FBE4 FC62 FBF0 FBF0 FC62 FC64	101 102 103 104 105 106 107 108 109 110 110 111 113 114 113 114 113 114 113 114 122 123 124 127 128 127 128 127 128 127 128 127 128 127 128 127 128 127 128 127 128 127 128 127 128 127 128 129 127 128 129 129 129 129 129 129 129 129 129 129	BASCALC BASCLC2 BELL1 BELL2 RTS2B STORADV ADVANCE RTS3 VIDDUT	FIN DESCRIPTION OF A DE	A # 0 # 0 # 50 # 50	<pre>CALCBASE ADDR IN BASL,H FOR GIVEN LINE ND. GALLBASE ADDR IN BASL,H FOR GIVEN LINE ND. ARG=000ABCDE, GENERATE BASH=000001CD AND BASL=EABAB000 BASL BASL=EABAB000 BASL=EABA000 BASL=EABA000 BASL=EABA000 BASL=EABA000 BASL=EABA</pre>
	FBC1: FBC1: FBC1: FBC2:40 FBC3:27 FBC3:27 FBC3:27 FBC3:27 FBC3:27 FBC3:27 FBC3:27 FBC3:27 FBC3:27 FBC3:28 FBC3:20 FBD2:04 FBD3:04 FBD4:65 FBD5:20 FBD5:20 FBD5:20 FBD5:20 FBD5:20 FBD7:20 FBD7:20 FBD7:20 FBD7:20 FB24:47 FB25:40 FB25:40 FB25:40 FB25:40 FB25:40 FB25:40 FB25:40 FB25:40 FB25:41 FB45:42 FB57:42 FB57:42 FB57:42 FB57:42 FB57:42 FB57:50 FB57:50 FB57:50 FC01:48 FC02:10 FC04:	FBEF FC C0 FBE4 FC62 FBF0 FBF0 FC62 FC66	$\begin{array}{c} 101\\ 101\\ 102\\ 103\\ 104\\ 105\\ 106\\ 107\\ 108\\ 109\\ 110\\ 111\\ 112\\ 113\\ 114\\ 112\\ 113\\ 114\\ 125\\ 124\\ 125\\ 124\\ 125\\ 124\\ 125\\ 123\\ 134\\ 135\\ 137\\ 140\\ 137\\ 144\\ 143\\ \end{array}$	BASCALC BASCLC2 BELL1 BELL2 RTS2B STORADV ADVANCE RTS3 VIDOUT	FIN DESCRIPTION OF A DE	A # 4 # 404 BASH # 518 BASLC2 # 577 BASL A A BASL	<pre>;CALCBASE ADDR IN BASL,H ;FOR GIVEN LINE ND. ;O<=LINE ND.<=%17 ;ARG=0000ABCDE, GENERATE ;BASH=000001CD ;ARD ;BASL=EABAB000 ;BASL=EABAB000 ;BASL=EABAB000 ;BASL=EABAB000 ;DELAY.01 SECONDS ;JOELAY.01 SECONDS ;JOELAY.01 SECONDS ;JOEQLE SPEAKER AT 1 KHZ ;FOR.1 SEC. ;CURSDR H INDEX TO Y-RE0 ;STORE CHAR IN LINE ;INCEMENT CURSDR H INDEX ;(MOVE RIGHT) ;PES, GR TO NEXT LINE.;NO, BETVOND WIDTH? ;YES, CR TO NEXT LINE.;NO, BETVOND WIDTH? ;YES, GR TO NEXT LINE.;NO, BETVOND WIDTH? ;YES, GR TO NEXT LINE.;NO, BUTPUT IT.; ;INCERSENT CUREOR IT. ;SCR? ;LINE FEED? ;IF SD. DO IT. ;BACK SFACE? (CONTROL-H)</pre>
	FBC1: FBC1: FBC1:48 FBC2:40 FBC3:2703 FBD3:052 FBD3:004 FB50:004	FBEF FC FC FBE4 FC62 FBF0 FBF0 FC62 FC66 FBD9	101 102 103 104 105 106 107 108 109 111 112 113 114 115 116 117 113 114 115 116 117 113 114 122 123 124 125 124 133 134 135 134 137 138 137 139 140 141	BASCALC BASCLC2 BELL1 BELL2 RTS2B STORADV ADVANCE RTS3 VIDOUT	FIN DESCRIPTION OF A DE	A # 4 # 5 # 5 # 5 # 5 # 5 # 5 # 5 # 5	<pre>CALCBASE ADDR IN BASL,H FOR GIVEN LINE NO. Collection Collection (CALCBASE ADDR IN BASL,H FOR GIVEN LONCEST (CALCBASE ADDR) Collection (CALCBASE ADDR) (CALCBASE ADDR) (C</pre>

FC12: 10 E8 FBFC					
	146			RTS3	IF POSITIVE, OK; ELSE MOVE UP.
FC14: A5 21	147			WNDWDTH	SET CH TO WINDOW WIDTH - 1.
FC16:85 24	148	S	TA	СН	
FC18: C6 24	149			сн	(RIGHTMOST SCREEN POS)
FC1A: A5 22	150			WNDTOP	CURSOR V INDEX
FC1C: C5 25	151			CV	
FC1E: BO OB FC2B	152			RTS4	IF TOP LINE THEN RETURN
FC20: C6 25	153			cv	DECR CURSOR V INDEX
FC22: A5 25				cv	GET CURSOR V INDEX
FC24: 20 C1 FB				BASCALC	GENERATE BASE ADDRESS
FC27:65 20	156			WNDLFT	ADD WINDOW LEFT OFFSET
FC29:85 28	157			BASL	; TO BASL
FC2B: 60	158		TS		;ESC '@'?
FC2C: 49 CO			OR	#\$C0	
FC2E: F0 28 FC58	160			HOME #SED	; IF SO DO HOME AND CLEAR ; ESC-A OR B CHECK
FC30: 69 FD FC32: 90 CO FBF4	161		DC	#\$FD ADVANCE	A ADVANCE
FC32: 90 C0 FBF4	163			BS BS	B BACKSPACE
FC34:F0 DA FC10 FC36:69 FD	164			#\$FD	ESC-C OR D CHECK
FC36: 69 FD FC38: 90 2C FC66	165			LF	C, DOWN
FC3A: FO DE FC1A	166			UP	D, GO UP
FC3C: 69 FD	167			#\$FD	ESC-E OR F CKECK
FC3E: 90 5C FC9C	168				E CLEAR TO END OF LINE
FC40: D0 E9 FC2B	169			RTS4	; E, CLEAR TO END OF LINE ; ELSE NOT F, RETURN
FC42: 0001	170	D		APPLEZE	; /RRA0981
FC42; FC42 FC42: A0 00			QU	*	; /RRA0981
FC42 A0 00	172			#O	CODE=CLREDP/RRA0981
FC44: FO 2C FC72	173		EQ	XGOTOCX	DO 40/80 /RRA0981
FC46: A8 C3 A7 A0	174		SC	'(C) 1	;D0 40/80 /RRA0981 1981-82, APPLE'
FC58:	175		LSE		; /RRA0981
5	176		DY	сн	ESC F IS CLR TO END OF PAGE
s	177			cv	
5			HA		; SAVE CURRENT LINE ND. ON STACK ; CALC BASE ADDRESS
5	179	J	SR	VTABZ	CALC BASE ADDRESS
5	180	JS	SR	CLEOLZ	CLEAR TO EOL. (SETS CARRY)
S	181	LI	DY	#\$00	CLEAR FROM H INDEX=0 FOR REST
S	182		LA		INCREMENT CURRENT LINE NO.
S	183		DC	#\$00	; (CARRY IS STILL SET)
S	184		MP	WNDBTM	DONE TO BOTTOM OF WINDOW?
S	185		CC	CLEDP1	; ND, KEEP CLEARING LINES.
5	186		CS	VTAB	; YES, VTAB TO CURRENT LINE.
FC58:	187		IN		/ /RRA0981
FC58: 0001	188	D		APPLE2E	; /RRA0981
FC58: FC58			QU	*	; /RRA0981
FC58: A0 01	190		DY	#1	CODE=HOME/RRA0981
FC5A: DO 16 FC72	191		NE	XGOTOCX	100 40/80 /RRA0981
FC5C:D2 C9 C3 CB	192		SC	RICK A	A' ; OUR HERD
FC62:	193		LSE		/RRA0981
				WNDTOP	; INIT CURSOR V
S				C11	AND IL INDICES
S	195			CV	; AND H INDICES
5 5	195 196	ū	DY	#\$00	; AND H INDICES
s S S	195 196 197	LI S'	DY TY	#\$00 CH	; AND H INDICES ; THEN CLEAR TO END OF PAGE.
ទ ទ ទ ទ	195 196 197 198	LI S' BI	DY TY EQ	#\$00	; AND H INDICES ; THEN CLEAR TO END OF PAGE. ; (ALWAYS TAKEN)
5 5 5 FC62:	195 196 197 198 199	Li S' Bi F	DY TY EQ IN	#\$00 CH CLEOP1	; AND H INDICES ;THEN CLEAR TO END OF PAGE. ;(ALWAYS TAKEN) ;(RRA0981
5 5 5 FC62: FC62: A7 00	195 196 197 198 199 200	LI S' Bi CR LI	DY EQ IN DA	#\$00 CH CLEOP1 #\$00	; AND H INDICES ;THEN CLEAR TO END OF PAGE. ;(ALWAYS TAKEN) ;/RRA0981 ;CURSOR TO LEFT OF INDEX
S S S FC62: FC62: A9 00 FC64: 85 24	195 196 197 198 199 200 201	LI S' Bi F CR LI S'	DY EQ IN DA	#\$00 CH CLEOP1 #\$00 CH	; AND H INDICES ;THEN CLEAR TO END OF PAGE. ;(ALWAYS TAKEN) ;(RATSOR TO LEFT OF INDEX ;(RET CURSOR H=O)
5 5 5 FC62: FC62: A9 00 FC64: B5 24 FC66: E6 25	195 196 197 198 199 200 201 202	LI S Bi F CR LI S LF II	DY EQ IN DA TA	#\$00 CH CLEOP1 #\$00 CH CV	; AND H INDICES ;THEN CLEAR TO END OF PAGE. ;(ALWAYS TAKEN) ;/RRA0981 ;CURSOR TO LEFT OF INDEX
S S S FC62: FC62: A7 00 FC64: B5 24 FC66: E6 25 FC68: A5 25	195 196 197 198 199 200 201	LI S Bi CR LI CR LI LF II LF LI	DY EQ IN DA	#\$00 CH CLEOP1 #\$00 CH CV CV	; AND H INDICES ;THEN CLEAR TO END OF PAGE. ;(ALWAYS TAKEN) ;/RRA0781 ;CURSOR TO LEFT OF INDEX ;(RET CURSOR H=0) ;INCR CURSOR V. (DDWN 3 LINE)
S S S FC62: A9 00 FC64: 85 24 FC66: E6 25 FC66: A5 25 FC66: C5 23	195 196 197 198 199 200 201 202 203 203	LI S S CR LI S LF II LI CI	DY EQ IN DA TA NC DA	#\$00 CH CLEOP1 #\$00 CH CV CV CV WNDBTM	; AND H INDICES ; THEN CLEAR TO END OF PAGE. ; (ALMAYS TAKEN) ; (RAR0498 LEFT OF INDEX ; CURSOR TO LEFT OF INDEX ; (RET CURSOR H=O) ; INCR CURSOR V. (DOWN 1 LINE) ; OFF SCREEN?
5 5 5 FC62: FC62: A7 00 FC64: 85 24 FC66: 85 25 FC68: A5 25 FC68: A5 25 FC66: C5 23 FC66: C5 23 FC66: C7 08 4 FC62	195 196 197 198 199 200 201 202 203 204 205	LI S B B F CR LI S LF II LF II C S B	DY EQ IN DA NC DA MP ICC	#\$00 CH CLEOP1 #\$00 CH CV WNDBTM VTABZ	; AND H INDICES ; THEN CLEAR TO END OF PAGE. ; (ALMAYS TAKEN) ; /RRA0781 ; CURSOR TO LEFT OF INDEX ; (RET CURSOR H=O) ; INCR CURSOR V. (DDWN 1 LINE) ; OFF SCREEN? NO, SET BASE ADDR
S S S FC62: FC62: A7 00 FC64: 85 24 FC66: 85 25 FC68: 65 23 FC66: 70 86 FC24 FC62: 62 25	195 196 197 198 200 201 202 203 204 205 206	LI BI CR LF LF D D D	DY EQ IN DA TA NC DA	#\$00 CH CLEOP1 #\$00 CH CV CV CV WNDBTM	; AND H INDICES ; THEN CLEAR TO END OF PAGE. ; (ALMAYS TAKEN) ; (RAR0498 LEFT OF INDEX ; CURSOR TO LEFT OF INDEX ; (RET CURSOR H=O) ; INCR CURSOR V. (DOWN 1 LINE) ; OFF SCREEN?
5 5 5 7662: 7664:85 24 7664:85 24 7666:85 25 7668:85 25 7668:85 25 7666:0 84 7666:66 25 7666:66 25 7670: 0001	195 196 197 198 199 200 201 202 203 204 205 206 207	LI S Bi CR LI S LF II LF I C B D D D	DY EQ IN DA STA NC DA MP CC DEC	#\$00 CH CLEOP1 #\$00 CV CV CV WNDBTM VTABZ CV	; AND H INDICES ; THEN CLEAR TO END OF PAGE. ; (ALWAYS TAKEN) ; (RAGO981 ; (REA CLEFT OF INDEX ; (RET CURSOR H=0) ; INCR CURSOR V. (DOWN 1 LINE) ; OFF SCREEN? , NO, SET BASE ADDR ; DECR CURSOR V. (BACK TO BOTTOM) ; /RRA0981
S S S FC62: FC62: A9 00 FC64: 85 24 FC66: 85 25 FC68: 85 25 FC68: 85 25 FC64: C5 23 FC66: C6 25 FC70: 0001 FC70: FC70	195 196 197 198 199 200 201 202 203 204 205 206 207	L S Bi F CR LI LF II CI CI C DI D SCROLL E	DY EQ IN DA TA NC DA MP ICC DEC	#\$00 CH CLEOP1 #\$00 CV CV CV WNDBTM VTABZ CV	; AND H INDICES ; THEN CLEAR TO END OF PAGE. ; (ALWAYS TAKEN) ; (RALWAYS TAKEN) ; (RACT CURSOR THEO) ; (RET CURSOR THEO) ; INCR CURSOR V. (DOWN 1 LINE)) OFF SCREEN? ; NO, SET BASE ADDR ; DECR CURSOR V. (BACK TO BOTTOM) ; /RRAO9B1
5 5 5 7662: 7664:85 24 7664:85 24 7666:85 25 7668:85 25 7668:85 25 7666:0 84 7666:66 25 7666:66 25 7670: 0001	195 196 197 198 199 200 201 202 203 204 205 204 205 206 207 208 209	CR LI CR LI LF II SCROLL LI	DY TY EQ IN DA NC DA MP CC DEC DEC DU	#\$00 CH CLEOP1 #\$00 CH CV CV V WNDBTM VTABZ CV APPLE2E *	; AND H INDICES ; THEN CLEAR TO END OF PAGE. ; (ALWAYS TAKEN) ; (RAGO981 ; (REA CLEFT OF INDEX ; (RET CURSOR H=0) ; INCR CURSOR V. (DOWN 1 LINE) ; OFF SCREEN? , NO, SET BASE ADDR ; DECR CURSOR V. (BACK TO BOTTOM) ; /RRA0981
5 5 5 7662: 7662:49 00 7664:85 24 7668:45 25 7668:45 25 7668:45 25 7668:45 25 7666:0 84 7666:62 5 7670: 0001 7670: 7670	195 196 197 290 201 202 203 204 205 206 207 208 209 210	CR LI CR LI LF II SCROLL LI	DY TY EQ IN DA NC DA MP CC DE C DE DE DE C DE C DE C DE C DE C DE C DE C DE C DE C DE C DE C DE	#\$00 CH CLEOP1 #\$00 CH CV CV WNDBTM VTABZ CV APPLE2E * #2	; AND H INDICES ; THEN CLEAR TO END OF PAGE. ; (ALWAYS TAKEN) ; (RAGO981 ; (URSOR TO LEFT OF INDEX ; (URSOR H=0) ; INCR CURSOR H=0) ; INCR CURSOR V. (DOWN 1 LINE) ; OFF SCREEN? NO. SET BASE ADDR ; (RRAO981 ; /RRAO981 ; (CODE=SCRULL/RRAO981
S S S FC62: FC62: A9 00 FC64: 85 24 FC66: 85 25 FC66: 85 25 FC66: 70 86 FC24 FC670: 6001 FC70: FC70 FC70: A0 02 FC72: 4C 84 FB	195 196 197 198 199 200 201 202 203 204 205 206 207 208 207 208 209 210 211	CR LI SCR LI LF LI SCROLL LI SCROLL LI XGOTOCX J	BY EQ IN DA STA NC DA MP CC CC CC CC CC CC CC CC CC CC CC CC CC	+\$00 CH CLEOP1 #\$00 CH CV CV WNDBTM VTABZ CV APPLE2E * #2 g0TOCX	; AND H INDICES ; THEN CLEAR TO END OF PAGE. ; (ALWAYS TAKEN) ; (RAGO981 ; (URSOR TO LEFT OF INDEX ; (URSOR H=0) ; INCR CURSOR H=0) ; INCR CURSOR V. (DOWN 1 LINE) ; OFF SCREEN? NO. SET BASE ADDR ; (RRAO981 ; /RRAO981 ; (CODE=SCRULL/RRAO981
S S S FC62: A9 00 FC64: B5 24 FC66: E6 25 FC66: A5 25 FC66: A5 25 FC66: C5 23 FC66: 90 B6 FC70: 0001 FC70: FC70 FC70: A0 02 FC72: 4C B4 FB FC75:	195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212	CR LI SCR LI LF II SCROLL E XGOTOCX J	BY EQ IN DA STA NC DA MP CC CC CC CC CC CC CC CC CC CC CC CC CC	+\$00 CH CLEOP1 #\$00 CH CV CV WNDBTM VTABZ CV APPLE2E * #2 g0TOCX	; AND H INDICES ;THEN CLEAR TO END OF PAGE. ;(ALWAYS TAKEN) ;/RRAO981 ;URSOR TO LEFT OF INDEX ;(RET CURSOR H=0) ;INCR CURSOR V. (DOWN 1 LINE) ;OFF SCREEN? ;NO. SET BASE ADDR ;DECR CURSOR V. (BACK TO BOTTOM) ;/RRAO981 ;/RRAO981 ;DD 40/B0 /RRAO981
S S S FC62: FC62: A7 00 FC64: 85 24 FC68: A5 25 FC68: A5 25 FC66: C6 25 FC70: P67 FC70: A02 FC70: A02 FC72: 4C B4 FB FC75: FC75:	195 196 197 198 200 201 202 203 204 205 204 205 204 205 206 207 208 209 210 211 212 213	CR LF II BB LF II SCROLL LC SCROLL EI XGOTOCX JJ * * IRQ SNIFFER * RDBOSTORE EI	DY TY EQ IN DA STA NC DA MP ECC DO CO CO CO CO CO CO CO CO CO CO CO CO CO	+\$00 CH CLEOP1 #\$00 CH CV CV WNDBTM VTABZ CV APPLE2E * #2 g0TOCX	; AND H INDICES ; THEN CLEAR TO END OF PAGE. ; (ALMAYS TAKEN) ; (RTAG981 ; CURSOR TO LEFT OF INDEX ; (RTC CURSOR H=O) ; INCR CURSOR V. (DOWN 1 LINE) ; OFF SCREEN? NO. SET BASE ADDR ; DECR CURSOR V. (BACK TO BOTTOM) ; /RA0981 ; CODE=SCROLL/RRA0981 ; D0 4/B0 /RRA0981 ; /RRA0981
5 5 5 5 FC62: A9 00 FC64: B5 24 FC66: E6 25 FC66: A5 25 FC66: A5 25 FC66: C5 23 FC66: C6 25 FC70: 0001 FC70: FC70 FC70: A0 02 FC72: 4C B4 FB FC75: FC75:	195 196 197 198 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214	CR LI CR LI LF II LF II SCROLL EI XGOTOCX J * IRQ SNIFFER * IRQ SNIFFER RDBOSTORE EI RDPAGE2 EI	DY STY EQ IN DA STA NC DA C DA STA STA NC DA STA STA STA STA STA STA STA STA STA ST	#\$00 CH CLEOP1 #\$00 CH CV CV VTABZ CV APPLE2E # # gOTOCX VIDEO CODE:	; AND H INDICES ;THEN CLEAR TO END OF PAGE. ;(ALWAYS TAKEN) ;/RRAO981 ;URSOR TO LEFT OF INDEX ;(RET CURSOR H=O) ;INCR CURSOR V. (DOWN 1 LINE) ;OFF SCREEN? ;NO, SET BASE ADDR ;DECR CURSOR V. (BACK TO BOTTOM) ;/RRAO981 ;JOE AO/BO //RRAO981 ;/RRAO981 ;/RRAO981 ;/RRAO981
S S S FC62: FC62: FC64:85 24 FC66:85 25 FC66:85 25 FC66:70 86 FC70: F	195 196 197 198 199 200 201 202 203 204 205 204 205 206 207 208 207 208 207 211 212 213 214 215 215	CR LI ST CR LI LF II SCROLL E SCROLL LI XGOTOCX JI * IRQ SNIFFER * RDBOSTORE E RDPAGE2 E	DY TY EQ IN DA TA NC DA CC DC CQU DY MP FOR CQU PHA	#\$00 CH CLEOP1 #\$00 CH CV CV CV VTABZ CV APPLE2E # 2 GOTOCX VIDEO CODE: \$C018 \$C010	; AND H INDICES ; THEN CLEAR TO END OF PAGE. ; (ALMAYS TAKEN) ; (RAG0981 ; CURSOR TO LEFT OF INDEX ; (RET CURSOR H=O) ; INCR CURSOR V. (DDWN 1 LINE) ; OFF SCREEN? NO, SET BASE ADDR ; DECR CURSOR V. (BACK TO BOTTOM) ; /RA0981 ; CODE=SCROLL/RRA0981 ; /RRA0981 ; /RRA0981 ; /RRA0981 ; /RRA0981 ; /RRA0981 ; /RRA0981 ; /RRA0981 ; /RRA0981
S S S FC62: FC62: A9 00 FC64: 85 24 FC66: 85 25 FC66: 85 25 FC66: 85 25 FC66: 62 5 FC70: 0001 FC70: 0001 FC70: C018 FC770: C018 FC75: FC75: FC75: C018 FC75: C	195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 214 215 214	CR LI SCR LI LF II SCROLL EI SCROLL EI XGOTOCX JI * IRQ SNIFFER * IRQ SNIFFER RDBOSTORE EI RDBOSTORE EI RDPAGE2 EI LL	DY TY EQ IN IDA TA NC DA DA DC CC DC CC DC CC DC CC DC CC DC CC DC CC DC CC DC CC C	#\$00 CH CLEOP1 #\$00 CH CV CV VTABZ CV APPLE2E #2 GOTOCX VIDE0 CODE: \$C018	; AND H INDICES ; THEN CLEAR TO END OF PAGE. ; (ALWAYS TAKEN) ; (RACO981 ; UCWSOR TO LEFT OF INDEX ; (RET CURSOR H=0) ; INCR CURSOR V. (DOWN 1 LINE) ; OFF SCREEN? ; NO, SET BASE ADDR ; DECR CURSOR V. (BACK TO BOTTOM) ; /RRAO981 ; PRESERVE AC/RRAO981 ; PLAG=>N (RRAO981
S S S FC62: FC62: FC64: 85 24 FC66: 85 25 FC66: 85 25 FC66: 85 25 FC66: 70 84 FC70:	195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 218	CR LI SS LF LI SCROLL E XGOTOCX J * * ROBOSTORE E RDPAGE2 E LL A	DY TY EQ IN DA TA TA TA TA TA TA TA TA TA T	#\$00 CH CLEOP1 #\$00 CH CV CV CV VTABZ CV APPLE2E # 2 GOTOCX VIDEO CODE: \$C018 \$C010	; AND H INDICES ; THEN CLEAR TO END OF PAGE. ; (ALMAYS TAKEN) ; (RARAO981 ; CURSOR TO LEFT OF INDEX ; (RET CURSOR H=0) ; INCR CURSOR V. (DDWN 1 LINE) ; OFF SCREEN? NO, SET BASE ADDR ; DECR CURSOR V. (BACK TO BOTTOM) ; /RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; FLAG=>N /RRAO981 ; FLAG=>N /RRAO981 ; FLAG=>N /RRAO981
S S S FC62: FC62: A9 00 FC64: 85 24 FC66: 85 25 FC66: 85 25 FC66: 65 25 FC66: C6 25 FC70: 0001 FC70: 0001 FC70: C018 FC70: FC75 FC75: C018 FC75: C018 FC76: C018 FC765 FC76 FC76 FC76 FC76 FC76 FC76 FC76 FC76 FC76 FC76 F	195 196 197 198 199 200 201 202 203 204 205 206 207 210 212 213 214 215 216 217 218 219	CR LI SCR LI LF II LF II SCROLL EI XGOTOCX JI * IRQ SNIFFER * RDBOSTORE EI RDPAGE2 EI RDPAGE2 FI LI A' PI	DY TTY EQ IN DA STA NC DA STA STA STA STA STA STA STA STA STA ST	#\$00 CH CLEOP1 #\$00 CV CV VTABZ CV APPLEZE #2 GOTOCX VIDEO CODE: \$C016 #C01C RD80STORE A	; AND H INDICES ; THEN CLEAR TO END OF PAGE. ; (ALWAYS TAKEN) ; (RAA0981 ; (REA0981 ; UCRSOR TO LEFT OF INDEX ; (RET CURSOR H=0) ; INCR CURSOR V. (DOWN 1 LINE) ; OFF SCREEN7 ; NO, SET BASE ADDR ; DECR CURSOR V. (DACK TO BOTTOM) ; /RRA0981 ; /RRA0981 ; /RRA0981 ; PRESERVE AC/RRA0981 ; FLAG=>N /RRA0981 ; FLAG=>N /RRA0981 ; FLAG=>C /RRA0981 ; FLSTOR EA/CRA0981 ; FLSTOR EA/CRA0981
S S S FC62: FC62: FC64: A9 00 FC64: 85 24 FC66: A5 25 FC66: A5 25 FC66: A5 25 FC66: A5 25 FC70: FC70: FC70: FC70: FC70: FC70: FC70: FC77: FC75:	195 197 198 197 200 201 202 203 204 205 204 205 204 205 204 205 204 205 204 207 208 207 208 211 212 213 214 215 214 215 214 215 214 217 218	CR LI SG LF LI SCROLL E SCROLL LI SCROLL LI XGOTOCX JI * RDBOSTORE E RDPAGE2 E RDPAGE2 E RDPAGE2 E C RDPAGE2 E RDPAGE2 E RDPAG	DY EQ IN DA DA DA DA CC DO CQU CQU CQU CQU CQU CQU CQU CQU CQU CQU	#\$00 CH CLEOP1 #\$00 CH CV CV CV VTABZ CV APPLE2E # 2 GOTOCX VIDEO CODE: \$C018 \$C010	; AND H INDICES ; THEN CLEAR TO END OF PAGE. ; (ALWAYS TAKEN) ; (RARO981 ; (RET CURSOR H=0) ; INCR CURSOR TO LEFT OF INDEX ; (RET CURSOR H=0) ; INCR CURSOR V. (DOWN 1 LINE) ; OFF SCREEN? NO, SET BASE ADDR ; DECR CURSOR V. (BACK TO BOTTOM) ; /RRAO981 ; /RESERVE AC/RRAO981 ; FLAG->N /RRAO981 ; RESGURE AC/RRAO981 ; REACH. (RRAO981 ; REACH. (RRAO981) ; REACH. (RRAO981)
S S S FC62: FC62: A7 00 FC64: 85 24 FC66: A5 25 FC66: A5 25 FC66: C6 25 FC70: 0001 FC70: 6001 FC70: 0001 FC70: C016 FC70: C016 FC70: C016 FC75: C018 FC75: C018 FC76: C01	195 196 197 198 197 200 201 203 204 205 204 205 204 205 206 207 208 210 211 212 213 214 215 216 215 216 217 218 219 221	CR LI SCR LI LF II LF II SCROLL E XGOTOCX JJ * * IRQ SNIFFER * IRQ SNIFFER * RD80STORE E RDPAGE2 EI RDPAGE2 EI RDPAGE2 PI LI A *	DY TTY TA TA TA TA TA TA TA TA TA TA TA TA TA	#\$00 CH CLEDP1 #\$00 CH CV CV VTABZ CV APPLE2E #2 GOTOCX VIDEO CODE: \$C010 \$C010 RDB0STORE A RDPAGE2	; AND H INDICES ; THEN CLEAR TO END OF PAGE. ; (ALMAYS TAKEN) ; (RAG098] ; CURSOR TO LEFT OF INDEX ; (RET CURSOR H=O) ; INCR CURSOR V. (DOWN 1 LINE) ; OFF SCREEN? NO. SEE BASE ADDR ; DCCR CURSOR V. (BACK TO BOTTOM) ; /RRA0981 ; CDDE=SCROLL/RRA0981 ; DA (FBO /RRA0981 ; FLAG->N /RRA0981 ; FLAG
S S S FC62: FC62: FC64: FC68: FC68: FC68: FC70:	193 194 197 197 200 201 203 203 204 205 203 204 205 203 204 207 208 212 212 213 214 212 212 213 214 214 217 218 214 217 218 214 217 218 219 220 222	CR LI SCR LI LF II SCROLL EI XGOTOCX JI * TRQ SNIFFER * TRQ SNIFFER * RDPAGE2 EI RDPAGE2 EI A RDPAGE2 EI A RDPAGE3 EI B B B B B B B B B B B B B B B	DY TTY TTY ITY ITY ITY ITY ITY ITY ITY IT	#\$00 CH CLEOP1 #\$00 CH CV CV VTABZ CV APPLE2E * #2 GOTOCX VIDED CODE: * CO18 *C018 CO18 * CO18 * CO18 * CO18 * CO18 * CO18 * CO16 CO17 CO17 CO17 CO17 CO17 CO17 CO17 CO17	; AND H INDICES ; THEN CLEAR TO END OF PAGE. ; (ALWAYS TAKEN) ; (RAD0981 ; (RET CURSOR TO LEFT OF INDEX ; (RET CURSOR H=0) ; INCR CURSOR V. (DOWN 1 LINE) ; OFF SCREEN? NO, SET BASE ADDR ; DECR CURSOR V. (BACK TO BOTTOM) ; /RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; FAGESRVE AC/RRAO981 ; FLAG->N /RRAO981 ; FLAG->N /RRAO981 ; FLAG->N /RRAO981 ; FLAG-YM /RRAO981 ; /RRAO981 ; /RRAO981 ; FLAG->N /RRAO981 ; FLAG->N /RRAO981 ; FLAG->N /RRAO981 ; /RRAO981 ; NCT BANKSWITCHING/RRAO981
S S S FC62: FC62: A7 00 FC64: 85 24 FC66: 62 25 FC66: 63 25 FC66: 70 86 FC24 FC66: C6 25 FC70: 0001 FC70: 0001 FC70: C001 FC70: C001 FC70: A0 22 FC75: FC	199 197 198 200 201 202 203 204 205 208 208 208 208 208 208 210 211 212 213 214 215 214 217 218 214 217 218 214 217 218 222 222	CR LI SCR LI LF II LF II SCROLL E XGOTOCX JJ * * IRQ SNIFFER * IRQ SNIFFER RDB0STORE E RDPAGE2 EI RDPAGE2 EI RDPAGE3 FI B SS	DY TTY TTY DA DA DA DA DA DA DA DA DA DA DA DA DA	#\$00 CH CLEDP1 #\$00 CH CV CV VTABZ CV APPLE2E #2 GOTOCX VIDEO CODE: \$C010 \$C010 RDB0STORE A RDPAGE2 RDCX \$C054	; AND H INDICES ; THEN CLEAR TO END OF PAGE. ; (ALMAYS TAKEN) ; (RAG0781 ; CURSOR TO LEFT OF INDEX ; (RET CURSOR H=O) ; INCR CURSOR V. (DOWN 1 LINE) ; OFF SCREEN? NO. SEE BASE ADDR ; DECR CURSOR V. (BACK TO BOTTOM) ; /RRA0781 ; CDDE=SCROLL/RRA0781 ; DOL/SOC /RRA0781 ; FLAG-SN /RRA0781 ; FLAG-SC /RRA0781 ; FLAG
S S S FC62: FC62: FC62: FC64: 85 24 FC66: 85 25 FC66: 85 25 FC66: 85 25 FC66: 9 84 FC70: FC70: FC70: FC70: FC70: FC70: FC77: FC75:	199 197 198 200 201 202 203 204 205 205 205 205 206 210 211 212 213 213 214 213 214 214 214 214 214 214 214 214 214 214	CR LI SCROLL EL XGOTOCX J * TRQ SNIFFER * TRQ SNIFFER * RDPAGE2 EL RDPAGE2 EL B RDCX B	DY ITY ITY ITA DA DA DA CC CO CO CO CO CO CO CO CO CO CO CO CO	#\$00 CH CLEOP1 #\$00 CH CV CV VTABZ CV APPLE2E # # GOTOCX VIDED CODE: * CO18 *C018 *C018 CO18 *C018 RDBOSTORE A RDPAGE2 RDCX *CO54 *CO54 *CO54 RDCXRM	; AND H INDICES ; THEN CLEAR TO END OF PAGE. ; (ALWAYS TAKEN) ; (RAD0981 ; (RET CURSOR TO LEFT OF INDEX ; (RET CURSOR H=0) ; INCR CURSOR V. (DOWN 1 LINE) ; OFF SCREEN? ; NO, SET BASE ADDR ; DECR CURSOR V. (BACK TO BOTTOM) ; /RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; FAGESRVE AC/RRAO981 ; FLAG->N (RRAO981 ; FLAG->N (RRAO981 ; /RRAO981 ; RESETURE AC/RRAO981 ; /RRAO981 ; NOT BANKSWITCHING/RRAO981 ; FLAG->N (RRAO981 ; FLAG->N (RRAO981 ; /RRAO981 ; /RRAO981 ; RESCORE AC/RRAO981 ; /RRAO981 ; /RRAO981 ; RESCORE AC/RRAO981 ; /RRAO981 ; FLAG->N (RRAO981 ; FLAG->N (RRAO981 ; FLAG->N (RRAO981 ; FLAG->N (RRAO981 ; FLAG-N (RRAO981 ; FLAG-N (RRAO981 ; FLAG-N (RRAO981) ; FLAG-N (RRAO981 ; FLAG-N (RRAO981) ; FLAG-N (
S S S FC62: FC62: FC64:485 FC66:A85 FC66:A5 FC66:A5 FC70: FC70	199 197 198 200 201 202 203 204 207 208 205 208 207 210 211 212 213 214 215 214 215 214 215 214 215 214 212 212 213 214 215 212 212 212 223 224 225 222 223 224 225 222 223	CR LI SCR LI LF II CR SI LF II SCROLL EI XGOTOCX JI * TRQ SNIFFER * IRQ SNIFFER RDPAGE2 EI RDPAGE2 EI RDPAGE2 EI B RDPAGE2 EI B RDPAGE2 EI B RDPAGE2 EI B RDPAGE2 EI B S S RDCX S S	DY TTG TTA DA DTA DDA DTA DDA DDA DDA DDA DDA	#\$00 CH CLEDP1 #\$00 CH CV CV VTABZ CV APPLE2E #2 GOTOCX VIDEO CODE: \$C010 \$C010 RDB0STORE A RDPAGE2 RDCX \$C054	; AND H INDICES ; HEN CLEAR TO END OF PAGE. ; (ALMAYS TAKEN) ; (RET CURSOR TO LEFT OF INDEX ; (RET CURSOR H=O) ; INCR CURSOR V. (DDWN 1 LINE) ; OFF SCREEN? NO. SET BASE ADDR ; DECR CURSOR V. (BACK TO BOTTOM) ; /RA0981 ; CODE=SCRDLL/RRA0981 ; /RRA0981 ; /RRA0981 ; /RRA0981 ; /RRA0981 ; /RRA0981 ; /RRA0981 ; /RRA0981 ; /RRA0981 ; FLAG=>N /RRA0981 ; FLAG=>N /RRA0981 ; FLAG=>N /RRA0981 ; FLAG=>N /RRA0981 ; FDRCE ME TXTPAGE/RRA0981 ; FDRCE ME TXTPAGE/RRA0981 ; FLAG=>N /RRA0981 ; FDRCE ME TXTPAGE/RRA0981 ; FLAG=>N /RRA0981 ; FDRCE ME TXTPAGE/RRA0981 ; FLAG=>N /RRA0981 ; FDRCE ME TXTPAGE/RRA0981 ; FLAG=>N /RRA0981
S S S FC62: FC62: FC64: 85 24 FC66: 85 25 FC66: 85 25 FC66: 85 25 FC66: 70 84 FC70: FC70: FC70: FC70: FC70: FC70: FC77: FC75:	199 194 197 198 200 201 202 203 204 205 204 205 204 203 204 211 212 213 214 213 214 213 214 217 212 213 214 217 212 213 214 217 212 214 217 218 214 217 218 214 217 218 214 217 218 218 218 218 219 219 219 200 201 201 202 203 204 204 205 204 205 204 205 205 205 205 205 205 205 205 205 205	CR LI SCROLL EI XGOTOCX J * TRQ SNIFFER * TRQ SNIFFER * RDPAGE2 EI RDPAGE2 EI RDPAGE3 S RDCX S CC CC CC CC CC CC CC CC CC CC CC CC CC	DY ITY ITY ITY ITA DA ITA DA ITA DA MP CCC DO COU ITA CCC DO COU ITA CCC DO COU ITA CCC DO COU ITA CCC DA ITA CCCC DA ITA CCCC DA ITA CCCC DA ITA CCCC ITA CCCC ITA CCCC ITA CCCC ITA CCCC ITA CCCC ITA CCCC ITA CCCC ITA CCCC ITA CCCC ITA CCCC ITA CCCCC ITA CCCCC ITA CCCCC ITA CCCCCC ITA CCCCCCCCC ITA CCCCCCCCCC	#\$00 CH CLEOP1 #\$00 CH CV CV VTABZ CV APPLE2E # # GOTOCX VIDED CODE: * CO18 *C018 *C018 CO18 *C018 RDBOSTORE A RDBOSTORE A RDPAGE2 RDCX *CO34 *CO34 *CO34 *CO34 *CO34 *CO34	; AND H INDICES ; THEN CLEAR TO END OF PAGE. ; (ALMAYS TAKEN) ; (RAO981 ; (CURSOR TO LEFT OF INDEX ; (URSOR TO LEFT OF INDEX ; INCT CURSOR H=0) ; INCT CURSOR V. (DOWN 1 LINE) ; OFF SCREEN? ; NO, SET BASE ADDR ; DECC CURSOR V. (DACK TO BOTTOM) ; /RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; FLAG->N /RRAO981 ; FLAG->N /RRAO981 ; /RRAO981 ; JLAG->C /RRAO981 ; /RRAO981 ; /RRAO981 ; FLAG->N /RRAO981 ; /RRAO981 ; /RRAO981 ; FLAG->N /RRAO981 ; ELAG->N /RRAO981 ; ELAG->N /RRAO981 ; ELAG->N /RRAO981 ; ELAG->N /RRAO981 ; ENALE RAG/RRAO981
S S S S FC62: FC62: FC64:485 FC66:C6 FC66:C6 FC66:C6 FC70: FC	199 197 198 197 198 200 201 202 203 204 205 204 207 208 207 208 207 212 213 214 215 214 215 214 215 214 215 214 212 212 212 212 212 212 212 212 212	CR LI SCROLL EI SCROLL EI SCROLL EI XGOTOCX JJ * TRO SNIFFER RDPAGE2 EI RDPAGE2 EI RDPAGE2 EI RDPAGE2 EI RDPAGE2 EI B RDS STORE EI S RDCX S S S S	DY TY TY IN IN IN IN IN IN IN IN IN IN IN IN IN	#\$00 CH CLEOP1 #\$00 CV CV VTABZ CV APPLEZE #2 GOTOCX VIDED CODE: \$CO16 RDBOSTORE A RDPAGE2 RDCX \$CO54 RDCX \$CO54 RDCXRDM SETSLDTCXRDM	; AND H INDICES ; THEN CLEAR TO END OF PAGE. ; (ALMAYS TAKEN) ; (RAC0781 ; CURSOR TO LEFT OF INDEX ; (RET CURSOR H=O) ; INCR CURSOR V. (DOWN 1 LINE) ; OFF SCREEN? NO. SET BASE ADDR ; DECC CURSOR V. (BACK TO BOTTOM) ; /RRA0781 ; /RRA0781 ; CODE=SCROLL/RRA0781 ; /RRA0781 ; FLAG=CX (RRA0781 ; FLAG=XN (RRA0781 ; FLAG
S S S FC62: FC62: FC62: FC66: FC68: FC68: FC68: FC70:	199 196 197 198 200 201 202 203 204 205 206 207 208 200 210 211 212 212 212 212 214 215 214 215 214 215 214 217 220 221 222 222 222 222 222 222 222 222	CR LI SCROLL EI XGOTOCX J * IRQ SNIFFER * IRQ SNIFFER * RDPAGE2 EI RDPAGE2 EI RDPAGE2 EI B B RDCX S SC SC SC SC SC SC SC SC SC SC SC SC SC	DY ITY ITY ITY ITA ITA ITA ITA ITA ITA ITA ITA ITA ITA	#\$00 CH CLEOP1 #\$00 CH CV CV VTABZ CV APPLE2E #2 GOTOCX VIDEO CODE: *C016 #C016 #C016 #C016 RDB0STORE A RDPAGE2 RDCX %C054 %C054 %C054 %C054 %C054 %C054 %C054 %C054 %C054 %C054 %C054 %C055	; AND H INDICES ; THEN CLEAR TO END OF PAGE. ; (ALMAYS TAKEN) ; (RAO981 ; (CURSOR TO LEFT OF INDEX ; (URSOR TO LEFT OF INDEX ; INCR CURSOR V. (DOWN 1 LINE) ; OFF SCREEN? NO. SET BASE ADDR ; DECR CURSOR V. (DACK TO BOTTOM) ; /RRAO981 ; FLAG->C /RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; RESTORE AC/RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; RESTORE BANK/RAO981 ; FLAG->N /RRAO981 ; FLAG-N /RAO981 ; FLAG-SC /RRAO981 ; ENSTORE BANK/RAO981 ; ENSTORE BANK/RAO981 ; NOT BANKSNITCHING/RRAO981 ; RESTORE BANK/RAO981 ; FORGE JESUFS/RRAO981 ; NOW DISABLE/RRAO981
S S S S S FC62: A7 00 FC64: 85 24 FC66: 64 25 FC66: 65 25 FC66: 70 84 FC70: P84 FC70: FC70 FC70: A0 02 FC72: 4C 84 FC75: FC75: FC75: C018 FC75: C018 FC77: C018 FC78: C010 FC78: B0 FC8 FC8 FC8 FC8 FC8 FC8 FC78 FC9 FC9 FC7 FC7 FC7 FC7 FC7 FC7 FC7 FC7	199 197 197 198 200 201 202 203 204 205 204 205 204 207 208 207 208 207 212 213 212 213 214 215 214 215 214 215 214 215 212 212 212 212 212 212 212 212 212	CR LI SCROLL EI SCROLL EI SCROLL EI XGOTOCX JJ * IRQ SNIFFER RDBASTORE EI RDBASTORE EI RDBAGE2 EI RDBAGE2 EI B RDBAGE2 EI SS RDCX S S S S S S	DY TYGINA TROMA MMCCCC DO DO DO DO DO DO DO DO DO DO DO DO DO	#\$00 CH CLEOP1 #\$00 CV VTABZ CV APPLE2E #2 GOTOCX VIDED CODE: *CC18 *CC16 RDB0STORE A RDPAGE2 RDCX *CC54 RCC54 RDCX RDCX SETSLOTS SETINTCXROM	<pre>; AND H INDICES ; HEN CLEAR TO END OF PAGE. ; (ALMAYS TAKEN) ; (RAD098] ; CURSOR TO LEFT OF INDEX ; (RET CURSOR H=O) ; INCR CURSOR Y. (DDWN 1 LINE) ; OF SCREEN? NO, SET BASE ADDR ; DECR CURSOR V. (BACK TO BOTTOM) ; /RRAO981 ; (RAO981 ; CODE=SCROLL/RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; FLAG=XN /RRA</pre>
S S S FC662: FC622: AP 00 FC642: B5 24 FC662: A5 25 FC663: A5 25 FC663: A5 25 FC70; D84 FC70; C9 84 FC70; C001 FC70; A0 02 FC77; A0 02 FC77; C018 FC75; C018 FC76; FC76 FC76; FC76 FC76 FC76 FC76 FC76 FC76 FC76 FC76 FC76 FC76 FC76 FC76 FC76 F	199 196 197 198 200 203 203 204 205 204 205 204 205 204 207 203 204 205 204 207 210 211 212 213 214 215 214 215 214 215 214 215 214 215 212 223 224 225 226 225 226 225 226 225 226 225 226 225 226 225 226 225 226 225 226 225 226 225 225	RDCX B RDCX B SCROLL C SCROLL C SCROLL C SCROLL C SCROLL C SCROLL C SCROLL C SCROLC C	DY TY TY TEG INA DA DA DA DA DA DA DA DA DA DA DA DA DA	#\$00 CH CLEOP1 #\$00 CH CV CV VTABZ CV APPLE2E #2 GOTOCX VIDEO CODE: *C016 #C016 #C016 #C016 RDB0STORE A RDPAGE2 RDCX %C054 %C054 %C054 %C054 %C054 %C054 %C054 %C054 %C054 %C054 %C054 %C055	; AND H INDICES ; THEN CLEAR TO END OF PAGE. ; (ALMAYS TAKEN) ; (RAO981 ; CURSOR TO LEFT OF INDEX ; (CURSOR TO LEFT OF INDEX ; (RET CURSOR H=0) ; INCR CURSOR V. (DOWN 1 LINE) ; OFF SCREEN? ; NO. SET BASE ADDR ; DECR CURSOR V. (BACK TO BOTTOM) ; /RRA0981 ; /RRA0981 ; /RRA0981 ; /RRA0981 ; /RRA0981 ; /RRA0981 ; PRESCRUE AC/RRA0981 ; FLAO=-N /RRA0981 ; FLAO=N /RRA0981 ; FLAO=R STATAGE/RRA0981 ; FLAO=R STATAGE/RRA0981 ; FLAO=R BANK/RRA0981 ; FLAO=R BANK/RRA0981 ; FLAO=SU / RRA0981 ; FLAO=SU / RRA0981 ; FLAO=SU / RRA0981 ; FLAO=SU / SCRUCK ; CASU / SCRUCK ; FLAO=SU /
S S S S S FC62: AP 00 FC64: 85 24 FC66: 64 25 FC66: 65 25 FC66: 70 86 FC70: C001 FC70: FC70 FC70: FC70 FC70: A0 02 FC72: 4C 84 FC75: FC75: FC75: C018 FC75: C018 FC75: C018 FC75: C018 FC75: C018 FC75: C018 FC75: C018 FC75: C018 FC75: C018 FC75: C018 FC77: 0A FC76: AD 18 C075: C018 FC77: C018 FC78: C00 FC78: B0 FC78: B0 FC78: B0 FC78: B0 FC78: FC88 FC78: FC88 FC79: FC78 FC78: FC78	199 197 198 200 203 203 204 205 207 203 204 205 207 210 212 212 213 214 215 214 215 214 215 214 215 214 215 212 222 223 224 221 221 221 221 222 223 224 220 221 222 223 224 220 221 222 223 224 227 226 227 228 227 228 227 228 227 228 227 228 227 228 227 228 227 228 227 228 227 228 227 228 227 228 229 229 229 229 229 229 229 229 229	CR LI SCROLL E SCROLL E SCROLL E SCROLL E SCROLL E XGOTOCX J * IRQ SNIFFER RDPAGE2 E RDPAGE2 E RDPAGE2 E S SCROLS S S S S S S S S S S S S S S S S S S S	DY TTG IDA DTAC DA DTAC DA DTAC DA DE DE DE DE DE DE DE DE DE DE DE DE DE	#\$00 CH CLEOP1 #\$00 CH CV VTABZ CV APPLEZE #2 GOTOCX VIDED CODE: *CO16 RDBOSTORE A RDPAGE2 RDCX *CO54 *CO54 RDCX *CO54 S	<pre>; AND H INDICES ; HEN CLEAR TO END OF PAGE. ;(ALMAYS TAKEN) ;/RAO981 ;CURSOR TO LEFT OF INDEX ;(RET CURSOR H=O) ;INCR CURSOR Y. (DOWN 1 LINE) ;OFF SCREEN? NO, SET BASE ADDR ;DECR CURSOR V. (BACK TO BOTTOM) ;/RAO981 ;/RAO981 ;CDDE-SCROLL/RAO981 ;/RAO981 ;CDDE-SCROLL/RAO981 ;/ARAO981 ;/ARAO981 ;/ARAO981 ;/ARAO981 ;/ARAO981 ;/ARAO981 ;/ARAO981 ;FLAG->N /RAO981 ;FLAG->N /RAO981 ;FLAG-N /FLAG-N /FLAG981 ;FLAG-N /FLAGPAN ;FLAG-N /FLAGPAN ;FLAGN /FLAGPAN ;FLAGN /FLAGPAN ;FLAGN /F</pre>
S S S FC662: FC662: FC662: FC662: FC662: FC662: FC662: FC70:	199 196 197 198 200 203 203 204 205 204 205 200 200 200 200 200 200 200 200 200	RDCX B RDCX B SCROLL E SCROLL E XGOTOCX J * IRQ SNIFFER * IRQ SNIFFER * RDPAGE2 E RDPAGE2 E RDPAGE2 SC SC SC SC SC SC SC SC SC SC	DY TTY TTY TTY TTY TTY TTY TTY TTY TTY T	#\$00 CH CLEOP1 #\$00 CH CV CV VTABZ CV APPLE2E * * 200TOCX VIDE0 CODE: * CO18 CO18 * CO18 CO18 CO18 CO18 * CO18 * CO18 CO18 CO18 CO18 CO18 CO18 CO18 CO18	; AND H INDICES ; THEN CLEAR TO END OF PAGE. ; (ALMAYS TAKEN) ; (RAO981 ; CURSOR TO LEFT OF INDEX ; (RET CURSOR H=0) ; INCR CURSOR V. (DOWN 1 LINE) ; OFF SCREEN? ; NO. SET BASE ADDR ; DECR CURSOR V. (BACK TO BOTTOM) ; /RRAO981 ; /RRAO981 ; /RRAO981 ; /RRAO981 ; PAESERVE AC/RRAO981 ; PAESERVE AC/RRAO981 ; FLAG->N /RRAO981 ; FLAG-SC B TXTFAGE/RRAO981 ; FLAG-SC B TXTFAGE/RRAO981 ; FLAG-SC B TXTFAGE/RRAO981 ; FLAG-SC SUT5/RRAO981 ; BANK-IN CX/RRAO981 ; BANK-IN CX/RRAO981 ; BANK-IN CX/RRAO981 ; JRRAO981 ; WAN TYID BANKED/RRAO981 ; HAH TYID BANKED/RRAO981
S S S S S S FC62: AP 00 FC64: 85 24 FC66: 64 25 FC66: 65 25 FC66: 70 86 FC70: C001 FC70: FC70 FC70: C001 FC70: A0 02 FC72: 4C 84 FC75: FC75: FC75: C018 FC75: C018 FC75: C018 FC75: C018 FC75: C018 FC75: C018 FC75: C018 FC75: C018 FC75: C018 FC75: C018 FC77: 0A 18 FC76: AD 18 C075: C018 FC77: C018 FC77: C018 FC77: C018 FC77: C018 FC77: C018 FC77: C03 FC78: AD 18 FC79: O3 FC78: B0 FC78: B0 FC79: FC78: C0 FC78: B0 FC78: C0 FC78: B0 FC78: B0 FC78: B0 FC78: B0 FC78: C0 FC78: B0 FC78: C0 FC78: B0 FC78: C0 FC78: B0 FC78: C0 FC78: B0 FC78: C0 FC78: B0 FC78: C0 FC78: C0 FC78: C0 FC78: B0 FC78: B0 FC78: B0 FC78: B0 FC78: B0 FC78: B0 FC78: B0 FC78: C0 FC78: B0 FC78: C0 FC78: B0 FC78: B0	199 197 198 200 201 202 203 204 205 207 210 211 212 212 212 213 214 215 216 216 217 218 216 211 212 213 214 215 216 212 212 223 222 222 222 222 222 222 222	CR LI SCROLL E SCROLL E SCROLL E XGOTOCX J * IRQ SNIFFER * IRQ SNIFFER * RDPAGE2 E RDPAGE2 E B RDCX B S SCROLS S S S S S S S S S S S S S S S S S S S	DY TTY TTY TTY TTY TTY TTY TTY TTY TTY T	*00 CH CLEOP1 **00 CC CV VTABZ CV APPLE2E ** #2 GOTOCX VIDED CODE: *CO16 RDBOSTORE A RDPAGE2 RDCX *CO54 *CO54 RDCX *CO54 RDCX *CO54 SETSLOTS SETSLOTS SETSLOTS SETSLOTS SETSLOTS SETSLOTS SETSLOTS	<pre>; AND H INDICES ; HEN CLEAR TO END OF PAGE. ; (ALMAYS TAKEN) ; (RARO981 ; CURSOR TO LEFT OF INDEX ; (RET CURSOR H=0) ; INCR CURSOR Y. (DDWN 1 LINE) ; OFF SCREEN? NO, SET BASE ADDR ; DECR CURSOR V. (BACK TO BOTTOM) ; /RRAO981 ; /RRAO981 ; /RRAO981 ; CDDE=SCROLL/RRAO981 ; /RRAO981 ; FLAG=>N /RRAO981 ; FLAG=N /RRAO981 ; FLAG=N /RRAO981 ; FLAG=N /RRAO981 ; FLAG=N /RRAO981 ; FLAG=N /RRAO981 ; FLAG=N /RAO981 ; SANK=IN CX/RRAO981 ; SANK=IN CX/RRAO981 ; JRAG981 ; JRAG981 ;</pre>
S S S FC662: FC662: FC662: FC662: FC662: FC662: FC662: FC70:	199 197 197 200 202 203 204 203 203 204 203 205 204 207 208 207 208 209 210 211 212 213 214 211 211 211 211 211 211 211 211 211	RDCX SISSLOTS E ISSLOTS E ISSLOTS I ISSLOTS I ISSLO	DY TEG IDA DA DA DA DA DA DA DA DA DA DA DA DA D	#\$00 CH CLEOP1 #\$00 CH CV CV VTABZ CV APPLE2E * * 200TOCX VIDE0 CODE: * CO18 CO18 * CO18 CO18 CO18 CO18 * CO18 * CO18 CO18 CO18 CO18 CO18 CO18 CO18 CO18	; AND H INDICES ; THEN CLEAR TO END OF PAGE. ; (ALMAYS TAKEN) ; (RAO981 ; CURSOR TO LEFT OF INDEX ; (CURSOR TO LEFT OF INDEX ; (RET CURSOR H=0) ; INCR CURSOR V. (DOWN 1 LINE) ; OFF SCREEN? ; NO. SET BASE ADDR ; DECR CURSOR V. (DACK TO BOTTOM) ; DECR CURSOR V. (BACK TO BOTTOM) ; JCRA0981 ; /RRA0981 ; /RRA0981 ; CODE=SCROLL/RRA0981 ; CODE=SCROLL/RRA0981 ; PAESERVE AC/RRA0981 ; FLAQ=-N /RRA0981 ; FLAQ=N /RRA0981 ; FLAQ=N /RRA0981 ; FLAQ=N /RRA0981 ; FLAQ=N /RRA0981 ; FLAQ=N /RRA0981 ; FLAG=ST /RRA0981 ; NOT BANKEN/RRA0981 ; JCRA0981 ; MANEL/RRA0981 ; JCRA0981 ; MANEL/RRA0981 ; JCRA0981 ; MANEL/RRA0981 ; JCRA0981 ; HAT VID BANKE/ARA0981 ; JCRA0981 ; HAT VID BANKE/ARA0981 ; JCRA0981 ; HAT VID BANKE/ARA0981 ; JCRA0981 ; HAT VID BANKE/ARA0981 ; JCREAPSE/RRA0981 ; JFORCE PASEZ/RRA0981
S S S S S S FC62: AP 00 FC64: 85 24 FC66: 64 25 FC66: 65 25 FC66: 70 86 FC70: C001 FC70: FC70 FC70: C001 FC70: A0 02 FC72: 4C 84 FC75: FC75: FC75: C018 FC75: C018 FC75: C018 FC75: C018 FC75: C018 FC75: C018 FC75: C018 FC75: C018 FC75: C018 FC75: C018 FC77: 0A 18 FC76: AD 18 C075: C018 FC77: C018 FC77: C018 FC77: C018 FC77: C018 FC77: C018 FC77: C03 FC78: AD 18 FC79: O3 FC78: B0 FC78: B0 FC79: FC78: C0 FC78: B0 FC78: C0 FC78: B0 FC78: B0 FC78: B0 FC78: B0 FC78: C0 FC78: B0 FC78: C0 FC78: B0 FC78: C0 FC78: B0 FC78: C0 FC78: B0 FC78: C0 FC78: B0 FC78: C0 FC78: C0 FC78: C0 FC78: B0 FC78: B0 FC78: B0 FC78: B0 FC78: B0 FC78: B0 FC78: B0 FC78: C0 FC78: B0 FC78: C0 FC78: B0 FC78: B0	199 197 197 200 202 203 204 203 203 204 203 205 204 207 208 207 208 209 210 211 212 213 214 211 211 211 211 211 211 211 211 211	RDCX ISSLOTS ISSLOT	DY TTY TTY TTY TTY TTY TTY TTY TTY TTY T	#\$00 CH CLEOP1 #\$00 CV CV VTAB2 CV APPLE2E #2 GOTOCX VIDEO CODE: \$C016 CC18 \$C016 RDB0STORE A RDPAGE2 RDCX RDCXROM SETSLOTCXROM ISSLOTS SETINTCXROM * ISPAGE1 19PAGE1 19PAGE1 \$C055	<pre>; AND H INDICES ; HEN CLEAR TO END OF PAGE. ; (ALMAYS TAKEN) ; (RARO981 ; CURSOR TO LEFT OF INDEX ; (RET CURSOR H=0) ; INCR CURSOR Y. (DDWN 1 LINE) ; OFF SCREEN? NO, SET BASE ADDR ; DECR CURSOR V. (BACK TO BOTTOM) ; /RRAO981 ; /RRAO981 ; /RRAO981 ; CDDE=SCROLL/RRAO981 ; /RRAO981 ; FLAG=>N /RRAO981 ; FLAG=N /RRAO981 ; FLAG=N /RRAO981 ; FLAG=N /RRAO981 ; FLAG=N /RRAO981 ; FLAG=N /RRAO981 ; FLAG=N /RAO981 ; SANK=IN CX/RRAO981 ; SANK=IN CX/RRAO981 ; JRAG981 ; JRAG981 ;</pre>

FC9A: EA FC9B: EA	237 238	NOP		;/RRA0981 ;/RRA0981
FC9C:	239	ELSE		7/RRA0981
S	240 SCROLL 241	LDA	WNDTOP	START AT TOP OF SCROLL WINDOW
5	241 242	JSR	VTABZ	GENERATE BASE ADDRESS
S	243 SCRL1	LDA	BASL	COPY BASL, H
5	244 245	STA	BAS2L BASH	; TO BAS2L, H
5	245	STA	BAS2H	
S	247	LDY	WNDWDTH	INIT Y TO RIGHTMOST INDEX
S	248 249	DEY		; OF SCROLLING WINDOW
5	250	PLA	#\$01	; INCR LINE NUMBER
S	251	CMP	WNDBTM	; DONE?
S	252 253	BCS PHA	SCRL3	; YES, FINISH
5	254	JSR	VTABZ	FORM BASL, H (BASE ADDR)
5	255 SCRL2	LDA	(BASL), Y	MOVE A CHAR UP ONE LINE
S	256 257	STA	(BAS2L), Y	NEXT CHAR OF LINE
S	258	BPL	SCRL2	MEXT CHAR OF CINE
S	259	BMI	SCRL1	INEXT LINE
S	260 SCRL3 261	LDY JSR	#\$00 CLEOLZ	CLEAR BOTTOM LINE
S	262	BCS	VTAB	CARRY IS SET
FC9C:	263	FIN		; /RRA0981
FC9C: 0001 FC9C: FC9C	264 265 CLREDL	DO EQU	APPLE2E	; /RRA0981 ; /RRA0981
FC9C: 18	266	CLC		SAY 'EOL'/RRA0981
FC9D: BO FC9E: FC9E	267 268 CLREDLZ	DFB	\$BO *	; 'BCS' OPCODE /RRA0981
FC9E: 38	268 CLREULZ 269	SEC	×	;/RRA0981 ;SAY 'EOLZ'/RRA0981
FC9F:84 1F	270	STY	\$1F	; VIDED'S YSAV1/RRA0981
FCA1: A0 03 FCA3: 90 CD FC72	271 272	LDY BCC	#3 XGOTOCX	;CODE=EOL/RRA0981 ;->IT'S EOL/RRA0981
FCA5: CB	273	INY	X00100X	CODE-EOLZ/RRA0981
FCA6: DO CA FC72	274	BNE	XGOTOCX	->ALWAYS/RRA0981
FCA8:	275 276 CLREDL	ELSE	сн	;/RRA0981 ;CURSOR H INDEX
s	277 CLEOLZ	LDA	#\$A0	STORE BLANKS FROM 'HERE'
S	278 CLEOL2	STA	(BASL), Y	; TO END OF LINES (WNDWDTH)
S	279	INY	WNDWDTH	
S	281	BCC	CLEDL2	
5	282	RTS		
FCA8: FCA8: 38	283 284 WAIT	FIN		;/RRA0981
FCA9: 48	285 WAIT2	PHA		
FCAA: E9 01 FCAC: DO FC FCAA	286 WAIT3 287	SBC	#\$01	
FCAC: DO FC FCAA FCAE: 68	288	PLA	WAIT3	;1.0204 USEC ;(13+2712*A+512*A*A)
FCAF: E9 01	289	SBC	#\$01	
FCB1: DO F6 FCA9 FCB3: 60	290 291	BNE	WAIT2	
FCB4: E6 42	292 NXTA4	INC	A4L	INCR 2-BYTE A4
FCB6: DO 02 FCBA	273	BNE	NXTA1	; AND A1
FCB8:E6 43 FCBA:A5 3C	294 295 NXTA1	INC LDA	A4H A1L	; INCR 2-BYTE A1.
FCBC: C5 3E	296	CMP	A2L	; AND COMPARE TO A2
FCBE: A5 3D FCC0: E5 3F	297 298	LDA	A1H A2H	; (CARRY SET IF >∞)
FCC2: E6 3C	299	INC	AZH	
FCC4: DO 02 FCC8	300	BNE	RTS4B	
FCC6: E6 3D FCC8: 60	301 302 RTS4B	INC	A1H	
FCC9: AO 4B	303 HEADR	LDY	#\$4B	WRITE A*256 'LONG 1'
FCCB: 20 DB FC	304	JSR	ZERDLY	HALF CYCLES
FCCE: DO F9 FCC9 FCD0: 69 FE	305 306	BNE ADC	HEADR #SFE	; (650 USEC EACH)
FCD2: BO F5 FCC9	307	BCS	HEADR	; THEN A 'SHORT O'
FCD4: A0 21	308 309 WRBIT	LDY	#\$21 ZERDLY	; (400 USEC) ;WRITE TWO HALF CYCLES
FCD4:20 DB FC FCD9:C8	309 WRBIT 310	JSR	ZERDLY	; DE 250 USEC ('0')
FCDA: CB	311	INY		; OR 500 USEC ('1')
FCDB:88 FCDC:D0 FD FCDB	312 ZERDLY 313	DEY BNE	ZERDLY	
FCDE: 90 05 FCE5	314	BCC	WRTAPE	Y IS COUNT FOR
FCE0: A0 32 FCE2: 88	315	LDY	#\$32	; TIMING LOOP
FCE2: BB FCE3: DO FD FCE2	316 ONEDLY 317	DEY	ONEDLY	
FCE5: AC 20 CO	318 WRTAPE	LDY	TAPEOUT	
FCEB: AO 2C	319	LDY	#\$2C	
FCEA: CA FCEB: 60	320 321	DEX		
FCEC: A2 08	322 RDBYTE	LDX	#\$08	B BITS TO READ
FCEE: 48 FCEF: 20 FA FC	323 RDBYT2 324	PHA JSR	RD2BIT	;READ TWO TRANSITIONS ; (FIND EDGE)
FCF2: 68	324	PLA	NDEDI	, THE EAST,
FCF3: 2A	326	ROL	A	NEXT BIT
FCF4: AO 3A FCF6: CA	327 328	LDY DEX	#\$3A	COUNT FOR SAMPLES
	320			

FCF7: D0 F5 FCEE	329	BNE	RDBYT2	
FCF9:60 FCFA:20 FD FC	330 331 RD2811	RTS	RDBIT	
FCFD: 88	332 RDBIT	DEY	RUBIT	DECR Y UNTIL
FCFE: AD 60 CO	333	LDA	TAPEIN	DECR Y UNTIL TAPE TRANSITION
FD01:45 2F FD03:10 F8 FCFD	334 335	EOR	LASTIN	
FD03:10 F8 FCFD FD05:45 2F	335	BPL EOR	LASTIN	
FD07:85 2F	337	STA	LASTIN	
FD09:C0 80 FD08:60	338 339	CPY	#\$80	SET CARRY ON Y-REG.
FDOC: A4 24	339 340 RDKEY	LDY	сн	
FDOE: B1 28	341	LDA	(BASL), Y	SET SCREEN TO FLASH
FD10:48	342	PHA		
FD11:29 3F FD13:09 40	343 344	AND ORA	#\$3F #\$40	
FD15:91 28	345	STA	(BASL), Y	
FD17:68	346	PLA		
FD18:6C 38 00 FD18: 0001	347 348	JMP DO	(KSWL) APPLE2E	GO TO USER KEY-IN
FD1B. FD1B	348 349 KEYIN	EQU	*	
FD1B: A0 06	350	L.DY	#6	RDKEY/RRA0981
FD1D: 4C B4 FB	351	JMP	GOTOCX	/RRA09B1
FD20: EA	352 353 RDESC	NOP	*	/ /RRA0981
FD21:20 0C FD	354	JSR	RDKEY	GET A KEY
FD24: A0 07	355	LDY	#7	CODE=FIXIT
FD26:4C B4 FB FD29:	356 357 *	JMP	GOTOCX	
FD29		ROM COT	OCX HERE:	
FD29:	359 *			
FD29:8D 06 CO	360	STA	SETSLOTCXROM	RESTORE BANK/RRA0981
FD2C: 28 FD2D: 60	361 362	PLP RTS		RESTORE IRG/RRA0981
FD2E:	363	ELSE		; /RRA0981
S	364 KEYIN	INC	RNDL	INCR RND NUMBER
S	365 366	BNE	KEYIN2 RNDH	
S	367 KEYIN2	BIT	KBD	KEY DOWN?
S	368	BPL	KEYIN	; LOOP UNTIL WE GET ONE
S	369 370	STA	(BASL), Y KBD	REPLACE FLASHING SCREEN
5	371	BIT	KBDSTRB	;GET KEYCODE ;CLR KEY STROBE
FD2E:	372	FIN		7/RRA0981
FD2E: 60	373	RTS		
FD2F: 0001 FD2F: 20 21 FD	374 375 ESC	DO JSR	APPLE2E RDESC	; /RRA0981 ; /RRA0981
FD32:	376	ELSE	NDEBC	; /RRA0981
S	377 ESC	JSR	RDKEY	GET KEYCODE
FD32: FD32:20 A5 FB	378 379	FIN JSR	ESCNEW	; /RRA0981
FD35:20 OC FD	380 RDCHAR	JSR	RDKFY	;HANDLE ESC FUNCTION. ;GO READ KEY
FD38: C9 98	381	CMP	#\$9B	7 1ESC 1?
FD3A: FO F3 FD2F	382	BEQ	ESC	YES, DON'T RETURN.
FD3C: 60 FD3D: A5 32	383 384 NOTCR	RTS LDA	INVFLG	
FD3F: 48	385	PHA	INVICO	
FD40: A9 FF	386	LDA	#\$FF	
FD42: 0001 FD42: EA	387 388	DO NOP	APPLE2E	;/RRA0981 ;DON'T CHANGE INPUT/RRA0981
FD43: EA	389	NOP		TO NORMAL/RRA0981
FD44:	390	ELSE.		; /RRA0981
S FD44:	391	STA	INVFLG	CONVERT TYPED CHAR TO 'NORMAL'
FD44: FD44:BD 00 02	372 373	FIN LDA	IN, X	;/RRA0981
FD47:20 ED FD	394	JSR	COUT	FCHD TYPED CHAR
FD4A: 68	395	PLA	TANKE C	
FD4B:85 32 FD4D:BD 00 02	396 397	STA	INVFLG IN, X	
FD50:C9 88	378	CMP	#\$88	CHECK FOR EDIT KEYS
FD52: FO 1D FD71	399	BEG	BCKSPC	- BACKSPACE
FD54:C9 98 FD56:F0 0A FD62	400 401	CMP BEQ	#\$98 CANCEL	; - CONTROL-X
FD58: E0 F8	402	CPX	#\$F8	
FD5A: 90 03 FD5F	403	BCC	NOTCR1	MARGIN?
FD5C:20 3A FF FD5F:E8	404 405 NOTCR1	JSR	BELL	YES, SOUND BELL
FD60:D0 13 FD75	405 NOTCR1 406	INX	NXTCHAR	ADVANCE INPUT INDEX
FD62: A9 DC	407 CANCEL	LDA	#\$DC	BACKSLASH AFTER CANCELLED LINE
FD64: 20 ED FD	408	JSR	COUT	
FD67:20 8E FD FD6A:A5 33	409 GETLNZ 410 GETLN	JSR	CROUT	DUTPUT 'CR' DUTPUT PROMPT CHAR
FD6C:20 ED FD	411	JSR	COUT	
FD6F: A2 01	412	LDX	#\$01	; INIT INPUT INDEX
FD71:8A FD72:F0 F3 FD67	413 BCKSPC 414	TXA	GETLNZ	WILL BACKSPACE TO O
FD74: CA	415	DEX		
FD75:20 35 FD	416 NXTCHAR	JSR	RDCHAR	
FD78:C9 95 FD7A:D0 02 FD7E	417 418	CMP BNE	#\$95 CAPTST	USE SCREEN CHAR
FD7C: B1 28	419	LDA	(BASL), Y	I TOK CONTROL-V
FD7E:C9 E0	420 CAPTST	CMP	#\$E0	LOWER CASE?

FD80: 90 02 FD84	421	BCC		
FD82: 0001	421	DO	ADDINP APPLEZE	/RRA0981
FD82: 29 FF	423	AND	#\$FF	DON'T CONVERT TO UPPER CASE!/RRA0981
FD84:	424	ELSE	****	/RRA0781
s	425	AND	##DF	SHIFT TO UPPER CASE
FDB4:	426	FIN		/RRA0981
FD84: 9D 00 02	427 ADDINP	STA	IN, X	ADD TO INPUT BUFFER
FD87: C9 8D	428	CMP	#\$8D	
FD89: DO B2 FD3D	429	BNE	NOTCR	
FD88:20 9C FC	430	JSR	CLREDL	CLR TO EOL IF CR
FD8E: A9 8D	431 CROUT	LDA	#\$8D	
FD90: DO 5B FDED	432	BNE	COUT	; (ALWAYS)
FD92: A4 3D	433 PRA1	LDY	A1H	PRINT CR, A1 IN HEX
FD94: A6 3C	434	LDX	A1L	
FD96:20 8E FD	435 PRYX2	JSR	CROUT	
FD99:20 40 F9	436	JSR	PRNTYX	
FD9C: A0 00	437	LDY	#\$00	
FD9E: A9 AD	438	LDA	#\$AD	PRINT '-'
FDAO: 4C ED FD FDA3: A5 3C	439 440 XAMB	JMP	COUT	
	440 XAM8 441		A1L #\$07	OFT TO CINICULAT
FDA5:09 07 FDA7:85 3E	442	ORA STA	A2L	;SET TO FINISH AT ; MOD 8≖7
FDA7: 65 3D	443	LDA	AIH	, MUD 8-7
FDAB: 85 3F	444	STA	A2H	
FDAD: A5 3C	445 MOD8CHK	LDA	AIL	
FDAF: 29 07	446	AND	#\$07	
FDB1: DO 03 FDB6	447	BNE	DATADUT	
FDB3: 20 92 FD	448 XAM	JSR	PRA1	
FDB6: A7 A0	449 DATADUT	LDA	#\$A0	
FDB8: 20 ED FD	450	JSR	COUT	JOUTPUT BLANK
FDBB: B1 3C	451	LDA	(A1L), Y	
FDBD: 20 DA FD	452	JSR	PRBYTE	OUTPUT BYTE IN HEX
FDCO: 20 BA FC	453	JSR	NXTA1	
FDC3: 90 EB FDAD	454	BCC	MODSCHK	NOT DONE YET. GO CHECK MOD 8
FDC5: 60	455 RTS4C	RTS		i DONE.
FDC6: 4A	456 XAMPM	LSR	A	DETERMINE IF MONITOR MODE IS
FDC7:90 EA FDB3	457	BCC	XAM	; EXAMINE, ADD OR SUBTRACT
FDC9: 4A	458	LSR	A	
FDCA: 4A	459	LSR	A	
FDCB:A5 3E FDCD:90 02 FDD1	460 461	LDA	A2L ADD	
FDCD: 40 02 FDD1	461	EOR	ADD #SEE	FORM 2'S COMPLEMENT FOR SUBTRACT.
FDD1:65 3C	463 ADD	ADC	AIL	FURN 2 5 CONFLEMENT FUR SUBTRACT.
FDD3: 48	464	PHA	HIL	
FDD4: A9 BD	465	LDA	#\$BD	PRINT '=', THEN RESULT
FDD6: 20 ED FD	466	JSR	COUT	Then the theorem
FDD9: 68	467	PLA		
FDDA: 48	468 PRBYTE	PHA		PRINT BYTE AS 2 HEX DIGITS
FDDB: 4A	469	LSR	Α	(DESTROYS A-REG)
FDDC: 4A	470	LSR	A	
FDDD: 4A	471	LSR	A	
FDDE: 4A	472	LSR	A	
FDDF:20 E5 FD	473	JSR	PRHEXZ	
FDE2: 68	474	PLA		
FDE3: 29 OF	475 PRHEX	AND	#\$0F	PRINT HEX DIGIT IN A-REG
FDE5: 09 BO	476 PRHEXZ	ORA	#\$B0	LSBITS ONLY.
FDE7:C9 BA FDF9:90 02 FDFD	477	CMP	#\$BA	
FDE3: 67 06	478 479	BCC ADC	COUT #\$06	
FDED: 6C 36 00	480 COUT	JMP	# \$ 08 (CSWL)	VECTOR TO USER OUTPUT ROUTINE
FDF0: C9 A0	481 COUT1	CMP	(USWL) #\$A0	FUTUR TO OBER DOTEVT ROUTINE
FDF2:90 02 FDF6	482	BCC	COUTZ	DON'T DUTPUT CTRL'S INVERSE
FDF4: 25 32	483	AND	INVELG	MASK WITH INVERSE FLAG
FDF6:84 35	484 COUTZ	STY	YSAV1	SAVE Y-REG
FDF8: 48	485	PHA		SAVE A-REG
FDF9: 20 78 FB	486	JSR	VIDWAIT	; DUTPUT CHR & CHECK FOR CTRL-S
FDFC: 68	487	PLA		; RESTORE A-REG
FDFD: A4 35	488	LDY	YSAV1	; AND Y-REG
FDFF: 60	489	RTS		; RETURN TO SENDER
FE00: C6 34	490 BL1	DEC	YSAV	
FE02: FO 9F FDA3	491	BEQ	XAMB	
FE04: CA FE05: DO 16 FE1D	492 BLANK 493	DEX	SETMDZ	BLANK TO MON
FE07: C9 BA	493	CMP	#\$BA	DATA STORE MODE?
FE07: DO BB FDC6	495	BNE	##BA	IDATA STORE MODE?
FEOB: 85 31	496 STOR	STA	MODE	
FEOD: A5 3F	496 51UR 497	LDA	A2L	KEEP IN STORE MODE
FE0F: 91 40	478	STA	(A3L),Y	STORE AS LOW BYTE AT (A3)
FE11:E6 40	499	INC	AGL	CONTRACTOR FOR PUTCHING TO THE
FE13: D0 02 FE17	500	BNE	RTS5	INCR A3, RETURN.
FE15: E6 41	501	INC	A3H	
FE17:60	502 RTS5	RTS	10001000	
FE18: A4 34	503 SETMODE	LDY	YSAV	SAVE CONVERTED 1: 1, 1+1,
FE1A: B9 FF 01	504	LDA	IN-1, Y	; '-', '.' AS MODE
FE1D:85 31	505 SETMDZ	STA	MODE	
FE1F: 60	506	RTS		
FE20: A2 01	507 LT	LDX	#\$01	
FE22: B5 3E	508 LT2	LDA	A2L, X	COPY A2 (2 BYTES) TO
FE24:95 42	509 510	STA	A4L, X	; A4 AND A5
		STA	ASL, X	
FE26: 95 44	510			
FE28: 95 44 FE28: CA FE29: 10 F7 FE22	511 512	DEX	LT2	

FE2B: 60	513	RTS		MOVE (A1) THRU (A2) TO (A4)
FE2C: B1 3C FE2E: 91 42	514 MOVE 515	LDA	(A1L),Y (A4L),Y	(MOVE (AI) THRO (A2) TO (A4)
	516	JSR	NXTA4	
FE30:20 B4 FC FE33:90 F7 FE2C	517	BCC	MOVE	
FE35: 60	518	RTS	110VL	
FE36: B1 3C	519 VFY	LDA	(A1L), Y	VERIFY (A1) THRU (A2)
FE38: D1 42	520	CMP	(A4L), Y	; WITH (A4)
FE3A: FO 1C FE58	521	BEQ	VEYOK	
FE3C: 20 92 FD	522	JSR	PRA1	
FE3F: B1 3C	523	LDA	(A1L), Y	
FE41:20 DA FD	524	JSR	PRBYTE	
FE44: A9 A0	525	LDA	#\$A0	
FE46:20 ED FD	526	JSR	COUT	
FE49: A9 AB	527	LDA	#\$A8	
FE4B:20 ED FD	528	JSR	COUT	
FE4E: B1 42	529	LDA	(A4L), Y	
FE50:20 DA FD	530	JSR	PRBYTE	
FE53: A9 A9	531	LDA	#\$A9	
FE55:20 ED FD	532	JSR	COUT	
FE58:20 B4 FC	533 VFYOK	JSR	NXTA4	
FE5B: 90 D9 FE36	534	BCC	VFY	
FE5D: 60	535	RTS		NOUT AL LO DUTEON TO
FE5E:20 75 FE FE61 A9 14	536 LIST 537	JSR	A1PC #\$14	; MOVE A1 (2 BYTES) TD ; PC IF SPEC'D AND
	538 LIST2	PHA	##14	; DISASSEMBLE 20 INSTRUCTIONS.
FE63:48 FE64:20 D0 F8	539	JSR	INSTDSP) DISASSENDLE 20 INSTRUCTIONS.
FE67:20 53 F9	540	JSR	PCADJ	ADJUST PC AFTER EACH INSTRUCTION.
FE6A: 85 3A	541	STA	PCL.	THEODER TO HE FER EIGHT INSTRUCTION
FE4C: 84 3B	542	STY	PCH	
FE6E: 68	543	PLA	1 011	
FE6F: 38	544	SEC		
FE70: E9 01	545	SBC	#\$01	NEXT OF 20 INSTRUCTIONS
FE72: DO EF FE63	546	BNE	LIST2	
FE74: 60	547	RTS		
FE75: 8A	548 A1PC	TXA		; IF USER SPECIFIED AN ADDRESS,
FE76: F0 07 FE7F	549	BEQ	A1PCRT5	; COPY IT FROM A1 TO PC.
FE78: 85 30	550 A1PCLP	LDA	A1L . X	YEP, SO COPY IT.
FE7A: 95 3A	551	STA	PCL, X	
FE7C: CA	552	DEX		
FE7D: 10 F9 FE78	553	BPL	AIPCLP	
FE7F: 60	554 A1PCRTS	RTS		
FE80: A0 3F	555 SETINV	LDY	#\$3F	SET FOR INVERSE VID
FE82: DO 02 FE86	556	BNE	SETIFLG	; VIA COUTI
FE84: AO FF	557 SETNORM	LDY	#\$FF	SET FOR NORMAL VID
FE86:84 32	558 SETIFLG	STY	INVFLG	
FE88:60 FE89:A9 00		LDA	#\$00	DO (IN#O)
FE89: A9 00 FE8B: 85 3E	560 SETKBD 561 INPORT	STA	#\$00 A2L	DO 'INHAREG'
FEBD: A2 38	562 INPRT	LDX	#KSWL	/ DO INWAREO
	563	LDY	#KEYIN	
		DNE		
FE91: DO 08 FE9B	564	BNE	10PRT	:DD /PR#0/
FE91: D0 08 FE9B FE93: A9 00	564 565 SETVID	LDA	#\$00	;DD 'PR#O' ;DD 'PR#AREG'
FE91: D0 08 FE9B FE93: A9 00 FE95: 85 3E	564 565 SETVID 566 OUTPORT	LDA	#\$00 A2L	;DO 'PR#O' ;DO 'PR#AREG'
FE91: D0 08 FE9B FE93: A9 00 FE95: 85 3E FE97: A2 36	564 565 SETVID	LDA STA LDX	#\$00 A2L #CSWL	
FE91: D0 08 FE9B FE93: A9 00 FE95: 85 3E FE97: A2 36	564 565 SETVID 566 OUTPORT 567 OUTPRT	LDA	#\$00 A2L	DO 'PR#AREG'
FE91: D0 08 FE9B FE93: A9 00 FE95: B5 3E FE97: A2 36 FE99: A0 F0	564 565 SETVID 566 OUTPORT 567 OUTPRT 568	LDA STA LDX LDY	#\$00 A2L #CSWL #COUT1	
FE91: D0 08 FE98 FE93: A9 00 FE95: 85 3E FE97: A2 36 FE99: A0 F0 FE98: A5 3E FE90: 29 0F	564 565 SETVID 566 OUTPORT 567 OUTPRT 568 568 IOPRT	LDA STA LDX LDY LDA	#\$00 A2L #CSWL #COUT1 A2L	DO 'PR#AREG'
FE91: D0 O8 FE9R FE93: A9 00 FE95: 85 3E FE97: A2 36 FE99: A2 36 FE97: A5 3E FE99: A5 3E FE90: 29 OF FE97: FE97: F0 06	564 565 SETVID 566 OUTPORT 567 OUTPRT 568 569 IOPRT 569	LDA STA LDX LDY LDA AND	#\$00 A2L #CSWL #COUT1 A2L #\$0F	DO 'PR#AREG'
FE91:D0 08 FE98 FE93:A9 00 FE95:B5 3E FE97:A2 36 FE99:A5 3E FE99:A5 3E FE99:A5 GE FE99:A5 3E FE99:A5 GE FE99:A5 3E FE99:A5 GE FE91:A5 GE FE92:A5 GE FE91:A5 GE FE92:A5 GE FE91:A5 GE FEA1:A5 GE FEA3:A0 00 FEA3:A6 GE	564 565 SETVID 566 QUTPORT 567 QUTPORT 569 IOPRT 570 571 572 572 572	LDA STA LDX LDY LDA AND BEQ DRA LDY	#\$00 A2L #CSWL #CQUT1 A2L #\$0F IOPRT1 # <ioadr #\$00</ioadr 	DO 'PR#AREG'
FE91:D0 08 FE98 FE93:A9 00 FE95:B5 32 FE97:A2 36 FE97:A2 36 FE99:A0 F0 FE98:A5 32 FE9D:29 0F FE9F:F0 06 FE97:F0 06 FEA7 FEA1:09 C0	564 565 SETVID 566 OUTPORT 567 OUTPRT 568 IOPRT 570 571 572 572 573 574	LDA STA LDX LDY LDA AND BEQ DRA	#\$00 A2L #CSWL #CQUT1 A2L #\$0F IOPRT1 # <ioadr #\$00 IOPRT2</ioadr 	DO 'PR#AREG'
FE91:100 GB FE98 FE93:47 00 FE98 FE97:82 36 FE97:A2 FE97:A2 36 FE98:A5 FE97:A0 G FE97 FE97:A2 G FE97 FE97:A3 G FE97 FE97:A4 G FE97 FE97:A5 G FE97 FE97:A5 G FE97 FE97:A5 G FE47 FE43:A0 O FE43 FE43:F0 C FE47 FE47:F0 FE47 FE47	544 555 SETVID 564 QUTPORT 564 568 569 IOPRT 570 571 572 573 573 573 575 IOPRT1	LDA STA LDX LDY LDA AND BEQ DRA LDY BEQ LDA	#\$00 A2L #CSWL #CQUT1 A2L #\$0F IOPRT1 # <ioadr #\$00</ioadr 	DO 'PR#AREG'
FE91:D0 0B FE98 FE93:A9 00 FE99 FE97:A2 36 FE99:A0 F0 FE97:A2 36 FE99:A0 F0 FE97:A2 36 FE99:A0 F0 FE97:A2 36 FE99:A0 F0 FE97:A2 36 FE97:F0 06 FE97:F0 06 FEA7 FEA3:F0 02 FEA9 FEA3:F0 02 FEA9 FEA7:A9 FD FEA9	544 545 SETVID 546 OUTPORT 548 548 549 IOPRT 570 571 572 573 573 574 574 IOPRT1 576 IOPRT1 576 IOPRT2	LDA STA LDX LDY LDA AND BEQ DRA LDY BEQ LDA EQU	#\$00 A2L #CSWL #CGUT1 A2L #\$0F IOPRT1 # <ioadr #\$00 IOPRT2 #<cout1 *</cout1 </ioadr 	DO 'PR#AREG'
FE91:100 08 FE98 FE93:47 00 FE98 FE93:85 36 FE97:24 36 FE97:24 36 FE98:25 FE98:26 FE98:25 0F FE99:20 FE99:20 FE99:20 0F FE97:00 FE29:20 FE29:27 F0 06 FEA7 FEA7:36 FEA7 FEA7 FEA7 FEA7:47 400 FEA7 FEA7	565 565 SETVID 566 QUTPORT 567 QUTPRT 568 569 570 571 572 572 572 573 574 575 IOPRT1 576 IOPRT1 576 IOPRT2 577	LDA STA LDX LDY LDA AND BEQ DRA LDY BEQ LDA EQU STY	#\$00 A2L #CSWL #CGUT1 A2L #\$0F I0PRT1 # <i0adr #\$00 I0PRT2 #<ccut1 * LDCO, X</ccut1 </i0adr 	DO 'PR#AREG'
FE91:D0 GB FE98 FE93:A9 00 FE93:B5 3E FE97:A3 3E FE98:A5 3E FE97:F0 06 FE43:P0 FE47 FE41:00 FEA3:F0 FEA3:F0 00 FEA3:F0 2 FEA7:A9 FD FEA7:A9 F0 FEA7 F0 FEA7 F0 FEA7 F0 FEA7 F0 FEA7 F0	564 565 SETVID 566 OUTPORT 567 OUTPRT 567 576 570 572 572 573 574 575 IOPRT1 576 IOPRT1 576 IOPRT2 577	LDA STA LDY LDY LDA AND BEG DRA LDY BEG LDA EGU STY STA	#\$00 A2L #CSWL #CGUT1 A2L #\$0F IOPRT1 # <ioadr #\$00 IOPRT2 #<cout1 *</cout1 </ioadr 	DO 'PR#AREG'
FE91:100.08 FE98 FE93:470.08 FE98 FE93:470.08 FE98 FE97:42.36 FE97 FE97:42.36 FE47 FE43:00 FE43:00 FE43:40.02 FE47 FE47:47 FE47 FE47:47 00 FE43:75.01 FE48 FE48:75.01 FE48	964 565 SETVID 566 OUTPORT 567 OUTPRT 568 569 IOPRT 570 571 572 572 574 575 IOPRT1 575 IOPRT1 576 IOPRT2 577 578 577	LDA STA LDX LDY LDA AND BEQ ORA LDY BEQ LDA EQU STY STA RTS	#\$00 A2L #CSWL #CGUT1 A2L #\$0F I0PRT1 # <i0adr #\$00 I0PRT2 #<ccut1 * LDCO, X</ccut1 </i0adr 	DO 'PR#AREG'
FE91:D0 08 FE98 FE93:A9 00 FE98 FE97:A0 G FE99:A0 F0 FE99:A0 F0 FE99:A0 G FE99:A0 G FE99:A0 G FE90:29 OF FE90:29 OF FE41:09 C0 FEA3:F0 C2 FEA7:A7 FD FEA7:A7 FD FEA7:A7 FEA7 FEA7:50 S0 FEA8:50 S0 FEA8:50 S0 FEA8:50 S0 FEA8:60 FEA7	564 565 SETVID 566 OUTPORT 567 OUTPRT 569 569 IOPRT 571 572 573 574 575 IOPRT1 576 IOPRT1 576 IOPRT2 577 578 578 579 580	LDA STA LDX LDA AND BEG ORA LDA EQU STY STA RTS NOP	#\$00 A2L #CSWL #COUT1 A2L #\$0F I0PRT1 #\$10ADR #\$00 I0PRT2 #\$CCUT1 * L0CC0, X L0C1, X	;DO 'PRHAREG'
FE91:D0 08 FE98 FE93:A7 08 FE98 FE93:A7 08 FE98 FE97:A2 36 FE97:A2 36 FE97:A2 36 FE47 FEA3:F0 02 FEA7 FEA7:A9 FD FEA7 FEA7:A9 7D FEA7 FEA7:A9 00 FEA7 FEA7:5 01 FEA7 FEA8:F0 02 FEA7 FEA8:F00 FEA7 FEA8:F00 FEA7	964 565 SETVID 566 OUTPORT 567 OUTPRT 568 569 570 571 572 573 574 574 574 575 IOPRT1 576 IOPRT2 577 577 578 579 580 CKSUMFIX	LDA STA LDX LDA AND BEG ORA LDA EQU STY STA RTS NOP DFB	#\$00 A2L #CSWL #CCUT1 A2L #\$0F IOPRT1 #\$00 IOPRT2 #\$COUT1 * LOC0, X LOC1, X	; DO 'PRWAREG' ; SET INPUT/DUTPUT VECTORS ; /RRA0981
FE91:D0 08 FE98 FE93:A9 00 FE98 FE97:A0 G FE97:A0 FE FE97:A0 FE FE97:A0 FE FE99:A0 FE FE90:A5 G FEA1:09 C0 FEA3:F0 FEA7 FEA3:A0 00 FEA7:A7 FD FEA7:A7 FEA7 FEA7:A7 FEA7 FEA7:A7 S0 FEA8:F0 FEA7 FEA8:EA FEA7:A7 FEA8:EA FEA7:A7 FEA8:EA FEA7:A7 FEA8:EA FEA7:A7 FEA8:EA FEA7:A7 FEA8:EA FEA7:A7 FEA7:EA FEA7:A7 FEA7:EA FEA7:A7 FEA7:EA FEA7:A7 <	564 565 SETVID 566 OUTPORT 567 OUTPRT 567 OUTPRT 570 571 572 573 574 575 IOPRT1 576 IOPRT1 576 IOPRT1 577 578 578 578 579 580 580 CSUMFIX 582 ★ ;=->20	LDA STA LDY LDA AND BEG CRA LDY BEG LDA EQU STY STA RTS NOP BFB DFB DRRECT	#\$00 A2L #CSWL #CDUT1 A2L #\$0F IOPRT1 #\$00 IOPRT2 #\$00 IOPRT2 #\$CCUT1 * LDC0, X LDC1, X	; DO 'PRHAREG' ; SET INPUT/DUTPUT VECTORS ;/RRAO981 CREATE TIME.
FE91:D0 08 FE98 FE93:A7 08 FE98 FE93:A5 32 FE97:A2 FE97:A2 36 FE47:F0 64 FEA3:F0 02 FEA7:A7 FEA7 FEA7:A7 FE FEA7:A7 FD FEA7:A7 FE FEA7:D0 FE FEB0:AC	564 565 SETVID 566 OUTPORT 567 OUTPRT 568 569 570 571 572 572 573 574 575 IOPRT1 576 IOPRT2 577 576 577 578 579 581 CKSUMFIX 582 ★ i>Ct	LDA STA LDY LDA AND BEG CRA LDY BEG LDA EQU STY STA RTS NOP DFE DRRECT JMP	#\$00 A2L #CSWL #CCUTI A2L #\$0F IOPRT1 #\$00 IOPRT2 # <cout1 * LDC0,X LDC1,X 0 CKSUM AT BASIC</cout1 	; DO 'PRWAREG' ; SET INPUT/DUTPUT VECTORS ; /RRA0981 CREATE TIME. ; TO BASIC, COLD START
FE91:100 08 FE98 FE93:49 00 FE93:49 36 FE97:42 36 FE99:40 F0 FE99:40 F0 FE99:40 F0 FE99:29 0F FE97:40 FE47 FE43:00 FEA7 FEA3:60 02 FEA7:47 F0 FEA7:54 F0 FEA7:57 FEA7 FEA7:50 FEA7 FEA8:50 FEA7 FEA8:50 FEA7 FEA8:50 FEA7 FEA8:60 FEA7 FEA8:60 FEA7 FEA7:00 FEA7 FEB0:40 00 FEB0:40 00	564 565 SETVID 566 OUTPORT 567 OUTPRT 569 570 571 572 573 574 575 IOPRT1 576 IOPRT1 576 IOPRT2 577 578 578 579 580 581 CKSUMFIX 582 * ;>CC 583 XBASIC 584 BASCONT	LDA STA LDY LDA LDA AND BEQ ORA LDY BEQ LDA EQU STY STA RTS STA RTS DFB DRECT JMP	H\$00 A2L #CSWL #CGUT1 A2L #\$0F IOPRT1 # <ioadr #\$00 IOPRT2 #<cgut1 * LDCC,X LDCC,X LDCL,X 0 CKSUM AT BASIC2</cgut1 </ioadr 	; DO 'PRHAREG' ; SET INPUT/DUTPUT VECTORS ; SET INPUT/DUTPUT VECTORS CREATE TIME. ; TO BASIC, COLD START ; TO BASIC, WARM START
FE91:D0 0B FE98 FE93:A9 00 FE98 FE97:A2 36 FE97:A2 36 FE97:A2 36 FE77:F0 06 FEA3:F0 02 FEA7 FEA2:F0 02 FEA7 FEA2:F0 02 FEA7 FEA2:F0 02 FEA7 FEA3:F0 02 FEA7 FEA2:F0 72 FE	564 565 SETVID 566 OUTPORT 567 OUTPRT 568 569 570 571 572 572 573 574 575 IOPRT1 576 IOPRT2 577 576 577 578 579 581 CKSUMFIX 582 ★ i>Ct	LDA STA LDY LDA AND BEG CRA LDY BEG LDA EQU STY STA RTS NOP DFE DRRECT JMP	#\$00 A2L #CSWL #CSUL #CSUT1 A2L #\$0F IOPRT1 #\$10ADR #\$00 IOPRT2 #CCOUT1 * LDC0.X LDC1.X 0 CKSUM AT BASIC2 BASIC2 AIPC	;DO 'PRWAREG' ;SET INPUT/DUTPUT VECTORS ;SET INPUT/DUTPUT VECTORS CREATE TIME. ;TO BASIC, COLD START ;TO BASIC, WARM START ;ADD TO PC IF SPECIFIED
FE91:100 08 FE98 FE93:40 00 FE93:45 32 FE97:42 36 FE99:40 F0 FE99:29 07 FE99:29 06 FE41:09 00 FEA3:F0 02 FEA3:F0 02 FEA3:50 02 FEA3:50 02 FEA3:50 501 FEA3:50 FEA7 FEA3:50 FEA7 FEA3:60 02 FEA3:60 02 FEA3:60 501 FEA5:60 FEA7 FEA5:60 FEA7 FEA7:60 FEA7 FEA7:60 FEA7 FEA7:60 FEA7 FEA7:60 FEA7 FEB0:40 70 FEB0:40 20 FEB3:40 20 FEB3:40 20 FEB3:20 75 FEB7:20 75	564 565 SETVID 566 OUTPORT 567 OUTPRT 567 570 571 572 573 574 575 IOPRT1 576 IOPRT2 577 576 577 577 577 581 CKSUMFIX 582 ★ ;>CC 583 XBASIC 584 BASCONT 585 GO	LDA STA LDX LDY LDA AND BEG DRA LDA EGU LDA EGU STY STA RTS STA RTS DRECT JMP JSR	H\$00 A2L #CSWL #CGUT1 A2L #\$0F IOPRT1 # <ioadr #\$00 IOPRT2 #<cgut1 * LDCC,X LDCC,X LDCL,X 0 CKSUM AT BASIC2</cgut1 </ioadr 	; DO 'PRHAREG' ; SET INPUT/OUTPUT VECTORS ; SET INPUT/OUTPUT VECTORS CREATE TIME. ; TO BASIC, COLD START ; TO BASIC, WARM START ; ADDR TO PC IF SPECIFIED ; RESTORE FAKE REGISTERS
FE91:D0 08 FE98 FE93:A9 00 FE98 FE97:A2 36 FE99:A9 FE97:A2 36 FE99:A9 FE97:A2 36 FE99:A9 FE97:A3 36 FE97:A9 FE97:A9 06 FE97:A9 FE37:A9 FEA7 FEA3:F0 02 FEA7 FEA3:F0 02 FEA7 FEA7:A9 F0 FEA7:A9 F0 FEA9:A00 FEA7 FEA9:50 FEA7 FEA9:50 FEA7 FEA9:60 FEA7 FEA9:70 FEA7 FEA9:00 FEA7 FEA9:70 FEA7 FEA9:70 FEA7 FEA9:70 FEA7 FEA9:00 FEA7 FEA9:00 FEA7 FEB0:40 G0 FEB0:40 G0 FEB0:40 G0 FEB0:40 G0 FEB0:40 G0 FEB0:20 G7 FEB0:20 G7 FEB0:20	564 565 SETVID 566 OUTPORT 567 OUTPRT 567 570 571 572 573 574 575 IOPRT1 576 IOPRT2 577 576 577 577 581 CKSUMFIX 582 ★ ,>CC 583 XBASIC 584 BASCONT 585 GO 586 585	LDA STA LDY LDA AND BEG DRA LDA EGU LDA EGU STA RTS STA RTS JRRECT JMP JSR JSR JMP	#500 A2L #CSWL #CSWL #CSWL #CSWL #CSWL #CSWL #CSWL #COTI #S00 10PRT2 #S00 10PRT2 #S00 10PRT2 #CCWT1 * LOC0.X LOC1.X 0 CKSUM AT BASIC2 AJPC RESTORE (PCL)	;DO 'PRWAREC' ;SET INPUT/DUTPUT VECTORS ;SET INPUT/DUTPUT VECTORS ;CT DASIC, COLD START ;TO BASIC, WARM START ;ADD TO PC IF SPECIFIED ;RESTORE FAKE REGISTERS ;AND 60!
FE91:D0 08 FE98 FE93:A9 00 FE98 FE97:A2 36 FE99:A9 FE97:A2 36 FE99:A9 FE97:A2 36 FE99:A9 FE97:A3 36 FE97:A9 FE97:A9 06 FE97:A9 FE37:A9 FEA7 FEA3:F0 02 FEA7 FEA3:F0 02 FEA7 FEA7:A9 F0 FEA7:A9 F0 FEA9:A00 FEA7 FEA9:50 FEA7 FEA9:50 FEA7 FEA9:60 FEA7 FEA9:70 FEA7 FEA9:00 FEA7 FEA9:70 FEA7 FEA9:70 FEA7 FEA9:70 FEA7 FEA9:00 FEA7 FEA9:00 FEA7 FEB0:40 G0 FEB0:40 G0 FEB0:40 G0 FEB0:40 G0 FEB0:40 G0 FEB0:20 G7 FEB0:20 G7 FEB0:20	564 565 SETVID 566 OUTPORT 567 OUTPRT 567 OUTPRT 576 577 577 577 574 575 IOPRT1 576 IOPRT2 577 578 578 579 580 583 XBASIC 583 XBASIC 584 BASCONT 585 GO 586 586 587 588 REGZ 589 TRACE	LDA STA LDX LDY LDA AND BEG DRA LDA EGU LDA EGU STY STA RTS STA RTS DRECT JMP JSR		; DO 'PRHAREG' ; SET INPUT/OUTPUT VECTORS ; SET INPUT/OUTPUT VECTORS CREATE TIME. ; TO BASIC, COLD START ; TO BASIC, WARM START ; ADDR TO PC IF SPECIFIED ; RESTORE FAKE REGISTERS
FE91:D0 08 FE98 FE93:A9 00 FE98 FE97:A0 G FE97:A0 F0 FE97:A0 F0 FE99:A0 F0 FE90:A5 G FE90:A5 G FE97:F0 OA FEA1:09 C0 FEA3:F0 C FEA3:F0 C FEA7:A7 FD FEA7:A7 FEA7 FEA3:F0 C FEA3:F0 C FEA5:F0 C FEA5:F0 FEA7 FEA8:F0 FEA7 FEA8:F0 FEA7 FEB0:A0 C0 FEB0:A0 C0 FEB0:A0 C0 FEB0:A0 C0 FEB0:A0 C0 FEB0:A0 SFF FEB0:A0 SF FEB0:A0 SF FEB0:A0 SF FEB0:A0 SF FEB0:A0 SF FEB0:A0 SF<	964 965 SETVID 967 OUTPRT 967 OUTPRT 968 969 IOPRT 971 972 973 973 974 975 IOPRT1 975 IOPRT2 977 977 978 977 980 977 980 981 CKSUMFIX 982 ★ ;>CC 984 BASCONT 985 GO 985 985 986 REGZ 590	LDA STA LDX LDY LDY LDY BEG ORA BEG DRA BEG LDA STY STA STY STA STY STY STY STY NOP BECT JMP JSR JSR JSR JMP	#500 A2L #CSWL #CSWL #CSWL #CSWL #CSWL #CSWL #CSWL #COTI #S00 10PRT2 #S00 10PRT2 #S00 10PRT2 #CCWT1 * LOC0.X LOC1.X 0 CKSUM AT BASIC2 AJPC RESTORE (PCL)	; DO 'PRHAREG' ; SET INPUT/OUTPUT VECTORS ; SET INPUT/OUTPUT VECTORS CREATE TIME. ; TO BASIC, COLD START ; TO BASIC, WARM START ; ADDR TO PC IF SPECIFIED ; RESTORE FAKE REGISTERS ; AND GD! ; GO DISPLAY REGISTERS ; TRACE IS GONE
FE91:D0 08 FE98 FE93:A9 00 FE98 FE97:A0 06 FE98 FE97:A0 70 FE97 FE97:A0 70 FE97 FE97:A0 70 FE79 FE97:A0 70 FE77 FE77:F0 06 FEA7 FEA3:F0 02 FEA7 FEA3:F0 02 FEA7 FEA7:A9 FD FEA7:FA 00 FEA7:F4 00 FEA7 FEA7:F0 03 E0 FEA7 FEA8:F0 03 E0 FEB0:40 30 E0 FEB0:40 07 5 FE FEB0:40 30 E0 FEB0:40 17 FA FEB0:40 10 FA FEB0:40 17 FA FEB0:40 10 FA FEB0:40 10 FA FE FE00:40 10 FA FE FE00:40 10 FA FE FE00:40 10 FA FE FE	564 565 SETVID 566 OUTPORT 567 OUTPRT 567 578 570 571 574 575 IOPRT1 576 IOPRT2 577 577 IOPRT2 578 578 579 580 580 CKSUMFIX 582 * ;->CC 583 XBASIC 584 BASCONT 585 GO 586 587 588 REGZ 589 TRACE 590 SEPZ	LDA STA LDY LDY AND BEQ DRA EQU STA EQU STA EQU STA STA STA STA JMP JSR MP JSR RTS NDP RTS RTS	#500 A2L #CSWL #CSWL #CSWL #CSWL #COUTI #CIOADR #500 10PRT2 #500 10PRT2 #500 10PRT2 #500 10PRT2 #500 10PRT2 #500 0 CKSUM AT BASIC2 A1PC RESTORE (PCL) REGDSP	; DO 'PRHAREG' ; SET INPUT/OUTPUT VECTORS ; SET INPUT/OUTPUT VECTORS CREATE TIME. ; TO BASIC, COLD START ; TO BASIC, WARM START ; ADDR TO PC IF SPECIFIED ; RESTORE FAKE REGISTERS ; AND GD! ; GO DISPLAY REGISTERS
FE91:D0 08 FE98 FE93:A7 08 FE98 FE93:A5 32 FE97:A2 36 FE97:A2 36 FE77:F0 06 FEA3:F0 02 FEA7 FEA3:F0 02 FEA7 FEA3:F0 02 FEA7 FEA3:F0 03 FEA7 FEA3:F0 04 FEA7 FEA5:F0 05 FEA7 FEB0:A2 00 FEA7 FEB0:A2 00 FEA7 FEB0:A2 00 FEB3:A2 00 FEB0:A2 00 FEB3:A2 00 FEB3:A2 03 FE FEB3:20 37 FE	564 565 SETVID 567 OUTPRT 567 OUTPRT 568 569 570 571 572 572 573 574 575 IOPRT1 576 IOPRT2 577 576 577 578 581 CKSUMFIX 582 ★ ,>CC 583 XBASIC 584 BASCONT 585 GO 586 885 GO 586 587 588 REGZ 589 TRACE 590 591 STEPZ 592	LDA STA LDY LDY LDY LDA EQO STA STA STA STA STA STA STA STA STA STA	#500 A2L #CSWL #CCUTI UDPRT1 #CCUTI #10ADR #500 IDPRT2 #CCUTI %CCUTI %CCUTI LCC0.X LCC1.X 0 CKSUM AT BASIC2 AIPC (PCL) RESTORE (PCL) RESTORE (PCL)	; DO 'PRHAREG' ; SET INPUT/OUTPUT VECTORS ; SET INPUT/OUTPUT VECTORS CREATE TIME. ; TO BASIC, COLD START ; TO BASIC, WARM START ; ADDR TO PC IF SPECIFIED ; RESTORE FAKE REGISTERS ; AND GD! ; GO DISPLAY REGISTERS ; TRACE IS GONE ; STEP IS GONE
FE91:D0 0B FE98 FE93:A9 00 FE98 FE97:A2 36 FE97:A2 36 FE97:A2 36 FEA7 FEA3:F0 02 FEA7 FEA3:F0 02 FEA7 FEA7:A7 40 FEA7 FEA7:A7 400 FEA7 FEA7:A7 50 FEA7 FEA7:A7 50 FEA7 FEA7:A7 400 FEA7 FEA7:A7 50 FE7 FEA7:A7 50 FE7 FEB3:40 03 E0 FE7 FEB0:20 37 FE FE7 FEB0:20 34 00 FE7 FE22:A20 34 FE7 FE22:A20 77 FA FE7 <td>564 565 SETVID 566 OUTPORT 567 OUTPRT 570 571 572 573 574 575 IOPRT1 576 IOPRT2 577 576 IOPRT2 577 578 579 580 580 581 CKSUMFIX 582 * ;->CC 583 XBASIC 584 BASCONT 585 GO 585 587 TRACE 590 589 TRACE 590 SR</td> <td>LDA STA LDA LDA LDA LDA LDA BEG DRA LDY BEG LDA EGU STA STA STA STA JMP RTS JSR RTS JMP RTS NDP RTS JMP RTS JMP RTS JMP</td> <td>*500 A2L *CSWL *CSWL *CSWL *CSWL *CSWL *COUTI *10PRT1 *500 10PRT2 *COUTI *COUTI *COUTI *COUTI *COUTI *COUTI *COUTI *COUTI *COUTI *COUTI *COUTI *COUTI *COUTI *COUTI *COUTI *STOR *COUTI *STOR *COUTI *STOR *STOR *STOR *STOR *STOR *COUTI *STOR</td> <td>; DO 'PRWAREG' ; SET INPUT/OUTPUT VECTORS ; SET INPUT/OUTPUT VECTORS ; TO BASIC, COLD START ; TO BASIC, WARM START ; ADDR TO PC IF SPECIFIED ; RESTORE FAKE REGISTERS ; AND GO! ; GO DISPLAY REGISTERS ; TRACE IS GONE ; STEP IS GONE ; JUMP TO CONTROL-Y VECTOR IN RAM</td>	564 565 SETVID 566 OUTPORT 567 OUTPRT 570 571 572 573 574 575 IOPRT1 576 IOPRT2 577 576 IOPRT2 577 578 579 580 580 581 CKSUMFIX 582 * ;->CC 583 XBASIC 584 BASCONT 585 GO 585 587 TRACE 590 589 TRACE 590 SR	LDA STA LDA LDA LDA LDA LDA BEG DRA LDY BEG LDA EGU STA STA STA STA JMP RTS JSR RTS JMP RTS NDP RTS JMP RTS JMP RTS JMP	*500 A2L *CSWL *CSWL *CSWL *CSWL *CSWL *COUTI *10PRT1 *500 10PRT2 *COUTI *COUTI *COUTI *COUTI *COUTI *COUTI *COUTI *COUTI *COUTI *COUTI *COUTI *COUTI *COUTI *COUTI *COUTI *STOR *COUTI *STOR *COUTI *STOR *STOR *STOR *STOR *STOR *COUTI *STOR	; DO 'PRWAREG' ; SET INPUT/OUTPUT VECTORS ; SET INPUT/OUTPUT VECTORS ; TO BASIC, COLD START ; TO BASIC, WARM START ; ADDR TO PC IF SPECIFIED ; RESTORE FAKE REGISTERS ; AND GO! ; GO DISPLAY REGISTERS ; TRACE IS GONE ; STEP IS GONE ; JUMP TO CONTROL-Y VECTOR IN RAM
FE91:D0 08 FE98 FE93:A7 08 FE98 FE93:A5 32 FE97:A2 36 FE97:A2 36 FE97:A2 36 FE77:A2 37 FE97 FEA3:F0 02 FEA7 FEA3:F0 02 FEA7 FEA3:F0 02 FEA7 FEA3:F0 03 FEA7 FEA5:F0 04 FEA7 FEA5:F0 05 FEA7 FEB0:AC 00 E0 FEB0:AC 03 E0 FEB0:AC 03 E0 FEB0:AC 07 FA FEB1:AC 07 FA FEC3:AC 07 FA FEC3:C4 FF FE35:C2 F2 F9 FEC3:C2 F2 FEC3:C2 F2 FEC3:C2 F2 FEC3:C2 F2 FE35:C2	564 565 SETVID 567 OUTPRT 567 OUTPRT 568 569 570 571 572 573 573 574 575 IOPRT1 576 IOPRT2 577 576 577 578 577 578 579 581 CKSUMFIX 582 ★ ;>CC 583 XBASIC 584 BASCONT 585 O 588 RECZ 598 RECZ 599 TRACE 599 591 USR 592 593 USR 594 WRITE	LDA STA LDY LDY LDY LDY LDY AND BEG CRA LDY LDA EQU LDA EQU LDA EQU STA STA STA STA STA JMP JSR JSR JSR JSR ASC JMP ASC JMA	#500 A2L #CSWL #CSUT1 UDPRT1 #CSUT1 #CIOADR #500 IDPRT2 #CCUT1 % LDCC.X LDCC.X 0 CKSUM AT BASIC AIPC AIPC MASIC2 AIPC (FCL) RESTORE (FCL) REST	; DO 'PRWAREG' ; SET INPUT/OUTPUT VECTORS ; SET INPUT/OUTPUT VECTORS ; TO BASIC. COLD START ; TO BASIC. COLD START ; TO BASIC. WARM START ; ADD TO PC IF SPECIFIED ; RESTORE FAKE REGISTERS ; AND GO! ; GO DISPLAY REGISTERS ; TRACE IS GONE ; STEP IS GONE ; JUMP TO CONTROL-Y VECTOR IN RAM ; JUMP TO CONTROL-Y VECTOR IN RAM
FE91:D0 08 FE98 FE93:A9 00 FE98 FE97:A3 36 FE97:A3 36 FE97:A3 36 FE98:A5 38 FE97:A2 36 FE97:A2 36 FE97:A3 36 FE97:A2 36 FE97:A5 38 FE97:A2 70 FE77:F0 06 FEA7 FEA3:F0 02 FEA7 FEA7:A7 FD FEA7 FEA7:A9 CO FEA7 FEA7 FEA7 FEA7 S0 FE FEA7 S0 FE FEB0:CO D7 FA FEC2:CO <td>564 565 SETVID 566 OUTPORT 567 OUTPRT 570 571 572 573 574 575 IOPRT1 576 IOPRT1 576 IOPRT1 577 578 579 580 580 CKSUMFIX 582 * ;>CC 883 XBASIC 583 XBASIC 584 BASCONT 584 BASCONT 585 REGZ 586 REGZ 596 ST 597 STEPZ 593 USR 595</td> <td>LDA STAX LDY LDY LDY AND BERA LDA LDA EGU STYA STSP JMP RTSP SREMP JSR RTSP JMP RTSP SREMP JSR LDA LDA JSR LDA</td> <td>*\$00 A2L #CSWL #CSWL #CSWL #COUTI #CUTI #CUTI *\$00 10PRT1 #\$00 10PRT2 #CUTI *C</td> <td>; DO 'PRWAREG' ; SET INPUT/OUTPUT VECTORS ; SET INPUT/OUTPUT VECTORS ; TO BASIC, COLD START ; TO BASIC, WARM START ; ADDR TO PC IF SPECIFIED ; RESTORE FAKE REGISTERS ; AND GO! ; GO DISPLAY REGISTERS ; TRACE IS GONE ; STEP IS GONE ; JUMP TO CONTROL-Y VECTOR IN RAM</td>	564 565 SETVID 566 OUTPORT 567 OUTPRT 570 571 572 573 574 575 IOPRT1 576 IOPRT1 576 IOPRT1 577 578 579 580 580 CKSUMFIX 582 * ;>CC 883 XBASIC 583 XBASIC 584 BASCONT 584 BASCONT 585 REGZ 586 REGZ 596 ST 597 STEPZ 593 USR 595	LDA STAX LDY LDY LDY AND BERA LDA LDA EGU STYA STSP JMP RTSP SREMP JSR RTSP JMP RTSP SREMP JSR LDA LDA JSR LDA	*\$00 A2L #CSWL #CSWL #CSWL #COUTI #CUTI #CUTI *\$00 10PRT1 #\$00 10PRT2 #CUTI *C	; DO 'PRWAREG' ; SET INPUT/OUTPUT VECTORS ; SET INPUT/OUTPUT VECTORS ; TO BASIC, COLD START ; TO BASIC, WARM START ; ADDR TO PC IF SPECIFIED ; RESTORE FAKE REGISTERS ; AND GO! ; GO DISPLAY REGISTERS ; TRACE IS GONE ; STEP IS GONE ; JUMP TO CONTROL-Y VECTOR IN RAM
FE91:D0 08 FE98 FE93:A9 00 FE98 FE93:A9 32 FE97:A2 36 FE97:A2 36 FE77:A2 36 FE77:A2 36 FEA7 FEA3:F0 02 FEA7 FEB0:A0 00 FEA7 FEB0:A0 00 FEA7 FEB0:A0 00 E0 FEB0:A0 00 E0 FEB0:A0 00 E0 FEB0:A0 07 FA	564 565 SETVID 566 OUTPORT 567 OUTPRT 567 OUTPRT 570 571 572 573 573 574 575 IOPRT1 576 IOPRT2 577 578 579 580 581 CKSUMFIX 582 * ,>CC 583 XBASIC 584 BASCONT 585 QB 586 586 587 588 FECZ 589 587 589 587 589 587 589 587 589 587 589 587 589 587 589 587 589 587 589 587 589 587 589 587 589 587 589 587 589 587 589 587 589 587 589 587 589 580 587 580 587 587 587 587 587 587 587 587	LDA STA LDX LDX LDA AND BEQ DRA LDY STA STA STA STA STA STA STA JSR RECP JSR RECP JSR RECP JSR RCP JSR RCP JSR RCP JSR LDA LDY LDA STA LDY LDA STA LDY LDY STA LDY LDY LDY LDY LDY LDY LDY LDY LDY LDY	#500 A2L #CSWL #CCUTI #CCUTI #CCUTI #CCUTI #CCUTI #CCUTI *	; DO 'PRWAREG' ; SET INPUT/OUTPUT VECTORS ; SET INPUT/OUTPUT VECTORS ; TO BASIC. COLD START ; TO BASIC. COLD START ; TO BASIC. WARM START ; ADD TO PC IF SPECIFIED ; RESTORE FAKE REGISTERS ; AND GO! ; GO DISPLAY REGISTERS ; TRACE IS GONE ; STEP IS GONE ; JUMP TO CONTROL-Y VECTOR IN RAM ; JUMP TO CONTROL-Y VECTOR IN RAM
FE91:D0 08 FE98 FE93:A9 00 FE98 FE97:A3 36 FE97:A3 36 FE97:A3 36 FE97:A3 36 FE97:A2 36 FE97:A3 36 FE97:A3 36 FE97:A3 70 FE97:A3 70 OF FE97:A7 87 D FEA3:F0 02 FEA7 FEA7:A7 FD FEA7:A9 FD FEA7:A9 F0 FEA7:A9 F0 FEA7:A9 F0 FEA7:A9 F0 FEA7:A9 F0 FEA7:A9 S0 FEA8:A0 FEA7 FEA7:A9 S0 FEB0:A0 S7 FEB0:A0 S7 FEB1:A0 S7	564 565 SETVID 566 OUTPORT 567 OUTPRT 570 571 572 573 574 575 IOPRT1 576 IOPRT1 576 IOPRT2 577 577 578 580 580 581 CSUMFIX 582 * ;>CC 583 XBASIC 583 XBASIC 584 BASCONT 585 60 586 587 TRACE 590 588 RECZ 593 USR 595 USR 595 595 595 WR1	LDA STA LDY LDY LDY LDA AND DRA LDY BEG DRA LDY STA STA STA JOP BCG STA AND DFB UDA JOP RTS JOP RTS JOP RTS JOP RTS JOP LDA ASC JOP LDA ASC JOP LDA LDY LDY LDA AND DFB US RECO LDY LDA AND DFB US STA LDY LDY LDA AND DFB US STA LDY LDY LDA AND DFB G US STA LDY LDY LDA AND DFB G US STA LDY LDY LDY LDA AND DFB G US STA LDY LDY LDY LDY LDA AND DFB G US STA LDY LDY LDY LDY LDA AND DFB G US STA LDY LDY LDA AND DFB G US STA LDY LDY LDA AND DFB CO LDA STA LDY LDY LDA AND DFB CO LDA STA LDY LDY LDA STA LDY LDY LDA STA LDY LDA STA LDY LDA STA LDY LDA LDA STA LDY LDA STA LDY LDA STA LDY LDA STA LDY LDA STA LDY LDA STA LDY LDA STA LDY LDA STA LDY LDA STA LDY LDA LDA STA LDY LDA LDA LDA STA LDY LDA LDA LDA STA LDY LDA LDA LDA LDA STA LDY LDA LDA LDA LDA LDA LDA LDA LDA LDA LDA	*\$00 ACL #COUTI #COUTI #COUTI #COUTI 10PRT1 *100RT1 *COUTI	; DO 'PRWAREG' ; SET INPUT/OUTPUT VECTORS ; SET INPUT/OUTPUT VECTORS ; TO BASIC. COLD START ; TO BASIC. COLD START ; TO BASIC. WARM START ; ADD TO PC IF SPECIFIED ; RESTORE FAKE REGISTERS ; AND GO! ; GO DISPLAY REGISTERS ; TRACE IS GONE ; STEP IS GONE ; JUMP TO CONTROL-Y VECTOR IN RAM ; JUMP TO CONTROL-Y VECTOR IN RAM
FE91:D0 08 FE98 FE93:A7 08 FE98 FE97:A2 36 FE77:P:0 6 FEA3:F0 02 FEA3:F0 02 FEA3:F0 02 FEA3:F0 02 FEA3:F0 02 FEA3:F0 03 FEA3:F0 03 FEA3:F0 04 FEA3:F0 03 FEA3:F0 75 FEB0:AC 03 FEB0:AC 03 FEB0:AC 07 FEC2:A3 00 FEC2:A0 7 <td>564 565 SETVID 566 OUTPORT 567 OUTPRT 567 OUTPRT 570 571 572 573 574 575 IOPRT1 576 IOPRT2 577 576 580 581 CKSUMFIX 582 * ;>CC 583 XBASIC 584 BASCONT 585 GO 586 586 587 588 RECZ 589 589 TRACE 590 591 STEPZ 592 593 USR 594 WRITE 595 596 597 WR1 598</td> <td>LDA STA ALDX LLDY LLDA AND DRA BEG DRA AND DFB STA ASC STA ASC JAMP JSR RTS JSR ASC JMP JSR TJ JSR TJ JSR TJ JSR CT JSR STA ASC JSTA JSTA JSTA ASC JSTA JSTA JSTA JSTA JSTA JSTA JSTA JSTA</td> <td>#500 A2L #CSWL #CCUTI #CCUTI #CCUTI #CCUTI #CCUTI #CCUTI *</td> <td>; DO 'PRWAREG' ; SET INPUT/OUTPUT VECTORS ; SET INPUT/OUTPUT VECTORS ; TO BASIC. COLD START ; TO BASIC. COLD START ; TO BASIC. WARM START ; ADD TO PC IF SPECIFIED ; RESTORE FAKE REGISTERS ; AND GO! ; GO DISPLAY REGISTERS ; TRACE IS GONE ; STEP IS GONE ; JUMP TO CONTROL-Y VECTOR IN RAM ; JUMP TO CONTROL-Y VECTOR IN RAM</td>	564 565 SETVID 566 OUTPORT 567 OUTPRT 567 OUTPRT 570 571 572 573 574 575 IOPRT1 576 IOPRT2 577 576 580 581 CKSUMFIX 582 * ;>CC 583 XBASIC 584 BASCONT 585 GO 586 586 587 588 RECZ 589 589 TRACE 590 591 STEPZ 592 593 USR 594 WRITE 595 596 597 WR1 598	LDA STA ALDX LLDY LLDA AND DRA BEG DRA AND DFB STA ASC STA ASC JAMP JSR RTS JSR ASC JMP JSR TJ JSR TJ JSR TJ JSR CT JSR STA ASC JSTA JSTA JSTA ASC JSTA JSTA JSTA JSTA JSTA JSTA JSTA JSTA	#500 A2L #CSWL #CCUTI #CCUTI #CCUTI #CCUTI #CCUTI #CCUTI *	; DO 'PRWAREG' ; SET INPUT/OUTPUT VECTORS ; SET INPUT/OUTPUT VECTORS ; TO BASIC. COLD START ; TO BASIC. COLD START ; TO BASIC. WARM START ; ADD TO PC IF SPECIFIED ; RESTORE FAKE REGISTERS ; AND GO! ; GO DISPLAY REGISTERS ; TRACE IS GONE ; STEP IS GONE ; JUMP TO CONTROL-Y VECTOR IN RAM ; JUMP TO CONTROL-Y VECTOR IN RAM
FE91:D0 08 FE98 FE93:A9 00 FE98 FE97:A2 36 FE97:A3 36 FE97:A2 36 FE97:A2 36 FE97:A2 36 FE97:A2 36 FE98:A5 38 FE97:A2 36 FE97:A2 00 FE47:F0 06 FEA3:F0 02 FEA7 FEA3:F0 02 FEA7 FEA7:A9 FD FEA7 FEA7 FEA7 FEA7 FEA7 FEA7 FEA7 FEB3:40 20 FED4:42 00 FED4	564 565 SETVID 566 OUTPORT 567 OUTPRT 570 571 572 573 574 575 IOPRT1 576 IOPRT1 576 IOPRT1 577 OPT2 577 580 580 CKSUMFIX 582 * ;>CC 583 XBASIC 582 * ;>CC 583 XBASIC 583 XBASIC 584 BASCONT 585 GO 585 RECZ 589 TRACE 590 586 RECZ 595 SYF URI 595 SYF URI 595 SYF URI 595 SYF URI 595 SYF URI	LDA STA LDY LDY LDA AND DRA STY STA STY STA STY STA STY STA STA STY STA STA STA STA STA STA STA STA STA STA	*\$00 A2L *COUT1 *COUT1 *COUT1 *COUT1 *10PRT1 *COUT1	; DO 'PRWAREG' ; SET INPUT/OUTPUT VECTORS ; SET INPUT/OUTPUT VECTORS ; TO BASIC. COLD START ; TO BASIC. COLD START ; TO BASIC. WARM START ; ADD TO PC IF SPECIFIED ; RESTORE FAKE REGISTERS ; AND GO! ; GO DISPLAY REGISTERS ; TRACE IS GONE ; STEP IS GONE ; JUMP TO CONTROL-Y VECTOR IN RAM ; JUMP TO CONTROL-Y VECTOR IN RAM
FE91:D0 08 FE98 FE93:A9 00 FE98 FE97:A2 36 FE97:A2 36 FE97:A2 06 FE97:A2 36 FE77:A2 70 FEA7:F0 FEA3:F0 02 FEA7 FEA3:F0 07 FEA7 FEB0:AC 00 E0 FEB0:AC 03 E0 FEB0:AC 03 E0 FEB0:AC 03 FF FEB0:AC 07 FA FEB1:AC 07 FA FEC2:AC 78 F0 FEC2:AC 07 FC F	564 565 SETVID 566 OUTPORT 567 OUTPRT 567 570 571 572 573 574 575 IOPRT1 576 IOPRT2 577 576 IOPRT2 577 580 581 CKSUMFIX 582 * ;>CC 583 XBASIC 584 BASCONT 585 GO 586 587 588 REGZ 589 TRACE 590 589 TRACE 591 STEPZ 595 594 WRITE 595 596 597 WRI 598 599 599 599	LDA STA ADDY LDY LDY ADD DRA STY DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	#500 A2L #CSWL #CCUTI #CCUTI #CCUTI #CCUTI #CCUADR #800 IDPRT2 #CCUTI * LDCC, X LDCI, X 0 CKSUM AT BASIC2 AJPC RESDSP "BYJAN" USRADR #840 HEADR #840 HEADR #840 (A1L, X) (A1L, X)	; DO 'PRWAREG' ; SET INPUT/OUTPUT VECTORS ; SET INPUT/OUTPUT VECTORS ; TO BASIC. COLD START ; TO BASIC. COLD START ; TO BASIC. WARM START ; ADD TO PC IF SPECIFIED ; RESTORE FAKE REGISTERS ; AND GO! ; GO DISPLAY REGISTERS ; TRACE IS GONE ; STEP IS GONE ; JUMP TO CONTROL-Y VECTOR IN RAM ; JUMP TO CONTROL-Y VECTOR IN RAM
FE91:D0 08 FE98 FE93:A9 00 FE98 FE93:A9 00 FE98 FE97:A3 36 FE99:A5 36 FE99:A5 36 FE97:A3 36 FE99:A5 36 FE97:A5 36 FE97:A5 36 FE97:A6 00 FE41:09 C0 FEA7 FEA3:F0 02 FEA7 FEA7:A9 FD FEA7 FEB7:A0 20 FF FEA7 FED4:A2 00 FED5:A1 30	564 565 SETVID 566 OUTPORT 567 OUTPRT 567 571 572 573 574 575 IOPRT1 576 IOPRT1 577 576 IOPRT1 577 580 581 CKSUMFIX 582 * ;>CC 583 XBASIC 582 * ;>CC 583 XBASIC 583 XBASIC 584 BASCONT 584 BASCONT 585 GO 586 587 586 587 588 RECZ 589 588 589 589 589 593 USR 595 595 595 597 WR1 599 599 600	LDA STA STA LDY LDY LDA AND DRA BEG DRA EGU DRA EGU DRA EGU STY STA STA STA STA STA STA STA STA STA STA	*\$00 ACL *COUTI	; DO 'PRWAREG' ; SET INPUT/OUTPUT VECTORS ; SET INPUT/OUTPUT VECTORS ; TO BASIC. COLD START ; TO BASIC. COLD START ; TO BASIC. WARM START ; ADD TO PC IF SPECIFIED ; RESTORE FAKE REGISTERS ; AND GO! ; GO DISPLAY REGISTERS ; TRACE IS GONE ; STEP IS GONE ; JUMP TO CONTROL-Y VECTOR IN RAM ; JUMP TO CONTROL-Y VECTOR IN RAM
FE91:D0 08 FE98 FE93:A9 00 FE98 FE97:A2 36 FE97 FE97:A2 36 FE97 FE97:A3 36 FE97 FE97:A9 0F FE97 FE77:A9 0F FE77 FE37:A0 00 FE37:C0 FE37:A0 00 FE37:C0 FE37:A9 FD FE37:C0 FE37:A0 00 FE37:C0 FE37:A9 C0 FE37:C0 FE32:A7 00 FE37:C0 FE32:A7 00 FE37:C0 FE32:A7 00 FE37:C0 FE33:A1 00 E0 FE33:A1 00 E0 FE32:A1 00 E0 FE43:A1 00 E0 FE44:A2 00 FE FE03:A	564 565 SETVID 566 OUTPORT 567 OUTPRT 567 OUTPRT 570 571 572 573 574 575 IOPRT1 576 IOPRT2 577 576 IOPRT2 577 580 581 CKSUMFIX 582 × ;>CC 583 XBASIC 583 XBASIC 584 BASCONT 585 GO 586 587 588 REGZ 589 TRACE 590 589 TRACE 591 STEPZ 593 594 WRITE 595 596 597 WR1 597 598 597 SC 598 597 SC 598 597 SC 598 597 SC 599 CS 597 SC 597 SC 598 SC 597 SC 597 SC 598 SC 597 SC 597 SC 597 SC 598 SC 597 SC 597 SC 597 SC 597 SC 598 SC 597 SC 597 SC 597 SC 598 SC 597 SC 598 SC 597 SC 597 SC 598 SC 599 SC 598 SC 599 SC 590 SC	LDA TA STA STA STA STA STA STA STA STA STA	#500 A2L #CSWL #CCUT1 #CUT1 #CICADR #500 IDPRT1 #CICADR #500 IDPRT2 #CCUT1 * LDCC0.X LDCC1.X 0 CKSUM AT BASIC2 AJPC RESDR (PCL) REGDSP /BTygan/ USRADR #520 #540 (A1L,X) WRBYTE NXTA1	; DO 'PRWAREG' ; SET INPUT/OUTPUT VECTORS ; SET INPUT/OUTPUT VECTORS ; TO BASIC. COLD START ; TO BASIC. COLD START ; TO BASIC. WARM START ; ADD TO PC IF SPECIFIED ; RESTORE FAKE REGISTERS ; AND GO! ; GO DISPLAY REGISTERS ; TRACE IS GONE ; STEP IS GONE ; JUMP TO CONTROL-Y VECTOR IN RAM ; JUMP TO CONTROL-Y VECTOR IN RAM
FE91:D0 08 FE98 FE93:A9 00 FE98 FE93:A9 00 FE98 FE97:A3 36 FE99:A5 36 FE99:A5 36 FE97:A3 36 FE99:A5 36 FE97:A5 36 FE97:A5 36 FE97:A6 00 FE41:09 C0 FEA7 FEA3:F0 02 FEA7 FEA7:A9 FD FEA7 FEB7:A0 20 FF FEA7 FED4:A2 00 FED5:A1 30	564 565 SETVID 566 OUTPORT 567 OUTPRT 567 571 572 573 574 575 IOPRT1 576 IOPRT1 577 576 IOPRT1 577 580 581 CKSUMFIX 582 * ;>CC 583 XBASIC 582 * ;>CC 583 XBASIC 583 XBASIC 584 BASCONT 584 BASCONT 585 GO 586 587 586 587 588 RECZ 589 588 589 589 589 593 USR 595 595 595 597 WR1 599 599 600	LDA STA STA LDY LDY LDA AND DRA BEG DRA EGU DRA EGU DRA EGU STY STA STA STA STA STA STA STA STA STA STA	*\$00 ACL *COUTI	; DO 'PRWAREG' ; SET INPUT/OUTPUT VECTORS ; SET INPUT/OUTPUT VECTORS ; TO BASIC. COLD START ; TO BASIC. COLD START ; TO BASIC. WARM START ; ADD TO PC IF SPECIFIED ; RESTORE FAKE REGISTERS ; AND GO! ; GO DISPLAY REGISTERS ; TRACE IS GONE ; STEP IS GONE ; JUMP TO CONTROL-Y VECTOR IN RAM ; JUMP TO CONTROL-Y VECTOR IN RAM

	605	BCC WR1	
FEE4: 90 EE FED4			5
FEE6: A0 22	606	LDY #\$22	
FEE8: 20 ED FE	607	JSR WRBYTE	
		BEQ BELL	
	608		
FEED: A2 10	609 WRBYTE	LDX #\$10	
FEEF: OA	610 WRBYT2	ASL A	
FEF0: 20 D6 FC	611	JSR WRBIT	
FEF3: DO FA FEEF	612	BNE WRBYT2	
FEID. DUTH FEEF			
FEF5: 60	613	RTS	
FEF6: 20 00 FE	614 CRMON	JSR BL1	HANDLE CR AS BLANK
FEF9: 68	615	PLA	; THEN POP STACK
FEFA: 68	616	PLA	AND RETURN TO MON
		BNE MONZ	
FEFB: DO 6C FF69	617		; (ALWAYS)
FEFD: 20 FA FC	618 READ	JSR RD2BIT	TAPE READ - FIND TAPEIN EDGE
FF00: A9 16	619	LDA #\$16	DELAY 3.5 SECONDS
FF00. A7 16			/ DEERT 3. 5 SEGURDS
FF02: 20 C9 FC	620	JSR HEADR	
FF05:85 2E	621	STA CHKSUM	; INITIAL CHECKSUM = \$FF
FF07 00 54 50			FIND AN EDGE LOOK FOR SYNC BIT
FF07: 20 FA FC	622	JSR RD2BIT	FIND AN EDGE
FFOA: AO 24	623 RD2	LDY #\$24	LOOK FOR SYNC BIT
FFOC: 20 FD FC	624	JSR RDBIT	; (SHORT O)
FFOF: BO F9 FFOA	625	BCS RD2	LOOP 'TIL FOUND
FF11:20 FD FC	626	JSR RDBIT	SKIP 2ND HALF CYCLE
		1 DY #\$3B	INDEX FOR 0/1 TEST
FF14: AO 3B	627		INDEX FOR OVI TEST
FF16:20 EC FC	628 RD3	JSR RDBYTE	;READ A BYTE ;PUT IT AT (A1)
FF19:81 3C	629	STA (A1L, X)	PUT IT AT (A1)
FF17.01 30		SIN (HIL/A/	
FF1B: 45 2E	630	EOR CHKSUM	UPDATE RUNNING CHECKSUM
FF1D:85 2E	631	STA CHKSUM	
FF1F:20 BA FC		JSR NXTA1	; INCR A1, COMPARE TO A2
	632		
FF22: A0 35	633	LDY #\$35	COMPENSATE 0/1 INDEX
FF24: 90 F0 FF16	634	BCC RD3	REPEAT TIL DONE.
FF26:20 EC FC	635	JSR RDBYTE	READ CHECKSUM BYTE
FF29: C5 2E	636	CMP CHKSUM	DOES THE RECORDED CHKSM MATCH DURS?
FF2B: FO OD FF3A	637	BEQ BELL	VED BEAD OK BEED AND DETUDN
			; YEP, READ OK. BEEP AND RETURN. ; PRINT 'ERR', THEN FALL INTO
FF2D: A9 C5	638 PRERR	LDA #\$C5	PRINT 'ERR', THEN FALL INTO
FF2F: 20 ED FD	639	JSR COUT	; FWEEPER.
			i rweeren.
FF32: A9 D2	640	LDA #\$D2	
FF34:20 ED FD	641	JSR COUT	
1134.20 20 10			
FF37:20 ED FD	642	JSR COUT	
FF3A: A9 87	643 BELL	LDA #\$87	MAKE A JOYFUL NOISE, THEN RETURN.
FEGG. 40 CD CD	644	JMP COUT	
FF3C:4C ED FD			
FF3F: A5 48	645 RESTORE	LDA STATUS	RESTORE 6502 REGISTER CONTENTS
FF41:48	646	PHA	; USED BY DEBUG SOFTWARE
	040		/ ODED DT DEDOG ODT TWINE
FF42: A5 45	647	LDA A5H	
FF44: A6 46	648 RESTR1	LDX XREG	
	649	LDY YREG	
FF46: A4 47			
FF48:28	650	PLP	
FF49: 60	651	RITS	
11 47.00			
FF4A:85 45	652 SAVE	STA A5H	SAVE 6502 REGISTER CONTENTS
FF4C:86 46	653 SAV1	STX XREG	; FOR DEBUG SOFTWARE
FF4E: 84 47	654	STY YREG	
FF50:08	655	PHP	
FF51:68	656	PLA	
	000		
FF52:85 48	657	STA STATUS	
FF54: BA	658	TSX	
FF55:86 49	659	STX SPNT	
FF57: D8	650	CLD	
FF58: 60	661	RTS	
1100.00			
FF59:20 84 FE	662 OLDRST	JSR SETNORM	; SET SCREEN MODE
FF5C:20 2F FB	663	JSR INIT	; AND INIT KBD/SCREEN
FF5F: 20 93 FE	664	JSR SETVID	; AS I/O DEVS.
FF62:20 89 FE	665	JSR SETKBD	
FF65: D8	666 MON	CLD	, MUST SET HEX MODE!
	667	JSR BELL	FWEEPER.
FF66:20 3A FF			
FF69: A9 AA	668 MONZ	LDA #\$AA	; '*' PROMPT FOR MONITOR
FF6B:85 33	669	STA PROMPT	
			ACTA A LANC OF THINK
FF6D:20 67 FD	670	JSR GETLNZ	READ A LINE OF INPUT
FF70:20 C7 FF	671	JSR ZMODE	CLEAR MONITOR MODE, SCAN IDX
FF73:20 A7 FF	672 NXTITM	JSR GETNUM	GET ITEM, NON-HEX
FF76:84 34	673		
FF78: A0 17		STY YSAV	; CHAR IN A-REG.
		LDY #\$17	; CHAR IN A-REG.
EE7A 00	674	LDY #\$17	; CHAR IN A-REG. ; X-REG=O IF NO HEX INPUT
FF7A: 88	674 675 CHRSRCH	LDY #\$17 DEY	; CHAR IN A-REG. ; X-REG=O IF NO HEX INPUT
FF7A: 88 FF7B: 30 E8 FF65	674 675 CHRSRCH 676	LDY #\$17 DEY BMI MON	; CHAR IN A-REG. ; X-REG=O IF NO HEX INPUT ;COMMAND NOT FOUND, BEEP % TRY AGAIN.
FF7A: 88 FF7B: 30 E8 FF65	674 675 CHRSRCH 676	LDY #\$17 DEY	; CHAR IN A-REG. ; X-REG=O IF NO HEX INPUT ;COMMAND NOT FOUND, BEEP % TRY AGAIN.
FF7A:88 FF7B:30 E8 FF65 FF7D:D9 CC FF	674 675 CHRSRCH 676 677	LDY #\$17 DEY BMI MON CMP CHRTBL,	; CHAR IN A-REG. ; X-REG=O IF NO HEX INPUT ; COMMAND NOT FOUND, BEEP & TRY AGAIN. y FIND COMMAND CHAR IN TABLE
FF7A: 88 FF7B: 30 E8 FF65 FF7D: D9 CC FF FF80: D0 F8 FF7A	674 675 CHRSRCH 676 677 678	LDY #\$17 DEY BMI MON CMP CHRTBL, BNE CHRSRCH	; CHAR IN A-REG. ; X-REG=O IF NO HEX INPUT ; COMMAND NOT FOUND, BEEP & TRY AGAIN. y FIND COMMAND CHAR IN TABLE
FF7A: 88 FF7B: 30 E8 FF65 FF7D: D9 CC FF FF80: D0 F8 FF7A	674 675 CHRSRCH 676 677	LDY #\$17 DEY BMI MON CMP CHRTBL,	; CHAR IN A-REG. ; X-REG=0 IF ND HEX INPUT ;COMMAND NDT FOUND, BEEP & TRY AGAIN. Y FIND COMMAND CHAR IN TABLE ;NOT THIS TIME ;QOT IT! CALL CORRESPONDING SUBROUTINE
FF7A:88 FF7B:30 E8 FF65 FF7D:D9 CC FF FF80:D0 F8 FF7A FF82:20 BE FF	674 675 CHRSRCH 676 677 678 679	LDY #\$17 DEY BMI MON CMP CHRTBL, BNE CHRSRCH JSR TOSUB	; CHAR IN A-REG. ; X-REG=0 IF ND HEX INPUT ;COMMAND NDT FOUND, BEEP & TRY AGAIN. Y FIND COMMAND CHAR IN TABLE ;NOT THIS TIME ;QOT IT! CALL CORRESPONDING SUBROUTINE
FF7A: 88 FF7B: 30 E8 FF65 FF7D: D9 CC FF FF80: D0 F8 FF7A FF82: 20 BE FF FF85: A4 34	674 675 CHRSRCH 676 677 678 679 680	LDY #\$17 DEY BMI MON CMP CHRTBL, BNE CHRSRCH JSR TOSUB LDY YSAV	; CHAR IN A-REG. ; X-REG=O IF NO HEX INPUT ; COMMAND NOT FOUND, BEEP & TRY AGAIN. y FIND COMMAND CHAR IN TABLE
FF7A:88 FF7B:30 E8 FF65 FF7D:D9 CC FF FF80:D0 F8 FF7A FF82:20 BE FF FF85:A4 34 FF87:4C 73 FF	674 675 CHRSRCH 676 677 678 679 680 681	LDY #\$17 DEY BMI MON CMP CHRTBL, BNE CHRSRCH JSR TOSUB LDY YSAV JMP NXTITM	; CHAR IN A-REG. ; X-REG=0 IF ND HEX INPUT ;COMMAND NDT FOUND, BEEP & TRY AGAIN. Y FIND COMMAND CHAR IN TABLE ;NOT THIS TIME ;QOT IT! CALL CORRESPONDING SUBROUTINE
FF7A:88 FF7B:30 E8 FF65 FF7D:D9 CC FF FF80:D0 F8 FF7A FF82:20 BE FF FF85:A4 34 FF87:4C 73 FF	674 675 CHRSRCH 676 677 678 679 680 681	LDY #\$17 DEY BMI MON CMP CHRTBL, BNE CHRSRCH JSR TOSUB LDY YSAV	; CHAR IN A-REG. ; X-REG=0 IF ND HEX INPUT ;COMMAND NDT FOUND, BEEP & TRY AGAIN. Y FIND COMMAND CHAR IN TABLE ;NOT THIS TIME ;QOT IT! CALL CORRESPONDING SUBROUTINE
FF7A:88 FF7B:30 E8 FF65 FF7D:D9 CC FF FF8D:D0 F8 FF7A FF82:20 BE FF FF85:A4 34 FF87:4C 73 FF FF8A:A2 03	674 675 CHRSRCH 676 677 678 679 680 681 682 DIG	LDY #\$17 DEY BMI MON CMP CHRTBL, BNE CHRSRCH JSR TOSUB LDY YSAV JMP NXTITM LDX #\$03	; CHAR IN A-REG. ; X-REG=0 IF ND HEX INPUT ;COMMAND NDT FOUND, BEEP & TRY AGAIN. Y FIND COMMAND CHAR IN TABLE ;NOT THIS TIME ;QOT IT! CALL CORRESPONDING SUBROUTINE
FF7A:88 FF7B:30 E8 FF65 FF7D:D9 CC FF FF8D:D0 FB FF7A FF82:20 BE FF FF85:A4 34 FF87:4C 73 FF FF8A:A2 03 FF8C:0A	674 675 CHRSRCH 676 677 678 679 680 681 682 DIG 683	LDY #\$17 DEY BMJ MON CMP CHRTBL, BNE CHRSRCH JSR TOSUB LDY YSAV JMP NXTITM LDX #\$03 ASL A	; CHAR IN A-REC. ; X-REG=0 IF NO HEX INPUT ; COMMAND NOT FOUND, BEEP & TRY AGAIN. Y ;FIND COMMAND CHAR IN TABLE ;NOT THIS TIME ;OT THIS TIME ;OT IT! CALL CORRESPONDING SUBROUTINE ;PROCESS NEXT ENTRY ON HIS LINE
FF7A:88 FF7B:30 E8 FF65 FF7D:D9 CC FF FF8D:D0 F8 FF7A FF82:20 BE FF FF85:A4 34 FF87:4C 73 FF FF8A:A2 03	674 675 CHRSRCH 676 677 678 679 680 681 682 DIG	LDY #\$17 DEY BMI MON CMP CHRTBL, BNE CHRSRCH JSR TOSUB LDY YSAV JMP NXTITM LDX #\$03	; CHAR IN A-REC. ; X-REG=0 IF ND HEX INPUT ; COMMAND NDT FOUND, BEEP & TRY AGAIN. Y FIND COMMAND CHAR IN TABLE ; NOT THIS TIME ; GOT TH? CALL CORRESPONDING SUBROUTINE ; PROCESS NEXT ENTRY ON HIS LINE ; GOT MEX DIGIT,
FF7A:B8 FF75:30 EB FF45 FF7D:D9 CC FF FF80:D0 F8 FF7A FF82:20 BE FF FF85:A4 34 FF87:4C 73 FF FF8A:A2 03 FF87:4C 73 FF87:6C:0A FF80:0A	674 675 CHRSRCH 676 677 678 679 680 681 682 DIG 683 683 684 684	LDY #\$17 DEY BMJ MON CMP CHRTBL, BNE CHRSRCH JSR TOSUB LDY YSAV JMP NXTITM LDX #\$03 ASL A	; CHAR IN A-REC. ; X-REG=0 IF ND HEX INPUT ; COMMAND NDT FOUND, BEEP & TRY AGAIN. ; FIND COMMAND CHAR IN TABLE ; NOT THIS TIME ; GOT TH? CALL CORRESPONDING SUBROUTINE ; PROCESS NEXT ENTRY ON HIS LINE ; GOT MEX DIGIT,
FF78:30 E8 FF45 FF70:30 E8 FF45 FF70:D9 CC FF FF80:D0 F8 FF7A FF82:20 BE FF FF85:24 34 FF87:4C 73 FF FF83:42 03 FF87:4C 73 FF FF83:42 03 FF80:0A FF80:0A	674 675 CHRSRCH 677 678 679 680 681 682 DIG 683 684 685 685	LDY #\$17 DEY BMI MON CMP CHRTBL, BNE CHRTBL, JSR TOSUB LDY YSAV JMP NXTITM LDX #\$03 ASL A ASL A ASL A	; CHAR IN A-REC. ; X-REG=0 IF NO HEX INPUT ; COMMAND NOT FOUND, BEEP & TRY AGAIN. Y ;FIND COMMAND CHAR IN TABLE ;NOT THIS TIME ;OT THIS TIME ;OT IT! CALL CORRESPONDING SUBROUTINE ;PROCESS NEXT ENTRY ON HIS LINE
FF7A: 88 FF78: 30 E8 FF65 FF70: D9 CC FF FF80: D0 F8 FF7A FF82: 20 BE FF FF87: 4C 73 FF FF87: 4C 73 FF FF87: 4C 73 FF FF87: 4C 73 FF FF80: 0A FF80: 0A FF80: 0A	674 675 CHRSRCH 676 677 678 679 681 681 682 DIG 683 684 683 684 685 685	LDY #\$17 DEY BMI MON CMP CHRTBL, BNE CHRTSCH JGR TOSUB LDY YSAV JMP NXTITM LDX #\$03 ASL A ASL A ASL A	; CHAR IN A-REC. ; X-REG=0 IF ND HEX INPUT ; COMMAND NDT FOUND, BEEP & TRY AGAIN. ; FIND COMMAND CHAR IN TABLE ; NOT THIS TIME ; GOT TH? CALL CORRESPONDING SUBROUTINE ; PROCESS NEXT ENTRY ON HIS LINE ; GOT MEX DIGIT,
FF7A: 88 FF78: 30 E8 FF65 FF70: D9 CC FF FF80: D0 F8 FF7A FF82: 20 BE FF FF87: 4C 73 FF FF87: 4C 73 FF FF87: 4C 73 FF FF87: 4C 73 FF FF80: 0A FF80: 0A FF80: 0A	674 675 CHRSRCH 677 678 679 680 681 682 DIG 683 684 685 685	LDY #\$17 DEY BMI MON CMP CHRTBL, BNE CHRTBL, JSR TOSUB LDY YSAV JMP NXTITM LDX #\$03 ASL A ASL A ASL A	; CHAR IN A-REC. ; X-REG=0 IF ND HEX INPUT ; COMMAND NDT FOUND, BEEP & TRY AGAIN. ; FIND COMMAND CHAR IN TABLE ; NOT THIS TIME ; GOT TH? CALL CORRESPONDING SUBROUTINE ; PROCESS NEXT ENTRY ON HIS LINE ; GOT MEX DIGIT,
FF7A: 88 FF73: 30 E8 FF65 FF70: 09 CC FF FF80: 00 F8 FF7A FF82: 20 BE FF FF85: A4 34 FF87: 4C 73 FF FF84: A2 03 FF80: 0A FF80: 0A FF86: 0A FF86: 0A	674 675 CHRSRCH 676 677 678 679 680 680 681 682 DIG 683 684 685 685 685	LDY #\$17 DEY BMI MON SME CHRTBL, BNE CHRTRCH JSR TOSUB LDY YSAV JMP NXTITM LDX #\$03 ASL A ASL A ASL A ASL A	; CHAR IN A-REC. ; X-REG=0 IF ND HEX INPUT ; COMMAND NDT FOUND, BEEP & TRY AGAIN. ; FIND COMMAND CHAR IN TABLE ; NOT THIS TIME ; GOT TH? CALL CORRESPONDING SUBROUTINE ; PROCESS NEXT ENTRY ON HIS LINE ; GOT MEX DIGIT,
FF7A: 88 FF7B: 30 E8 FF65 FF7D: D9 CC FF FF8D: D0 F8 FF7A FF8D: 20 BE FF FF8D: 44 34 FF8D: 0A FF8D: 0A FF8D: 0A FF8D: 0A FF8D: 0A FF8F: 0A FF97: 26 3E	674 675 CHRSRCH 676 677 679 680 681 682 DIG 683 684 685 685 686 685 688 888	LDY #\$17 DEY BMI MON CMP CHRTBL, JSR TOSUB JSR TOSUB LDY YSAV JMP NXTITM LDX \$403 ASL A ASL A ASL A ASL A ASL A	; CHAR IN A-REC. ; X-REG=0 IF ND HEX INPUT ; COMMAND NDT FOUND, BEEP & TRY AGAIN. ; FIND COMMAND CHAR IN TABLE ; NOT THIS TIME ; GOT TH? CALL CORRESPONDING SUBROUTINE ; PROCESS NEXT ENTRY ON HIS LINE ; GOT MEX DIGIT,
FF7A: 88 FF75: 30 E8 FF65 FF70: 09 CC FF FF80: 00 F8 FF7A FF82: 20 BE FF FF85: A4 34 FF87: 4C 73 FF FF87: 4C 73 FF FF87: 4C 73 FF FF87: 0A FF80: 0A FF87: 0A FF87: 0A FF79: 0A FF79: 26 3F	674 675 676 677 677 678 679 680 680 681 682 068 683 684 663 684 663 684 665 685 685 685 686 884 687 NXTBI1 688	LDY #\$17 DEY BMI MON CMP CHRTBL, BNE CHRTBL, DY YSAV JMP NXTITM LDY #403 ASL A ASL A ASL A ASL A ASL A ASL A ASL A RDL A2L RDL A2H	; CHAR IN A-REC. ; X-REG=0 IF NO HEX INPUT ; COMMAND NOT FOUND, BEEP % TRY AGAIN. ; FIND COMMAND CHAR IN TABLE ; NOT THIS TIME ; ODT IT! CALL CORRESPONDING SUBROUTINE ; PROCESS NEXT ENTRY ON HIS LINE ; GOT HEX DIGIT. ; SHIFT INTO A2
FF7A: 88 FF75: 30 E8 FF65 FF70: 09 CC FF FF80: 00 F8 FF7A FF82: 20 BE FF FF85: A4 34 FF87: 4C 73 FF FF87: 4C 73 FF FF87: 4C 73 FF FF87: 0A FF80: 0A FF87: 0A FF87: 0A FF79: 0A FF79: 26 3F	674 675 CHRSRCH 676 677 679 680 681 682 DIG 683 684 685 685 686 685 688 888	LDY #\$17 DEY BMI MON CMP CHRTBL, JSR TOSUB JSR TOSUB LDY YSAV JMP NXTITM LDX \$403 ASL A ASL A ASL A ASL A ASL A	; CHAR IN A-REC. ; X-REG=0 IF NO HEX INPUT ; COMMAND NOT FOUND, BEEP % TRY AGAIN. ; FIND COMMAND CHAR IN TABLE ; NOT THIS TIME ; ODT IT! CALL CORRESPONDING SUBROUTINE ; PROCESS NEXT ENTRY ON HIS LINE ; GOT HEX DIGIT. ; SHIFT INTO A2
FF7A: 08 FF7B: 30 E8 FF65 FF7D: D9 CC FF FF8D: D0 F8 FF7A FF8D: 20 BE FF FF8D: 44 34 FF8D: 0A FF8D: 0A FF8D: 0A FF8E: 0A FF8E: 0A FF8E: 0A FF8E: 0A FF8E: 0A FF91: 26 3E FF91: 26 3E FF93: 26 3F	674 675 CHRSRCH 676 677 677 679 680 681 682 082 DIG 683 684 685 685 884 685 888 888 888 889 689	LDY #117 DEY BMI MON CMP CHRTBL, JSR TOSUB JSR TOSUB LDY YSAV JMP NXTITM LDX #403 ASL A ASL A ASL A ASL A ASL A ASL A ROL A2L ROL A2L	; CHAR IN A-REC. ; X-REG=0 IF ND HEX INPUT ; COMMAND NDT FOUND, BEEP & TRY AGAIN. ; FIND COMMAND CHAR IN TABLE ; NOT THIS TIME ; GOT TH? CALL CORRESPONDING SUBROUTINE ; PROCESS NEXT ENTRY ON HIS LINE ; GOT MEX DIGIT,
FF7A: 88 FF75: 30 E8 FF65 FF70: 09 CC FF FF80: 00 F8 FF7A FF82: 20 BE FF FF85: A4 34 FF87: 4C 73 FF FF84: A2 34 FF82: 0A FF82: 0A FF82: 0A FF85: 0A FF79: 0A FF90: 0A 36 FF91: 26 36 FF97: 26 3F FF97: CA 56 FF97:	674 675 CHRSRCH 676 677 678 677 680 680 681 682 068 682 068 683 684 685 684 685 686 687 NXTBI1 688 687 887 687	LDY #\$17 DEY DEY BMI MON SME CHRTBL, JSR TOSUB LDY YSAV JMP NXTIITM NXTIITM H\$03 ASL A ASL A ASL A ASL A ASL A ASL A ASL A LDL A2L RUL A2H DEX BPL NXTBIT	; CHAR IN A-REC. ; X-REG=0 IF NO HEX INPUT ; COMMAND NOT FOUND, BEEP % TRY AGAIN. ; FIND COMMAND CHAR IN TABLE ; NOT THIS TIME ; ODT IT! CALL CORRESPONDING SUBROUTINE ; PROCESS NEXT ENTRY ON HIS LINE ; GOT HEX DIGIT. ; SHIFT INTO A2
FF7A: 88 FF7B: 30 E8 FF65 FF7D: D9 CC FF FF8D: D0 F8 FF7A FF8D: D0 F8 FF7A FF8D: 20 BE FF FF8D: 43 44 FF8D: 0A FF8D: 0A FF8D: 0A FF8E: 0A FF8E: 0A FF8E: 0A FF90: 26 3E FF91: 26 3E FF92: 26 3F FF92: 26 FF90 FF95: 10 F8 FF90 FF98: 3 1	674 675 CHRSRCH 676 677 678 679 679 680 681 682 082 082 084 685 685 888 685 888 686 887 888 687 888 689 689 889 689	LDY #117 DEY BMI MON CMP CHRTBL, JSR TOSUB JSR TOSUB LDY YSAV JMP NXTITM LDX #403 ASL A ASL A ASL A ASL A ASL A ASL A ROL A2L ROL A2L ROL A2L DEX BPL NXTBIT LDA MODE	; CHAR IN A-REC. ; X-REG=0 IF NO HEX INPUT ; COMMAND NOT FOUND, BEEP & TRY AGAIN. ; FIND COMMAND CHAR IN TABLE ; NOT THIS TIME ; GOT THI: CALL CORRESPONDING SUBROUTINE ; PROCESS NEXT ENTRY ON HIS LINE ; GOT HEX DIGIT. ; SHIFT INTO A2 ;LEAVE X=\$FF IF DIG
FF7A: 88 FF7B: 30 E8 FF65 FF7D: D9 CC FF FF8D: D0 F8 FF7A FF8D: D0 F8 FF7A FF8D: 20 BE FF FF8D: 43 44 FF8D: 0A FF8D: 0A FF8D: 0A FF8E: 0A FF8E: 0A FF8E: 0A FF90: 26 3E FF91: 26 3E FF92: 26 3F FF92: 26 FF90 FF95: 10 F8 FF90 FF98: 3 1	674 675 CHRSRCH 676 677 678 679 679 680 681 682 082 082 084 685 685 888 685 888 686 887 888 687 888 689 689 889 689	LDY #117 DEY BMI MON CMP CHRTBL, JSR TOSUB JSR TOSUB LDY YSAV JMP NXTITM LDX #403 ASL A ASL A ASL A ASL A ASL A ASL A ASL A ROL A2L ROL A2L ROL A2L ROL A2L ROL NXTBIT LDA MODE	; CHAR IN A-REC. ; X-REG=0 IF NO HEX INPUT ; COMMAND NOT FOUND, BEEP & TRY AGAIN. ; FIND COMMAND CHAR IN TABLE ; NOT THIS TIME ; GOT THI: CALL CORRESPONDING SUBROUTINE ; PROCESS NEXT ENTRY ON HIS LINE ; GOT HEX DIGIT. ; SHIFT INTO A2 ;LEAVE X=\$FF IF DIG
FF7A: 88 FF75: 30 E8 FF65 FF70: 09 CC FF FF80: 00 F8 FF7A FF82: 20 BE FF FF85: A4 34 FF87: 4C 73 FF FF84: A3 44 FF87: 4C 73 FF FF82: 0A FF88: 0A FF88: 0A FF88: 0A FF87: 0A FF90: 0A FF90: 0A FF97: 26 3E FF97: 26 3F FF97: CA FF96: 10 F8 FF90 FF98: A5 31 FF96: 10 66 FF42	674 675 CHRSRCH 676 677 678 678 679 680 680 681 682 684 685 684 685 686 685 686 687 NXTBI1 688 687 687 687 687 689 689 693	LDY #\$17 DEY DEY BMI MON CMP CHRTBL, JSR TOSUB LDY YSAV JMP NXTITM NXTITM NXTITM ASL A ASL A BOL A2L DEX BPL NXTBIT LDA MDDE BNE NXTBS2	<pre>; CHAR IN A-REC. ; X-REG=0 IF NO HEX INPUT ; COMMAND NOT FOUND, BEEP % TRY AGAIN. ;FIND COMMAND CHAR IN TABLE ;NOT THIS TIME ;OOT IT! CALL CORRESPONDING SUBROUTINE ;PROCESS NEXT ENTRY ON HIS LINE ;GOT HEX DIGIT, ; SHIFT INTO A2 ;LEAVE X=\$FF IF DIG ;IF MODE IS ZERO.</pre>
FF7A: 88 FF7B: 30 E8 FF65 FF7D: 09 CC FF FF8D: 00 F8 FF7A FF8D: 00 F8 FF7A FF8D: 04 FF77 FF8D: 04 FF8D: 04 FF8D: 04 FF8D: 04 FF8E: 04 FF79: 04 FF90: 04 FF90: 04 FF91: 26 3F FF92: 26 FF92: 26 FF92: 26 FF92: 57 FF92: 55 FF92: 55 F	674 675 CHRSRCH 676 677 677 679 680 681 682 082 082 084 685 685 888 685 888 685 888 687 870 690 691 888 689 687 888 689 687 688 689 687 688 689 689 689 689 689 689 689 689 689	LDY #117 DEY BMI MON CMP CHRTBL, JSR TOSUB LDY YSAV JMP NXTITM LDX #103 ASL A ASL A ASL A ASL A ASL A ASL A ASL A ASL A BPL NXTBIT DEX BPL NXTBIT LDA MODE BNR NXTBSZ LDA A2H, X	; CHAR IN A-REC. ; X-REG=0 IF NO HEX INPUT ; COMMAND NOT FOUND, BEEP & TRY AGAIN. ; FIND COMMAND CHAR IN TABLE ; NOT THIS TIME ; GOT THI: CALL CORRESPONDING SUBROUTINE ; PROCESS NEXT ENTRY ON HIS LINE ; GOT HEX DIGIT. ; SHIFT INTO A2 ;LEAVE X=\$FF IF DIG
FF7A: 88 FF75: 30 E8 FF65 FF70: 09 CC FF FF80: 00 F8 FF7A FF82: 20 BE FF FF85: A4 34 FF84: A3 34 FF84: A3 34 FF86: 0A FF86: 0A FF86: 0A FF87: 0A FF70: 0A FF70: 0A FF70: 26 35 FF73: 26 3F FF75: CA FF76: 10 F8 FF90 FF76: 10 F8 FF90 FF76: 31 FF76: 35 3F FF76: 35 3F	674 675 CHRSRCH 676 677 678 678 679 680 680 681 682 684 685 684 685 686 685 686 687 NXTBI1 688 687 687 687 687 689 689 693	LDY #\$17 DEY BMI MON CMP CHRTBL, BNE JSR TDSUB BLY JMN CHRSRCH JSR JSR TDSUB BLY JMP NXTITM LDY LDX #603 ASL ASL A ASL ASL A ACL ROL A2L ROL DEX BPL NXTBIT DA ACH, X STA STA ALH, X STA	<pre>; CHAR IN A-REC. ; X-REG=0 IF NO HEX INPUT ; COMMAND NOT FOUND, BEEP % TRY AGAIN. ;FIND COMMAND CHAR IN TABLE ;NOT THIS TIME ;OOT IT! CALL CORRESPONDING SUBROUTINE ;PROCESS NEXT ENTRY ON HIS LINE ;GOT HEX DIGIT, ; SHIFT INTO A2 ;LEAVE X=\$FF IF DIG ;IF MODE IS ZERO.</pre>
FF7A: 88 FF75: 30 E8 FF65 FF70: 09 CC FF FF80: 00 F8 FF7A FF82: 20 BE FF FF85: A4 34 FF84: A3 34 FF84: A3 34 FF86: 0A FF86: 0A FF86: 0A FF87: 0A FF70: 0A FF70: 0A FF70: 26 35 FF73: 26 3F FF75: CA FF76: 10 F8 FF90 FF76: 10 F8 FF90 FF76: 31 FF76: 35 3F FF76: 35 3F	674 675 676 677 677 678 679 680 681 682 082 068 684 685 684 685 684 685 686 697 690 699 699 699 699 693 693 693 693	LDY #\$17 DEY BMI MON CMP CHRTBL, BNE JSR TDSUB BLY JMN CHRSRCH JSR JSR TDSUB BLY JMP NXTITM LDY LDX #603 ASL ASL A ASL ASL A ACL ROL A2L ROL DEX BPL NXTBIT DA ACH, X STA STA ALH, X STA	<pre>; CHAR IN A-REC. ; X-REG=0 IF NO HEX INPUT ; COMMAND NOT FOUND, BEEP % TRY AGAIN. ;FIND COMMAND CHAR IN TABLE ;NOT THIS TIME ;OOT IT! CALL CORRESPONDING SUBROUTINE ;PROCESS NEXT ENTRY ON HIS LINE ;GOT HEX DIGIT, ; SHIFT INTO A2 ;LEAVE X=\$FF IF DIG ;IF MODE IS ZERO.</pre>
FF7A: 88 FF7B: 30 E8 FF65 FF7D: 09 CC FF FF8D: 00 F8 FF7A FF8D: 00 F8 FF7A FF8D: 04 FF77 FF8D: 04 FF8D: 04 FF8D: 04 FF8D: 04 FF8E: 04 FF8E: 04 FF90: 04 FF90: 04 FF91: 26 3F FF92: 26 FF92: 26 FF92: 26 FF92: 57 FF92: 55 FF92: 55 F	674 675 CHRSRCH 676 677 677 679 680 681 682 082 082 084 685 685 888 685 888 685 888 687 870 690 691 888 689 687 888 689 687 688 689 687 688 689 689 689 689 689 689 689 689 689	LDY ##17 DEY BMI MON CMP CHRTBL, BNE JRM CHRSRCH JSR JSR TOSUB BNE LDY YSAV JMP JMP NXTITM ASL ASL A ASL ASL A ASL ROL A2L ROL BL NXTBIT LDA MDE BNE NXTBIT LDA A2H, X STA	<pre>; CHAR IN A-REC. ; X-REG=0 IF NO HEX INPUT ; COMMAND NOT FOUND, BEEP % TRY AGAIN. ;FIND COMMAND CHAR IN TABLE ;NOT THIS TIME ;OOT IT! CALL CORRESPONDING SUBROUTINE ;PROCESS NEXT ENTRY ON HIS LINE ;GOT HEX DIGIT, ; SHIFT INTO A2 ;LEAVE X=\$FF IF DIG ;IF MODE IS ZERO.</pre>

FFA2: EB	697 NXTBS2	INX		
FFA3: FO F3 FF98	698	BEQ	NXTBAS	
FFA5: DO 06 FFAD	699	BNE	NXTCHR	
FFA7: A2 00	700 GETNUM	LDX	#\$00	CLEAR A2
FFA9:86 3E	701	STX	A2L	
FFAB: 86 3F	702	STX	A2H	
FFAD: 89 00 02	703 NXTCHR	LDA	IN, Y	;GET CHAR
FFBO: C8	704	INY		
FFB1:49 B0	705	EOR	#\$BO	
FFB3:C9 0A	706	CMP	#\$0A	
FFB5: 90 D3 FF8A	707	BCC	DIG	; BR IF HEX DIGIT
FFB7: 69 88	708 709	ADC CMP	#\$88 #\$FA	
FFB9:C9 FA FFBB:B0 CD FF8A	710	BCS	DIG	
FFBD: 60	711	RTS	010	
FFBE: A9 FE	712 TOSUB	LDA	# <g0< td=""><td>DISPATCH TO SUBROUTINE, BY</td></g0<>	DISPATCH TO SUBROUTINE, BY
FFCO: 48	713	PHA		PUSHING THE HI-ORDER SUBR ADDR.
FFC1: B9 E3 FF	714	LDA	SUBTBL, Y	THEN THE LO-ORDER SUBR ADDR
FFC4: 48	715	PHA		; ONTO THE STACK,
FFC5: A5 31	716	LDA	MODE	; (CLEARING THE MODE, SAVE THE OLD
FFC7: A0 00	717 ZMODE	LDY	#\$00	; MODE IN A-REG),
FFC9:84 31	718	STY	MODE	
FFCB: 60	719	RTS		; AND 'RTS' TO THE SUBROUTINE!
FFCC: BC	720 CHRTBL	DFB	\$BC	; CC (BASIC WARM START)
FFCD: B2	721	DFB	\$82	; ^Y (USER VECTOR)
FFCE: BE	722	DFB	\$BE	; ^E (DPEN AND DISPLAY REGISTERS)
FFCF: B2	723	DFB	\$B2	; T (ONCE WAS TRACE; NEVER USED.)
FFDO: EF	724	DFB	\$EF	V (MEMORY VERIFY)
FFD1:C4	725	DFB	\$C4	; ^K (IN#SLOT)
FFD2: B2	726	DFB	\$B2	;S (DNCE WAS STEP; NOW NEVER USED.) ;^P (PR#SLOT)
FFD3: A9	727	DFB	\$A9	;^P (PR#SLOT) ;^B (BASIC COLD START)
FFD4: BB	728	DFB DFB	\$BB \$A6	/ - (SUBTRACTION)
FFD5: A6 FFD6: A4	729 730	DFB	\$A4	; '+' (ADDITION)
FFD7:06	731	DFB	\$06	M (MEMORY MOVE)
FFD8: 95	732	DFB	\$95	; '<' (DELIMITER FOR MOVE, VFY)
FFD9:07	733	DFB	\$07	N (SET NORMAL VIDED)
FEDA: 02	734	DFB	\$02	I (SET INVERSE VIDEO)
FFDB: 05	735	DFB	\$05	(DISASSEMBLE 20 INSTRS)
FFDC: FO	736	DFB	\$FO	W (WRITE TO TAPE)
FFDD: 00	737	DFB	\$00	G (EXECUTE PROGRAM)
FFDE: EB	738	DFB	\$EB	; R (READ FROM TAPE)
FFDF: 93	739	DFB	\$93	;':' (MEMORY FILL)
FFEO: A7	740	DFB	\$A7	; '. ' (ADDRESS DELIMITER)
FFE1: C6	741	DEB	\$C6	; 'CR' (END OF INPUT)
FFE2: 99	742	DFB	\$99	BLANK
FFE3: B2	743 SUBTBL	DFB	\$B2	TABLE OF LO-ORDER MONITOR ROUTINE
FFE4: C9	744	DFB	\$C9	; DISPATCH ADDRESSES
FFE5: BE	745	DFB	\$BE	
FFE6:C1	746	DFB	\$C1	
FFE7: 35 FFE8: 8C	747 748	DFB DFB	\$35 \$80	
FFE9: C4	749	DFB	\$C4	
FFEA: 96	750	DFB	\$96	
FFEB: AF	751	DFB	\$AF	
FFEC: 17	752	DFB	\$17	
FFED: 17	753	DFB	\$17	
FFEE: 2B	754	DFB	\$2B	
FFEF: 1F	755	DFB	\$1F	
FFF0: 83	756	DFB	\$83	
FFF1:7F	757	DFB	\$7F	
FFF2: 5D	758	DFB	\$5D	
FFF3: CC	759	DFB	\$CC	
FFF4: B5	760	DFB	\$B5	
FFF5: FC	761	DFB	\$FC	
FFF6: 17	762	DFB	\$17	
FFF7: 17	763	DFB	\$17 \$F5	
FFF8: F5	764	DFB	\$F03	
FFF9:03 FFFA:FB 03	765 766	DFB	NMI	; NON-MASKABLE INTERRUPT VECTOR
FFFC: 62 FA	767	DW	RESET	RESET VECTOR
FFFE: 40 FA	768	DW	IRG	INTERRUPT REQUEST VECTOR
		~		

Monitor Symbol Table, Sorted by Symbol

				-		-	
30	AIH	36	AIL	FE/J	A1PC	FE/8	AIPULP
FE7F	AIPCRTS	3F	A2H	3E	A21_	41	A3H
40	ABL	43	A4H	42	A4L	45	A5H
44	A5L	45	ACC	FDD1	ADD	FD84	ADDINP
FBF4	ADVANCE	?03F5	AMPERV	01	APPLE2E	FB60	APPLEII
? 2B	BAS2H	? 2A	BAS2L	FBC1	BASCALC	FBDO	BASCLC2
2FEB3	BASCONT	29	BASH	F000	BASIC	E003	BAS1C2
	BACI	5071	BCKEBC	EBDG	BELLI	EDEA	BELLO
5534	DRUL	FD/1	BUKSFG	FBD7	DELLI	FBE4	DECLE
FFJA	BELL	FEOO	BLI	TEU4	BLANK	FA4C	BREAK
0.3F0	BRKV	FC10	BS	FD62	CANCEL	FD7E	CAPISI
F984	CHAR1	F9BA	CHAR2	2E	CHKSUM	FF7A	CHRSRCH
24	CH	FFCC	CHRTBL	?FEAF	CKSUMFIX	20059	CLRANO
2C05B	CLRAN1	?C05D	CLRAN2	2005F	CLRAN3	FC9C	CLREDL
2EC9E	CLREOL7	2EC42	CLREDP	CEEE	CI BROM	E838	CLRSC2
FRAC	CLRSC3	25832	CLRSCR	5934	CLETOP	30	COLOR
EDED	COUT	EDEO	COUTI	EDE4	COUTZ	EDOE	CREWIX
FDCD	0001	FUFU	CUUTI	- DFO	00012	FUEL	CRUCI
FLOR	CR	PEF6	CRIMUN	· 3/	USWH	36	LSWL
25	CV	FDB6	DATADUT	FF8A	DIG	F802	DISKID
F8A5	ERR	FD2F	ESC	FC2C	ESC1	FBA5	ESCNEW
FB9B	ESCNOW	FB97	ESCOLD	FA9B	FIXSEV	F962	FMT1
F9A6	FMT2	2E	FORMAT	F847	GBASCALC	27	GBASH
26	GBASI	E856	GBCALC	F8A9	GETEMT	2FD6A	GETL N
ED67	GETLN7	FEA7	GETNUM	FRRA	COTOCY	FERA	60
1 207	10	ECC0.	USADD	20057	UIDEE	COSE	UTCCD.
05040	rie,	FCC7	HENDR	20007	HIRES	20000	HISCH
21814	HLINE	F81C	HLINE1	FC58	HOME	F848	TEVEN
?FA6F	INITAN	FB2F	INIT	?FE8B	INPORT	?FE8D	INPRT
F882	INSDS1	?F88C	INSDS2	FBDO	INSTDSP	32	INVFLG
0200	IN	C000	IDADR	FEA7	IOPRT1	FEA9	IOPRT2
FE9B	IOPRT	03FE	IRQLOC	FA40	IRG	FC99	ISPAGE1
FC91	ISSI DIS	0000	KBD	C010	KBDSTRB	FR88	KBDWATT
FDIR	KEYIN	2 39	KGUH	79	KSHI	25	ASTIN
75	LENOTH	ECLL	1 C	0400	L THE 1	EE40	LISTO
25	LENGIN	FLOO	LF	0400	LINEI	FE63	LISIE
THESE	L151	50	LINEM	00	2000	01	LUCI
C056	LORES	C054	LOWSCR	?FE20	LT	FE22	LT2
2E	MASK	20052	MIXCLR	C053	MIXSET	F9C0	MNEML
FAOO	MNEMR	FBBE	MNNDX1	F8C2	MNNDX2	FBC9	MNNDX3
FDAD	MODSCHK	31	MODE	FF65	MON	FF67	MONZ
FE2C	MOVE	07F8	MSLOT	2FAB1	NEWMON	OBER	NMI
FAAR	NOFIX	EDSE	NOTCR1	ED3D	NOTCR	EB94	NOUATT
FCRA	NYTAI	ECRA	NYTAA	5500	NYTRAC	FERO	NYTRIT
FEAG	NYTROO	FCD4	NATAT	CD 70	NATEMS	FF 70	NYTCHO
FFAZ	NX IBS2	FAC/	NXIBYI	FD/5	NXICHAR	FFAD	NXICHR
3F82F	NX I COL	FF73	NXTITM	FA59	OLDBRK	7FF 59	DLDRST
FCE2	ONEDLY	?FE95	OUTPORT	?FE97	OUTPRT	C064	PADDLO
?F954	PCADJ2	F956	PCADJ3	F953	PCADJ	F95C	PCADJ4
3B	PCH	3A	PCL	? 95	PICK	FBOE	PLOT1
F800	PLOT	FD92	PRA1	F910	PRADR1	F914	PRADR2
F926	PRADES	F92A	PRADR4	E930	PRADRS	F944	PRBI 2
25940	BBBL 3	5949	DODI NIK	EDDA	BOBYTE	EBOS	BBEADO
OFRIC	PREAD	000000	DOCOD	CDCE	DOUCYZ	05050	DDUCY
FDIE	FREAD	TFED	PRERK	FDED	PRHEAZ	PDE3	PRHEA
FBFD	PRMNI	F8F9	PRMN2	21 941	PRNIAX	FRDB	PRNIBL
FBD4	PRNTOP	?F944	PRNTX	F940	PRNTYX	33	PROMPT
FD96	PRYX2	C070	PTRIG	FAFD	PWRCON	03F4	PWREDUP
FAA6	PWRUP	FCFA	RD2BIT	FFOA	RD2	FF16	RD3
C018	RDBOSTORE	FCFD	RDBIT	FCEE	RDBYT2	FCEC	RDBYTE
FD35	RDCHAR	C015	RDCXRDM	FC84	RDCX	FD21	RDESC
EDOC	RDKEY	COLC	RDPACE2	FAF4	RDSP1	2FFFD	READ
EAD7	RECOSP	SEEBE	REO7	25020	RELADR	EA47	RESET
EEOF	RECTORE	DEEAA	REGI I	- F 7 3 8	BODGRI	r md2	DMNEM
FFJF	RESIDRE	76644	RESIRI	PADA	RGDSPI	20	REINER
? 4F	RNDH	? 4E	RNDL	FB19	RIBL	F80C	RIMASK
F87F	RTMSKZ	FB31	RTS1	F961	RTS2	FBEF	RTS2B
FB2E	RTS2D	FBFC	RTS3	FCC8	RTS4B	?FDC5	RTS4C
FC2B	RTS4	FE17	RTS5	FF4C	SAV1	?FF4A	SAVE
2F871	SCRN	FB79	SCRN2	?FC70	SCROLL	C058	SETANO
C054	SETAN1	20050	SETAN2	2005E	SETANG	2F864	SETCOL
25840	SETCR	FEPA	SETTELO	0007	SETINTCYROM	SEEBO	SETINU
FERO	CETURN	FE10	CETMD7	25510	CETMODE	FERM	SETNORM
FE89	SCINBU	FEID	SE I FIDZ	SEE 18	SE I MODE	FE84	OCTOLOTOVOCH
21 AA9	BE IFGS	FAAB	SEIFLE	FB6F	DETENIC	0006	SCISLUICXRUM
?FB39	SETTXT	FE93	SETVID	FB4B	SETWND	? 2F	SIGN
FABA	SLOOP	03F2	SOFTEV	C030	SPKR	49	SPNT
48	STATUS	?FEC4	STEPZ	FB65	STITLE	?FEOB	STOR
FBFO	STORADV	FFE3	SUBTBL	?FB5B	TABV	C060	TAPEIN
C020	TAPEOUT	FB09	TITLE	FFBE	TOSUB	?FEC2	TRACE
C050	TXTCLR	C051	TXTSET	FC1A	UP	?FECA	USR
OBER	USRADR	20	102	2FBP3	VERSION	FE58	VEYOK
EE34	UEV	EPEN	UTDOUT	EB70	UTDUATT	F829	VLINE
F03/		FCCC	UTAD	5004	UTAD7	FCAR	LIATT
F020	VLINEZ UATTO	FC22	A1L A2H A2H A4H ACC BA3H BCKSPC BA3H BCKSPC BL1 BS CHAR2 CHAR2 CHAR2 CLREOP CLREOP CLREOP CLREOP CLREOP CLREOP CLRAN2 CLREOP CLRAN2 CLREOP CLRAN2 CLREOP CLRAN2 CLREOP CLRAN2 CLREOP CLRAN2 CLREOP CLRAN2 CLREOP CLRAN2 CLREOP CLRAN2 CLREOP CLRAN2 CLREOP CLRAN2 CLREOP CLRAN2 CLREOP CLRAN2 CLREOP CLRAN2 CLREOP CLRAN2 CLREOP CLRAN2 CLREOP CLRAN2 CLRAN4 CLR	FU24		FUAB	UNDI ET
r CA9	WHILE	FCAA	CITHM	23	WINDBIN	20	HINDLP I

Monitor Symbol Table, Sorted by Symbol

22	WNDTOP	21	WNDWDTH	FED4	WR1	FCD6	WRBIT
FEEF	WRBYT2	FEED	WRBYTE	?FECD	WRITE	FCE5	WRTAPE
FDB3	XAM	FDA3	XAMB	FDC6	XAMPM	?FEBO	XBASIC
FC72	XGOTOCX	FB11	XLTBL	46	XREG	47	YREG
34	YSAV	35	YSAV1	FCDB	ZERDLY	FFC7	ZMODE
** SU	CCESSFUL A	SSEMBLY :=	NO ERROR	5			
** AS	SEMBLER CR	EATED ON 05	JAN-82	000004			
** TO	TAL LINES	ASSEMBLED	1435				
** FR	EE SPACE P	AGE COUNT	67				
2	BJS. SRC2						

Monitor Symbol Table, Sorted by Address

00 L0C0	01	APPLE2E	01	LOC1	20	WNDL.FT
21 WNDWDTH	22	WNDTOP	23	WNDBTM	24	СН
25 CV	26	GBASL	27	GBASH	28	BASL
29 BASH	? 2A	BAS2L	? 2B	BAS2H	20	H2
2C LMNEM	2D	V2	20	RMNEM	2E	MASK
2E CHKSUM	2E	FORMAT	2F	LASTIN	2F	LENGTH
? 2F SIGN	30	COLOR	31	MODE	32	INVFLG
33 PROMPT	34	YSAV	35	YSAV1	36	CSWL
? 37 CSWH	38	KSWL	? 39	KSWH	ЗA	PCL
3B PCH	30	AIL	30	AIH	3E	A2L
3F A2H	40	A3L	41	A3H	42	A4L
43 A4H	44	ASL	45	ASH	45	ACC
46 XREG	4/	YREG	48	STATUS	49	SPNI
4E RNDL	· 7 4⊦	RNDH	2 95	PICK	0200	IN
OSFO BRKV	03F2	SUFIEV	03F4	PWREDUP	20345	AMPERV
OJES MELOT	COOO	IDADE	COOO	KOLUC	0400	CETCL OTCYDOM
COOT SETINICYBOM	0000	KEDETER	C000	RDCYROM	C018	PROCEEDE
COLC BDBAOS2	010	TARCOUT	013	COKP	010	TYTCLP
COS1 TYTEET	20052	MIXCLE	0050	MIYCET	0054	
20055 HISCR	054	LUBES	20057	HIRES	058	SETANO
20059 CL RANO	C054	SETANI	20058	CLRANI	20050	SETANZ
20050 CL RAN2	20055	SETANS	20055	CL RAN3	0000	TAPEIN
CO64 PADDI O	0070	PTRIG	CEEE	CLRROM	E000	BASIC
E003 BASIC2	E800	PLOT	FBOC	RTMASK	FBOE	PLOT1
2F819 HLINE	FBIC	HLINE1	F826	VLINEZ	F828	VLINE
F831 RT51	2F832	CLRSCR	FB36	CLRTOP	F838	CLRSC2
F83C CLRSC3	F847	GBASCALC	F856	GBCALC	?F85F	NXTCOL
?F864 SETCOL	7F871	SCRN	F879	SCRN2	F87F	RTMSKZ
F882 INSDS1	?F88C	INSDS2	F89B	IEVEN	F8A5	ERR
F8A9 GETFMT	FBBE	MNNDX1	FBC2	MNNDX2	F8C9	MNNDX3
F8D0 INSTDSP	F8D4	PRNTOP	FBDB	PRNTBL	F8F5	PRMN1
F8F9 PRMN2	F910	PRADR1	F914	PRADR2	F926	PRADR3
F92A PRADR4	F930	PRADR5	?F938	RELADR	F940	PRNTYX
?F941 PRNTAX	?F944	PRNTX	F948	PRBLNK	F94A	PRBL2
PRBL3	F953	PCADJ	?F954	PCADJ2	F956	PCADJ3
F95C PCADJ4	F961	RT52	F962	FMT1	F9A6	FMT2
F9B4 CHAR1	F9BA	CHAR2	F9C0	MNEML	FAOO	MNEMR
FA40 IRQ	FA4C	BREAK	FA59	OLDBRK	FA62	RESET
7FA6F INITAN	?FA81	NEWMON	FA9B	FIXSEV	FAAG	NOFIX
FAA6 PWRUP	?FAA9	SETPG3	FAAB	SETPLP	FABA	SLOOP
FAC7 NXTBYT	FAD7	REGDSP	FADA	RGDSP1	FAE4	RDSP1
FAFD PWRCON	FB02	DISKID	FB09	TITLE	F811	XL TBL
FB19 RTBL	PFBIE	PREAD	FB25	PREAD2	FB2E	RISZD
PB2P INII	2FB39	SELLAL	2PB40	SEIGR	7846	SETRURC
FROB LIDUALT	F860	APPLEII	FB60	NOUATT	(F86F	SET PWRL
FB78 VIDWAIT	FB88	KBDWALL	P 574	NUWATI	FB7/	COTOCX
FB7B ESCNUW	FBAD	BARCLOD	(FBB3	PERSION	FBB4	BELLO
EREE PIERP	FBDO	STOPADU	EDEA	ADUANCE	EDEC	BICCE
FRED VIDOUT	EC10	85	EC14	UP	FC22	VTAB
EC24 VTAB7	FC2B	RTS4	EC2C	ESC1	2EC42	CLREDP
FC58 HOME	EC62	CR	EC66	LE	2FC70	SCROLL
FC72 XGBTDCX	FC84	RDCX	FC91	ISSLOTS	FC99	ISPAGE1
FC9C CLREDL	?FC9E	CLREOLZ	FCAB	WAIT	FCA9	WAIT2
FCAA WAIT3	FCB4	NXTA4	FCBA	NXTA1	FCCB	RTS4B
FCC9 HEADR	FCD6	WRBIT	FCDB	ZERDLY	FCE2	ONEDLY
FCE5 WRTAPE	FCEC	RDBYTE	FCEE	RDBYT2	FCFA	RD2BIT
FCFD RDBIT	FDOC	RDKEY	FD1B	KEYIN	FD21	RDESC
FD2F ESC	FD35	RDCHAR	FD3D	NOTCR	FD5F	NOTCR1
FD62 CANCEL	FD67	GETLNZ	?FD6A	GETLN	FD71	BCKSPC
FD75 NXTCHAR	FD7E	CAPTST	FD84	ADDINP	FDBE	CROUT
FD92 PRA1	FD96	PRYX2	FDAG	XAMB	FDAD	MODSCHK
FDB3 XAM	FDB6	DATAOUT	?FDC5	RTS4C	FDC6	XAMPM
FDD1 ADD	FDDA	PRBYTE	?FDEG	PRHEX	FDE5	PRHEXZ
FDED COUT	FDFO	COUT1	FDF6	COUTZ	FE00	BL.1
PEO4 BLANK	?FEOB	STOR	FE17	RTS5	?FE18	SETMODE
FEID SETMDZ	?FE20	LT	FE22	LT2	FE2C	MOVE
FEJ6 VFY	FE.58	VEYOK	7FE5E	LIST	FE63	LIST2
FEAD AIPC	FE78	AIPCLP	FE7F	AIPCRIS	PFE80	SET INV
PEGA SEINURM	FE86	SETUTA	7E87	OUTBORT	CFE88	INPUK I
FEOR INPRT	FE73	TOPPT1	(FE93	TOPPTO	CEEY/	CKCUMETY
2EEBO XBASIC	25592	BASCONT	FER4	OD	2FEAF	DEC7
2FEC2 TRACE '	2FEC4	STEP7	25504	USP	2FECD	LIR TTE
FED4 WR1	FEED	WRBYTE	FEFF	WRBYT2	2FFFA	CRMON
00 LOCO 21 WNDWDTH 25 CV 27 BASH 22 CLNNEM 22 CLNNEM 23 FACM 33 FACM 33 FACM 33 FACM 43 A4H 44 XREG 7 4E RNDL 03F0 USRADR 07F8 MSL0T 03F8 USRADR 07F8 MSL0T 03F8 USRADR 07F8 MSL0T 03F8 USRADR 07F8 MSL0T 0075 ETINTCXROM 001C RDPAGE2 0031 KTSET 70059 CLRAN0 70059 CLRAN0 77645 FIST F831 RTS1 F832 CLRSC3 77F34 SETCOL F832 INSDS1 F845 GETFMT F800 INSTDSP F859 PRMN2 F952 CCADJ4 F984 CHAR1 F800 INSTDSP F859 PRMN2 F974 PRNTA 75740 FRS1 F9750 CLRSC3 77F34 SETCOL F819 RTS1 F827 GLRSC3 F950 CCADJ4 F984 CHAR1 F819 RTS1 F825 FINIT 7855 TASU F857 KIDWAIT F819 RTS2 F827 KIDWAIT F819 RTS2 F827 KIDWAIT F819 RTS2 F827 CLRED F827 CLRED F827 CLRED F827 CLRED F828 TASU F827 CLRED F828 TASU F827 CLRED F828 TASU F828 TASU F829 CLRED F828 TASU F829 CLRED F829 CLRED F820 CLR						

Monitor Symbol Table, Sorted by Address

?FEFD	READ	FFOA	RD2	FF16	RD3	?FF2D	PRERR
FF3A	BELL	FF3F	RESTORE	?FF44	RESTR1	?FF4A	SAVE
FF4C	SAV1	?FF59	OLDRST	FF65	MON	FF69	MONZ
FF73	NXTITM	FF7A	CHRSRCH	FF8A	DIG	FF90	NXTBIT
FF98	NXTBAS	FFA2	NXTBS2	FFA7	GETNUM	FFAD	NXTCHR
FFBE	TOSUB	FFC7	ZMODE	FFCC	CHRTBL	FFE3	SUBTEL
** SUG	CESSFUL	ASSEMBLY :=	NO ERRORS				
** ASS	SEMBLER (CREATED ON OS	5-JAN-82 00	00004			
** TO	TAL LINES	5 ASSEMBLED	1438				

** FREE SPACE PAGE COUNT 67 2 BJS. SRC2

80-Column Firmware Listing

0000:		2	*****	*****	****	****
0000:		з	*			
0000:			* Apple //e	VIDEO	FIRMWARE	
0000:		5	*			
0000:		6	* RICK AURIC	CHID C	8/81	
0000:		7	*			
0000:		8	* (C) 1981,	APPLE	COMPUTER	INC.
0000:		9	* ALL R	GHTS F	RESERVED	
0000:		10	*			
0000:		11	*******	*****	*******	****
0000:		12	*			
0000:	0006	13	GOODF8	EQU	6	;F8 ROM VERSION
0000:		14	*			
0000:		15	* HARDWARE E	QUATES	5:	
0000:		16	*			
0000:	0000		KBD	EQU	\$C000	;KEYBOARD PORT
0000:	0000		CLRBOCOL	EQU	\$C000	DISABLE BOCOL STORE
0000:	C001		SETBOCOL	EQU	\$C001	; ENABLE BOCOL STORE
0000:	C005		RDMAINRAM	EQU	\$C002	; READ MAINBOARD RAM
0000:	C003	21	RDCARDRAM	EQU	\$C003	; READ CARD RAM
0000:	C004	22	WRMAINRAM	EQU	\$C004	WRITE MAINBOARD RAM
0000:	C005		WRCARDRAM	EQU	\$C005	WRITE CARD RAM
0000:	C007		SETINTCXROM		\$C007	; SET INTERNAL CXOO RDM
0000:	C008	25	SETSTDZP	EQU	\$C008	SET STD ZP/STK
0000:	C009	26	SETALTZP	EQU	\$C007	; SET ALT ZP/STK
0000:	COOB	27	SETSLOTCORON	1 EQU	\$COOB	
0000:	COOC	28	CLRBOVID	EQU	\$C00C	DISABLE BOCOL VIDEO
0000:	COOD	29	SET80VID	EQU	\$C00D	; ENABLE BOCOL VIDED
0000:	COOE	30	CLRALTCHAR	EGU	\$C00E	; NORM LC, FLASH UC
0000:	COOF	31	SETALTCHAR	EQU	\$COOF	;NORM/INV LC, NO FLASH
0000:	C010	32	KBDSTRB	EQU	\$C010	CLEAR STROBE
0000:	CO11	33	RDLCBNK2	EQU	\$C011	READS LC BANK2
0000:	C012		RDLCRAM	EQU	\$C012	READS LC RAM ENABLE
0000:	C013	35	RDRAMRD	EQU	\$C013	READS RAMREAD STATE
0000:	C014	36	RDRAMWRT	EQU	\$C014	READS BANKWRT STATE
0000:	C018	37	RD80COL	EQU	\$C018	; READS SETBOCOL
0000:	C019	38	RDVBLBAR	EQU	\$C019	; 'VBL SIGNAL
0000:	CO1A	39	RDTEXT	EQU	\$C01A	; READS TXT MODE
0000:	COIC	40	RDPAGE2	EQU	\$CQ1C	; PAGE1/2 STATUS
0000:	CO1F	41	RDBOVID	EQU	\$C01F	READS SETBOVID
0000:	C030	42	SPKR	EQU	\$C030	; TOGGLE SPEAKER
0000:	C054	43	TXTPAGE1	EQU	\$C054	PAGE1 TEXT
0000:	C055	44	TXTPAGE2	EQU	\$C055	PAGE2 TEXT
0000:		45	*			
0000:		46	* MONITOR E	JUATES:		
0000:		47	*			
0000:	FBB3	48	F8VERSION	EQU	\$FBB3	FB ROM ID
0000:	FDOC	49	RDKEY	EQU	\$FDOC	; GET A KEYSTROKE
0000:	FE89		SETKBD	EQU	\$FE89	; IN#O
0000:	FE93	51	SETVID	EQU	\$FE93	; PR#0
0000:	FF58		IORTS	EQU	\$FF58	KNOWN RTS
0000:		54	* ZEROPAGE	EQUATES	5:	
0000:		55	*			
0000:		56		DSECT		
001F:	001F	57		ORG	\$1F	
001F:	0001	58	YSAV1	DS	1	SAFE PLACE IN ALL ENVIRONS
0020:	0001		WNDLFT	DS	1	SCROLLING WINDOW LEFT
0021:	0001	60	WNDWDTH	DS	1	SCROLLING WINDOW WIDTH
0022:	0001		WNDTOP	DS	1	SCROLLING WINDOW TOP
0023:	0001	62	WNDBTM	DS	1	SCROLLING WINDOW BOTTOM
0024:	0001		СН	DS	1	CURSOR HORIZONTAL
0025:	0001	64	CV	DS	1	CURSOR VERTICAL
0026:	0002	65		DS	2	; GBASL
0028:	0002	66	BASL	DS	2	; BASE ADDRESS
002A:	0029	67	BASH	EQU	BASL+1	
002A:	0002	68	BAS2L	DS	2	BASE ADDR FOR SCROLL
0020:	002B		BAS2H	EQU	BAS2L+1	
0032:	0032	70		ORG	\$32	
0032:	0001	71	INVFLG	DS	1	;>127=NORMAL
0033:	0003	72		DS	з	; N/A
0036:	0002	73	CSWL	DS	2	COUT HOOK
0038:	0037	74	CSWH	EQU	CSWL+1	
0038:	0002	75	KSWL	DS	2	;KEYIN HOOK
003A:	0039		KSWH	EQU	KSWL+1	
0030:	0030	77		ORG	·\$3C	
0030:	0002	78	AIL	DS	2	MONITOR TEMPS FOR MOVE

80-Column Firmware Listing

003E:		003D	79	A1H	EQU	A1L+1	
003E:		0002	80	A2L	DS	2	
0040:		003F	81	A2H	EQU	A2L+1 2	A3 NOT USED
0042:		0002		A4L	DS	2	
0044: 004E:		0043 004E	85	A4H	EQU	A4L+1 \$4E	
004E: 0050:		0002 004F		RNDL	DS EQU	2 RNDL+1	; RANDOM NUMBER SEED
0000:		004F	88		DEND		
0000:			90 91	* PERMANENT	DATA	IN SCREENHO	LES
0000:			92	* NOTE: THE * THE BO-C	SE RES	DE IN PAGE	1 OF
0000:			93 94	* THE BO-C	OLUMN	SCREEN PAIR SETS PAGE2	; ANY *MUST*
0000			95	* RESTORE	BACK T	O PAGE1 SO	THAT
0000:			96 97	* WE CAN C	ORRECT	LY ACCESS T	HESE TANCES
0000:			98	* IS ANY R	OUTINE	TO BE CALL BANKED IN!	ED WHILE
0000:			100	* WE HAVE I	PAGE2	BANKED IN!	
0000:		0478		TT THAT A	EQU	\$478 \$478+3	;A TEMP ;OLD CH SET FOR USER
0000:		047B 04FB	102	DLDCH MODE	EQU	\$4F8+3	OPERATING MODE
0000:			104	* MODE BITS			
0000			105	* 1	- ESC-	R ACTIVE	
0000:			107	* 0	- BASI	C PRINT	
0000:			109	* 0	- LANG	UAGE=BASIC	
0000:			110	* 1	- LANG	UAGE=PASCAL	DE
0000			112	* 1	- LITE	RAL UC/LC M	ODE
0000			113	*0	- GOTO	IXY N/A	FSS
0000:			115	*0	- NORM	AL VIDED (PASCAL)
0000			116	* 1	 INVE PASC 	RSE VIDED (PASCAL)
0000:			118	* 1.	- PASC	AL 1. 0 INTE	RFACE
0000:			119	MODE * MODE BITS *	– CALL	ER SEI'D (B ER CLI'D (B	ASIC) ASIC)
0000:			121	* 0	- NORM	AL MODE (PA	SCAL)
0000:		0080	122	* 1 · M.ESCR	- TRAN EQU	\$80	E (PASCAL) ; ESC-R ACTIVE
0000		0040	124	M. BINPUT	EQU	\$40 \$20	BASIC INPUTTING
0000		0020	125	M. PASCAL M. LIT M. GDXY M. VMODE	EQU	\$10	; PASCAL RUNNING ; LITERAL UC/LC INPUT
0000:		0008	127	M. GOXY	EQU	\$08 \$04	GOTOXY IN PROGRESS PASCAL VIDEO MODE PASCAL 1.0 MODE
0000:					EQU	\$02	PASCAL 1.0 MODE
0000:		0001	130	M. IRQ M. IRQ M. TRANS DURCH DURCV CHAR	EQU	\$01 \$01	;IRQ ENABLED (BASIC ONLY) ;TRANSPARENT MODE IF F/W PROTOCOL
0000:		057B	132	OURCH	EQU	\$578+3 \$5F8+3	; BO-COL CH ; CURSOR VERTICAL
0000:		05FB	133	DURCV CHAR	EQU	\$5F8+3 \$678+3	;CURSOR VERTICAL ;IN/OUT CHAR
0000:		06FB	135	XCOORD OLDBASL OLDBASH	EQU	\$6F8+3	; X-COORD (GOTOXY)
0000:		077B	136	OLDBASL	EQU	\$778+3 \$7F8+3	PASCAL SAVED BASL PASCAL SAVED BASH
0000:		0712	138	*			
0000:			139	* GENERAL S	CREEN	STUFF	
0000:		07F8	141	CBSLDT		\$7F8	; IRQ CE PROTOCOL
0000:			142 4		CHR	DE BEUNC	
C100:	NEXT	DBJECT C100		NAME IS VID	EO. OBJ ORG	0 \$C100	
C100:		C100	з	BFUNCPG	EQU	*	
C100: C100:		FD29 FBC1		FUNCEXIT F. BASCALC	EQU	\$FD29 \$FBC1	RETURN ADDRESS
C100:		FC22	6	F. VTAB F. VTABZ	EQU	\$FC22	
C100: C100:		FC24	7	F. VTABZ	EQU	\$FC24	
C100:			9	* BASIC FUN	CTION	HOOK:	
C100: C100:				* THIS ROUT	INE IS	CALLED BY	THE
C100:			12	* PATCHED	F8 ROM	1.	
C100: C100:			14	* FUNCTION	HERE	ALWAYS PER	TO THE
C100:			15				
C100: C100			16	* NOTE: F8 !	ROM DI	SABLES I/D	TO GET US
C100: C100:			18	* RUNNING	HERE.	WE RETURN T	D F8 SPACE.
C100			20	* INPUT: Y=			WS:
C100: C100:			21 22	상 북	0=CLR		
C100			23	*	2=SCR	OLL	
C100: C100:				*	3≂CLR 4=CLE		
C100:			26	*		T @ RESET	
C100: C100:			27			ESCAPE CHA	R

C100:	29 *	B≖SETWN	ND	
C100:	30 *			
C100: C100:	31 * S 32 * 1	DF BANK 8	HP FOR STATU & IRQ BIT	JS
C100:	33 * VOLATILE	AC, Y		
C100: C100:	34	WE HAVE	A CARD INS	
C100:	35 * NOTE: IF 36 * THEN US 37 * WE 'OWN	E THE VIE	DEG ROUTINES	B, SINCE
C100: C100:	37 * WE 'OWN 38 * IF NOT, 3	' SLOTS S	SCREENHOLES.	TANCO
C100:	39 * AND AVO	ID SLOT3	INTERFERENC	CE.
C100:	40			
C100: C100:	41 * VECTOR T 42 * TO AVOI	D AC DEST	ESCEIX IMMEI TRUCTION:	DIATELY
C100:	43 *			
C100: C100 C100:C0 06	44 B.FUNC 45	EQU #	* #6	IS IT KEYIN?
C102: DO 03 C107	46	BNE E	B. FUNCNK) NO
C104:4C 88 C2 C107: C107	47 48 B.FUNCNK	JMP EQU #	B. KEYIN	
C107: CO 07	49		#7	IS IT ESCAPE-FIX?
C109:D0 03 C10E C108:4C 6E C2	50 51	BNE B	B.FUNCNE B.ESCFIX	; ND ; =>YES !
C10E: C10E C10E: 78	52 B. FUNCNE		*	
C10E: 98 C10F: 48	53 54	TYA PHA		SAVE Y
C110:20 24 CB	55	JSR T	TESTCARD	DO WE HAVE A CARD?
C113: DO OA C11F C115:	56 57 *	BNE B	B. OLDFUNC	; =>N0
C115:	58 * NOTE: TH	IS TEST C	COULD TURN (דענ
C115:	59 * WRONG DI	N POWER-L	UP, SINCE TH	+Ε
C115: C115:	AL & SINCE T	HE MONITC	EFINED. HOWE DR IS DOING	Δ
C115:	62 * SIMPLE	SETWND'	CALL, WE WE	T' MC
C115: C115:	63 * GET INT	TROUBLE	CALL, WE WE E EVEN IF WE DECISION	E
C115:	65 *			
C115: AD FB 04 C118: 29 28	66 67		MODE	; IS MODE VALID? GDXY ; FOR BASIC
C118: 27 28 C11A: DO 03 C11F		BNE B	B. OLDFUNC	;=>DEFINITELY NOT! ;=>YES, GO NEW WAY
C11C:4C A4 C1 C11F:	69 70 *	JMP E	B. FUNCO	;=>YES, GO NEW WAY
C11F:	71 * NO CARD.	DO THING	S THE OLD .	AAY.
C11F:	72 *			
C11F: C11F C11F: 68	73 B. OLDFUNC 74	EQU *	*	
C120: AB	75	TAY		RESTORE Y
C121: A9 C1 C123: 48	76 77	LDA #		; TRANSFER VIA ; THE RTS-TRICK
C124: 89 EA CF	78			GET LO ADDRESS
C124. 57 EA CF				
C127:48	79	PHA		TRANSFER TO ROUTINE
C127:48 C128:60 C129:	79 80	PHA RTS		TRANSFER TO ROUTINE
C127: 48 C128: 60 C129: C129: A4 24	79 80	PHA RTS	сн	; TRANSFER TO ROUTINE ; ESC F IS CLR TO END OF PAGE
C127:48 C128:60 C129: C129:A4 24 C128:A5 25 C12D:48	79 80 81 82 F.CLREOP 83 84 CLEOP1	PHA RTS LDY C LDA C PHA	сн сv	ne was and out the set the still the set
C127:48 C128:40 C129: C129:44 24 C128:45 25 C12D:48 C12E:20 24 FC	79 80 81 82 F. CLREOP 83 84 CLEOP1 85	PHA RTS LDY C LDA C PHA JSR F	CH CV F. VTABZ	ne was and out the set the still the set
C127:48 C128:60 C129: C129:44 24 C128:45 25 C120:46 C126:20 24 FC C131:20 F4 C2 C134:40 00	79 80 81 82 F.CLREOP 83 84 CLEOP1 85 86 87	PHA RTS LDY C LDA C PHA JSR F JSR X LDY #	сн сv	ne was and out the set the still the set
C127:48 C128:60 C129:44 24 C129:A5 25 C120:48 C120:48 C122:20 24 FC C131:20 F4 C2 C134:60 00 C136:68	79 80 81 82 F. CLREOP 83 84 CLEOP1 85 86 87 88 88	PHA RTS LDY C LDA C PHA JSR F JSR X LDY # PLA	CH CV F. VTABZ X. CLEOLZ #\$00	ne was and out the set the still the set
C127:48 C128:60 C129:44 24 C129:54 25 C120:48 25 C120:48 C120:20 24 FC C131:20 F4 C2 C134:68 C137:65 C139:65 23	79 80 81 82 F.CLREOP 83 84 CLEOP1 85 86 87	PHA RTS LDY C LDA C PHA JSR F JSR F JSR X LDY # PLA ADC #	CH CV F. VTABZ X. CLEOLZ	ne was and out the set the still the set
C127:48 C128:60 C129:40 C129:44 24 C128:A5 25 C120:48 C128:20 24 FC C131:20 F4 C2 C134:40 00 C137:47 00 C137:47 00 C137:75 23 C138:70 F0 C128	79 80 81 82 F. CLREOP 83 84 CLEOP1 85 84 87 88 88 88 88 87 90 90 91	PHA RTS LDY C LDA C PHA JSR F JSR X LDY # PLA ADC # BCC C	CY CY CY CY CLEOLZ #\$00 WDBTM CLEOP 1	ne was and out the set the still the set
C127:48 C128:40 C129:4 24 C129:A 24 C129:A 24 C121:A 24 C121:A 24 FC C131:20 F4 FC C131:20 F4 C2 C134:A0 00 C134:46 C137:47 00 C137:47 00 C137:5 23 C138:90 F0 C12D C138:07 F0 C12D C138:20 F0 C12D C138:20 F0 C12D	79 80 81 82 F. CLREOP 83 84 CLEOP1 85 86 87 88 87 88 87 90 91 91 92 93	PHA RTS LDY C LDA C PHA C JSR F JSR X LDY # PLA ADC # ADC # BCC C JSR C	 CH SV X. CLEOLZ #\$00 #\$00 WNDBTM	; ESC F IS CLR TO END OF PAGE
C127:48 C128:60 C129:44 C129:44 C129:44 C128:45 25 C120:44 C128:20 24 FC C131:20 F4 C2 C134:40 00 C135:68 C137:67 00 C139:C5 23 C138:70 F0 C12D C130:20 22 FC C140:4C EB C2 C143:	79 80 81 82 F. CLREOP 83 84 CLEOP1 85 86 87 88 87 88 87 90 91 92 92 93 94	PHA RTS LDY C LDA C PHA JSR F JSR X LDY H PLA PLA PLA PLA BCC C JSR F JMP F	CH CV CV K. CLEOLZ #\$00 #\$00 MNDBTM CLEOP1 F. VTAB F. RETURN	; ESC F IS CLR TO END OF PAGE
C127:48 C128:40 C129:4 24 C129:A5 25 C120:44 24 C128:A5 25 C120:48 C128:20 24 FC C131:20 24 FC C131:20 24 FC C131:20 74 C2 C136:46 C137:47 00 C137:47 00 C137:47 00 C137:47 00 C138:47 FC C138:47 FC C140:44 E8 C2 C143:45 Z2	79 80 81 82 F. CLREOP 83 84 CLEOP1 85 86 87 88 87 88 87 90 91 91 92 93	PHA RTS LDY C LDA C PHA JSR F JSR F LDY # PLA ADC # BCC C JSR F JMP F LDA W	CH CV CV X. CLEOLZ #\$00 #NDBTM CLEOP1 - VTAB	; ESC F IS CLR TO END OF PAGE
C127:48 C128:40 C129:4 24 C129:A5 25 C129:44 24 C128:A5 25 C120:44 24 C128:A5 25 C120:48 C128:40 00 C131:20 24 FC C131:20 F4 C2 C134:64 C136:46 C136:47 00 C139:70 C120 C138:70 F0 C128:70 F0 C129:70 F0 F0 C129:70 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F0 F	79 80 81 82 F. CLREOP 83 84 CLEOP1 85 86 87 88 89 90 90 91 92 93 93 94 95 F. HOME 95 97	PHA RTS LDY C LDA C LDA C JSR F JSR F JSR F ADC # ADC # ADC # BCC # JSR F JMP F LDA W STA C LDY #	CH SV SV X. CLEOLZ #\$00 NNDBTM CLEOP1 VTAB VTAB VTAB VTAB VTAB VTAB VTAB VTABZ 	; ESC F IS CLR TO END OF PAGE
C127:48 C128:40 C129:44 C129:44 C128:45 25 C120:44 C128:20 24 FC C131:20 F4 C2 C131:20 F4 C2 C134:40 00 C137:47 00 C137:47 00 C139:75 C3 C139:76 F0 C139:76 F0 C139:76 F0 C139:20 22 FC C140:4C EB C2 C143:45 25 C147:40 00 C147:40 00 C147:40 00 C147:40 00 C147:40 00 C147:40 00 C147:40 00 C147:40 00 C147:40 00 C147:40 24	79 80 81 82 F. CLREOP 83 84 CLEOP 85 86 87 88 87 88 87 90 91 92 92 93 94 94 94 95 F. HOME 96 97 98	PHA RTS LDY C LDA C PHA JSR F JSR S ADC # ADC # ADC # BCC C JSR F JMP K STA C LDA C LDY # STY C	CH CV CV CV CU CV CV CV CV CV CV CV CV CV CV	; ESC F IS CLR TO END OF PAGE
C127:48 C128:60 C129:44 24 C128:A5 25 C120:44 24 C128:A5 25 C120:48 C121:20 F4 C2 C131:20 F4 C2 C134:A0 00 C137:47 00 C137:47 00 C137:47 00 C137:47 00 C138:70 F0 C120 C138:70 F0 C120 C138:70 F0 C120 C138:70 F0 C120 C140:45 22 C140:45 22 C145:85 25 C147:A0 00 C147:45 22 C147:85 26 C147:F0 E0 C120 C148:F0 E0 C120 C148:F0 E0 C120 C148:F0 E0 C120 C149:85 24 C148:F0 E0 C120 C149:F0 E0 C120 C140:F0 E0	79 80 81 82 F. CLREOP 83 CLEOP1 85 86 87 88 87 90 91 92 93 94 94 94 94 95 F. HOME 96 97 98 99 90	PHA RTS LDY C LDA C JSR Z LDY H JSR X LDY H PLA PLA PLA BCC C JSR F JSR F JSR F LDA M STA C BEQ C	CH CV CV CV CLEOLZ **00 **00 **00 CLEOP1 - VTAB - RETURN ANDTOP VV **00 CN CN LEOP1	; ESC F IS CLR TO END OF PAGE
C127:48 C128:40 C129:4 24 C129:A5 25 C129:A4 24 C128:A5 25 C120:44 24 C128:A5 25 C120:48 C128:A5 24 C131:20 74 C2 C131:20 74 C2 C131:20 74 C2 C136:50 70 C139:C5 23 C139:C5 23 C149:A5 22 C149:A5 22 C149:A5 22 C149:C5 C12D C149:C5 C12D C140:C5 C12D C	79 80 81 82 F. CLREOP 83 84 CLEOP1 85 86 87 88 88 89 90 90 91 91 92 93 94 95 F. HOME 95 95 95 95 95 95 95 95 95 95 95 96 90 90 91 91 92 93 93 94 95 95 95 95 96 96 97 90 90 90 90 90 90 90 90 90 90 90 90 90	PHA RTS LDY C LDA C DFHA JSR F JSR F PLA LDY H PLA ADC H BCC C C JSR F JSR F LDA M STC C LDA M STA C LDY H STA C LDA M	CH CV CV CV CU CV CV CV CV CV CV CV CV CV CV	; ESC F IS CLR TO END OF PAGE
C127:48 C128:40 C129:4 24 C129:A5 25 C129:A4 24 C128:A5 25 C120:44 24 C128:A5 25 C120:48 C128:A5 24 C131:20 F4 C2 C131:20 F4 C2 C136:70 F0 C139:C5 23 C139:C5 23 C139:C5 23 C139:C5 23 C139:C5 23 C149:45 22 C140:45 22 C140:45 22 C143:45 22 C143:45 22 C147:A0 00 C149:F6 C12D C149:F6 C12D C140:F6 C12D	79 80 81 82 F. CLREOP 83 84 CLEOP1 85 86 87 90 91 92 93 94 95 F. HOME 95 95 97 95 97 90 97 93 93 94 95 95 96 95 97 90 97 90 97 90 97 90 97 90 97 90 90 90 90 90 91 90 91 91 92 93 93 94 95 95 95 96 96 96 97 97 96 97 90 97 90 97 90 97 90 97 90 97 90 97 90 97 90 97 90 97 90 97 90 97 90 97 90 97 90 97 90 97 90 97 97 97 97 97 97 97 97 97 97 97 97 97	PHA RTS LDY C LDA C PHA C JSR F JSR F JSR X LDY # PLA K BCC C JSR C C JSR C STA C STA C STA C STA C LDA K PHA F JSR F	CH CV CV CV CV CLEDLZ H*GO H*GO H*GO CV CV H*GO CV CV H*GO CV CV H*GO CV CV CV CV CV CV CV CV CV CV CV CV CV	; ESC F IS CLR TO END OF PAGE
C127:48 C128:40 C129:44 24 C128:A5 25 C120:44 24 C128:A5 25 C121:48 C128:20 24 FC C131:20 F4 C2 C134:40 00 C137:47 00 C137:47 00 C137:47 00 C139:70 F0 C138:70 F0 C138:70 F0 C138:70 F0 C138:70 F0 C140:40 E C2 C140:40 E C2 C140:40 E C2 C140:45 22 C145:85 25 C147:40 00 C149:45 24 C148:F0 E0 C120 C149:45 22 C149:45 28 C149:45 2	79 80 81 82 F. CLREOP 83 84 CLEOP1 85 86 87 88 87 89 90 91 92 93 93 93 95 F. HOME 95 96 97 95 96 97 90 91 92 93 93 94 90 90 91 90 91 90 91 90 90 90 90 90 90 90 90 90 90 90 90 90	PHA RTS LDY C LDA C PHA JSR F JSR F JSR X LDY H PLA ADC H BCC CC JSR F JSR F LDA W STA C BEG C LDA W STY C BEG C	CH CV CV CV CV CV CV CV CV CV CV	; ESC F IS CLR TO END OF PAGE
C127:48 C128:40 C129:44 24 C129:A5 25 C129:44 24 C128:A5 25 C120:44 24 C128:20 24 FC C131:20 74 C2 C131:20 74 C2 C134:40 00 C137:47 00 C137:47 00 C137:47 00 C137:47 00 C137:47 00 C138:40 F0 C C12D C138:40 F0 C C12D C140:40 E8 C2 C140:40 E8 C2 C140:45 E8 C2 C143:45 22 C147:40 00 C149:45 22 C149:45 22 C149	79 80 81 82 F. CLREOP 83 84 CLEOP1 85 86 87 88 87 89 90 91 93 93 93 94 95 F. HOME 95 95 F. HOME 95 95 95 95 95 95 95 95 95 95 95 95 95	PHA RTS LDY C LDA C PHA JSR F JSR F JSR F LDY H ADC H ADC H BCC C JSR F LDA M STA C BEG C LDA M STY C BEG C LDA M STY C BEG C	CH CV CV CV CV CV CV CV CV CV CV	; ESC F IS CLR TO END OF PAGE
C127:48 C128:40 C129:4 24 C129:A5 25 C129:44 24 C128:A5 25 C120:44 24 C128:A5 25 C120:48 C128:42 4 C128:42 4 C131:20 F4 C2 C131:20 F4 C2 C136:45 C137:47 00 C139:C5 23 C139:C5 23 C139:C5 23 C139:C5 23 C139:C5 23 C139:C5 23 C139:C5 23 C139:C5 23 C139:C5 23 C149:45 22 C140:45 22 C140:45 22 C143:A5 22 C147:A0 00 C147:A0 00 C147:A0 00 C147:A0 00 C147:A5 22 C147:A5 22 C14	79 80 81 82 F. CLREOP 83 84 CLEOP1 85 86 87 90 91 92 93 F. HOME 95 95 F. HOME 97 95 97 90 97 93 90 90 91 00 57. CLREOP 88 89 90 91 92 93 93 94 93 94 95 90 95 95 90 97 95 96 90 97 97 98 90 90 97 90 97 90 97 98 90 90 97 90 90 90 90 90 90 90 90 90 90 90 90 90	PHA RTS LDY C LDA C PHA JSR F JSR F JSR F LDA H CMP H ADC H CMP H STA C LDY H STA B BEG C LDA M BEG C LDA M STA B STA B STA B	CH CV CV CV K CLEOLZ **00 **00 **00 **00 CV CV **00 CV CV **00 CV CV **00 CV CV **00 CV CV **00 CV CV **00 CV CV CV **00 CV CV CV CV CV CV CV CV CV CV CV CV CV	; ESC F IS CLR TO END OF PAGE
C127: 48 C128: 40 C129: 44 24 C129: A5 25 C129: A4 24 C128: A5 25 C129: 44 24 C128: A5 25 C129: 44 24 C128: 40 00 C131: 20 F4 C2 C134: 40 00 C139: 65 23 C139: 45 23 C140: 40 EB C2 C140: 40 EB C2 C140: 40 EB C2 C140: 45 22 C147: 40 00 C147: 49 C148: F0 E0 C12D C14D: 50 C149: 65 22 C147: 40 20 C149: 52 22 C147: 45 22 C157: 45 28 C157: 45 28	79 80 81 82 F. CLREOP 83 84 CLEOP1 85 86 87 90 91 92 92 93 F. HOME 95 95 F. HOME 97 95 F. HOME 99 97 93 90 90 91 91 92 93 93 94 95 95 96 95 96 97 97 98 97 99 97 90 97 98 97 90 97 98 97 98 97 90 97 93 97 95 96 90 97 95 96 97 97 98 97 97 98 90 97 97 98 90 97 90 97 97 98 90 97 90 97 97 98 90 90 97 97 97 98 90 97 90 90 97 90 97 90 97 90 97 90 97 90 97 90 97 90 97 90 97 90 97 90 90 97 90 90 97 90 90 90 90 90 90 90 90 90 90 90 90 90	PHA RTS LDY C LDA C PHA JSR F JSR F JSR F LDY H PLA ADC H CMP IA CMP IA STA C LDY H STA C LDA M STA BEG STA B STA B STA B STA B STA B STA B LDY H	CH CV CV CV CV CV CV CV CV CV CV	; ESC F IS CLR TO END OF PAGE
C127: 48 C128: 40 C129: 44 24 C129: A5 25 C129: 44 24 C128: A5 25 C120: 44 24 C128: A5 25 C120: 48 C128: 40 00 C131: 20 74 C2 C131: 20 74 C2 C131: 20 74 C2 C131: 20 74 C2 C131: 70 00 C139: 70 70 C139: 70 70 C149: 65 23 C140: 45 22 C143: 45 22 C143: 45 22 C147: 40 00 C149: 76 20 C149: 76 20 C149: 76 20 C149: 76 20 C149: 76 20 C149: 75 22 C149: 85 28 C153: 85 28 C1	79 80 81 82 F. CLREOP 83 84 CLEOP1 85 86 87 88 87 88 97 90 91 93 93 93 94 95 F. HOME 95 95 95 99 90 101 F. SCROLL 105 104 SCRL1 105 106 107	PHA RTS DDY CLDY CLDY CLDY CLDY CLDY CLDY CLDY C	CH CV CV K. CLEOLZ #\$00 MNDBTM LEOP1 F. VTAB F. RETURN MNDTOP CV H*500 CH F. VTAB E. VTAB F. VTAB F. VTAB SASL SASL SASL SASH MNDWDTH	; ESC F IS CLR TO END OF PAGE
C127:48 C128:40 C129:44 24 C129:A5 25 C129:44 24 C128:A5 25 C120:44 24 C128:40 00 C131:20 F4 C2 C131:20 F4 C2 C131:40 00 C139:47 00 C139:47 00 C139:47 00 C139:57 07 C12D C139:70 F0 C12D C139:70 F0 C140:46 E8 C2 C140:46 E8 C2 C143:45 22 C143:45 22 C145:45 22 C147:40 00 C149:45 22 C149:45 22 C159:45 27 C159:45 27 C159:45 21 C159:46 21 C159:46 21 C159:46 21 C149:67 22 C149:67 22 C149:67 22 C159:44 21 C159:46 21 C159:46 21 C149:67 22 C149:67 22 C149:67 22 C149:67 22 C159:46 21 C159:46 21 C159:46 21 C149:67 22 C149:67 22 C149:67 22 C149:67 22 C149:67 22 C149:67 22 C159:46 21 C159:46 21 C159:47 C1	79 80 81 82 F. CLREOP 83 84 CLEOP1 85 85 86 87 88 87 88 87 90 91 93 93 93 94 95 F. HOME 95 95 95 F. HOME 95 97 95 95 97 95 95 96 97 95 95 96 97 95 95 97 95 95 96 96 97 95 96 97 95 96 97 95 96 97 95 96 97 95 97 95 96 96 97 95 97 95 96 97 95 97 95 97 95 96 97 95 97 90 97 95 97 96 97 97 97 97 97 97 97 97 97 97 97 97 97	PHA RTS LDY CLDA C PHA JSR F JSR F JSR F LDY H PLA ADC H GCC C JSR F PLA ADC H STA C LDY H STA E BEG C LDA M STA E STA B STA B STA B STA B STA B STA STA	CH CV CV CV K CLEOLZ **00 **00 **00 **00 CV CV **00 CV CV **00 CV CV **00 CV CV **00 CV CV **00 CV CV **00 CV CV CV **00 CV CV CV CV CV CV CV CV CV CV CV CV CV	; ESC F IS CLR TO END OF PAGE
C127: 48 C128: 40 C129: 44 24 C129: A5 25 C129: 44 24 C128: A5 25 C129: 44 24 C128: A5 25 C129: 44 24 C128: 40 00 C136: 68 C137: 47 00 C139: 65 23 C139: 65 23 C149: 65 23 C149: 65 25 C140: 65 25 C140: 65 25 C147: A5 22 C147: A5 22 C	79 80 81 82 F. CLREOP 83 84 CLEOP1 85 86 87 90 91 92 93 94 94 95 94 96 95 96 97 97 98 96 97 90 97 97 98 96 97 97 98 90 97 97 97 98 90 97 97 97 97 97 97 97 97 97 97 97 97 97	PHA RTS LDY C LDA C PHA JSR F JSR F JSR F JSR F LDY H PLA ADC H CMP L BCC C C JSR F JSR F JSR F JSR F LDA L STA C LDY L STA C LDA L STA C LDA E STA C LDA E STA C LDA E STA C LDA E STA C STA C STA C LDA C STA S STA S	CH CV CV F. VTABZ K. CLEOLZ #\$00 \$ \$00 \$ SO CLEOP1 F. VTAB CV SV * *00 CV CV * *00 CV CV * *00 CV CV * *00 CV CV * *00 CV CV * *00 CV CV CV CV CV CV CV CV CV CV CV CV CV	; ESC F IS CLR TO END OF PAGE
C127: 48 C128: 40 C129: 44 24 C129: A5 25 C129: A4 24 C129: A5 25 C129: 44 24 C128: A5 25 C129: 44 24 C128: 45 25 C131: 20 F4 C2 C131: 20 F7 0 C131: 20 F5 23 C139: 65 23 C139: 65 23 C139: 65 23 C139: 65 23 C149: 45 22 C140: 46 EB C2 C140: 45 22 C143: 45 22 C143: 45 25 C147: A0 0 C147: A0 0 C147: 40 0 C147: 40 0 C147: 45 22 C147: 45 22 C155: 45 28 C155: 45 28 C155	79 80 81 82 F. CLREOP 83 84 CLEOP1 85 86 87 90 91 92 93 93 95 F. HOME 95 97 95 97 97 98 97 90 97 97 98 97 90 97 90 97 98 97 98 97 99 97 90 97 90 97 98 97 90 97 90 97 97 98 97 97 98 97 97 98 90 97 97 98 90 97 97 98 97 97 98 97 97 98 97 97 98 97 97 98 97 97 98 97 97 98 97 97 98 97 97 98 97 97 98 97 97 98 90 97 97 98 97 97 98 97 97 98 97 97 98 97 97 98 97 97 98 97 97 98 97 97 98 97 97 98 97 97 97 97 97 98 97 97 97 97 97 97 97 97 97 97 97 97 97	PHA RTS LDY CLDA C JSR F JSR F PLA JSR F PLA JSR F PLA K CMP & STA C LDA & STA C LDA & STA C LDA & STA C LDA & STA C LDA & STA S STA S STA S LDA & STA S STA S S STA S STA S S STA S S S S S S S S S S S S S S S S S S S	CH CV CV K CLEOLZ #\$00 W\$00 W\$00 ULEOP1 - VTAB - RETURN NDDTOP CV #\$00 CH CLEOP1 - VTAB - VTA	; ESC F IS CLR TO END OF PAGE
C127: 48 C128: 40 C129: 44 24 C129: A5 25 C129: A4 24 C129: A5 25 C129: 44 24 C128: A5 25 C129: 44 24 C128: 45 25 C131: 20 F4 C2 C131: 20 F7 0 C131: 20 F5 23 C139: 65 23 C139: 65 23 C139: 65 23 C139: 65 23 C149: 45 22 C140: 46 EB C2 C140: 45 22 C143: 45 22 C143: 45 25 C147: A0 0 C147: A0 0 C147: 40 0 C147: 40 0 C147: 45 22 C147: 45 22 C155: 45 28 C155: 45 28 C155	79 80 81 82 F. CLREOP 83 F. CLEOP1 84 CLEOP1 85 86 87 93 93 94 93 93 94 95 F. HOME 97 95 95 95 95 95 95 95 95 95 95 95 95 95	PHA RTS LDY C LDA C JSR F JSR F JSR F JSR F PLA F R STA C LDA K STA C LDA K STA C LDA K STA C LDA K STA C LDA K STA C LDA K STA C STA S STA S S STA S S S S S S S S S S S S S S S S S S S	CH CY CY CY CY CY CY CY CY CY CY CY CY CY	; ESC F IS CLR TO END OF PAGE
C127: 48 C128: 40 C129: 44 24 C128: A5 25 C129: 44 24 C128: A5 25 C121: 42 4 C128: 20 24 FC C131: 40 00 C133: 47 00 C137: 47 00 C138: 40 CE C2 C140: 40 EB C2 C140: 40 EB C2 C140: 40 EB C2 C140: 40 EB C2 C140: 45 22 C145: 45 25 C147: 40 00 C149: 45 22 C149: 45 22 C155: 45 26 C159: 45 27 C159: 4	79 80 81 82 F. CLREOP 83 84 CLEOP1 85 86 87 88 87 90 91 93 93 93 94 95 F. HOME 95 95 95 95 97 95 95 97 95 97 95 97 95 97 97 95 97 97 95 97 97 97 93 97 95 97 95 97 95 97 95 97 95 97 95 97 95 97 97 95 97 95 97 95 97 95 97 97 97 97 97 97 97 97 97 97 97 97 97	PHA RTS LDY C LDA C JSR F JSR F JSR F JSR F PLA F R STA C LDA K STA C LDA K STA C LDA K STA C LDA K STA C LDA K STA C LDA K STA C STA S STA S S STA S S S S S S S S S S S S S S S S S S S	CH CV CV CV CV CV CV CV CV CV CV	; ESC F IS CLR TO END OF PAGE
C127: 48 C128: 40 C129: 44 24 C129: A5 25 C129: A4 24 C129: A5 25 C129: 44 24 C128: A5 25 C129: 44 24 C128: 45 25 C131: 20 F4 C2 C131: 20 F7 0 C131: 20 F5 23 C139: 65 23 C139: 65 23 C139: 65 23 C139: 65 23 C149: 45 22 C140: 46 EB C2 C140: 45 22 C143: 45 22 C143: 45 25 C147: A0 0 C147: A0 0 C147: 40 0 C147: 40 0 C147: 45 22 C147: 45 22 C155: 45 28 C155: 45 28 C155	79 80 81 82 F. CLREOP 83 F. CLEOP1 84 CLEOP1 85 86 87 93 90 90 93 93 94 93 93 94 95 F. HOME 97 95 95 97 100 95 97 100 95 97 100 95 97 100 97 95 97 100 97 95 97 100 95 97 100 97 95 97 100 97 98 97 100 97 98 97 100 97 98 97 100 97 98 97 100 97 98 97 100 97 98 97 100 97 98 97 100 97 98 97 100 97 98 97 100 97 98 97 100 97 98 97 100 97 98 97 100 97 98 97 100 97 98 97 100 97 98 97 100 97 98 97 100 97 97 98 97 100 97 97 98 97 100 97 97 98 97 100 97 97 98 97 100 97 97 98 97 100 97 98 97 100 97 97 97 98 97 100 97 97 98 97 100 97 97 98 100 97 97 97 97 97 97 97 97 97 97 97 97 97	PHA RTS JSR F JSR F JSR F JSR F JSR F PLA MDC H CCMP W STA C UDY H STA B STA B LDA W STA B LDA W STA B LDA W STA B LDA B STA B LDA B STA STA C CMP W STA STA STA C LDA STA	CH CY CY CY CY CY CY CY CY CY CY CY CY CY	; ESC F IS CLR TO END OF PAGE

80-Column Firmware Listing

C170:30 E1 C153 C172:A0 00 C174:20 F4 C2 C177:20 22 FC C177:40 E8 C2 C170:A4 24 C181:91 28 C183:C8 C184:C4 21 C184:C4 21 C184:C4 21 C184:C7 C181 C186:T0 F7 C181 C186:T0 T C181	121 BCRL3 122 123 124 125 F. CLREOL 126 127 CLEOL2 128 128 129 130 131	B MIY J B B M D A MDA L D R B C B C B C B C B C B C B C B C B C B	SCRL1 #800 X.CLEOLZ F.VTAB F.RETURN CH #84A0 (8ASL),Y WNDHDTH CLEOL2 F.GORET	;->DONE ; DONE (ALWAYS TAKEN)
C18A: C18A: C18A	132 133 F. SETWND	EQU	*	
C18A: A9 28 C18C: 85 21	134 135	LDA	#40 WNDWDTH	
C18E: A9 18 C190: 85 23	136	LDA	#24 WNDBTM	
C192: A9 17 C194: B5 25	138	LDA	#23	
C194:83 23 C196:20 22 FC C199:4C EB C2	139 140	STA	CV F. VTAB	
C177:4C EB C2 C17C:	141 142	JMP	F. RETURN	
C19C: C19C C19C: A4 1F	143 F. CLEDLZ 144	EQU	* YSAV1	- BESTORE HOBIT BOSTTION
C19E: 20 F4 C2 C1A1: 4C EB C2	145	JSR	X. CLEOLZ	RESTORE HORIZ POSITION
C1A4: C1A4	146 F. GORET 148 B. FUNCO	JMP	F. RETURN #	; DONE
C1A4: 68 C1A5: A8	149 150	PLA TAY		RESTORE Y
C1A6: C1A6:	151 * 152 * SET IRGM			
C1A6:	153 *			
C1A6:AD FB 04 C1A9:29 FE	154 155	L DA AND	MODE #255-M. IRG	ASSUME IRG IS DISABLED
C1AB: BD FB 04 C1AE: 68	156 157	STA	MODE	PULL CXBANK STATUS
CIAF: 80 78 04 C182: 68	158 159	STA PLA	TEMP 1	PULL CXBANK STATUS FOFF STACK FOFT USER'S PSTATUS
C183: 48 C184: 4A	160	PHA	A	(LEAVE ALONE ON STACK)
C185: 4A	162	LSR	A	THE CARRY
C1B6:4A C1B7:AD 78 04	163 164	LDA	A TEMP1	PUT CABANK STATUS
C18A: 48 C18B: BO OB C1C5	165 166	PHA BCS	NOI	; BACK ON STACK ; =>HE'S INHIBITED
C18D: AD FB 04 C1C0: 07 01	167	LDA	MODE	J-VIL 3 MAIDTIEN
C1C2: 8D FB 04	169	ORA STA	#M. IRQ MODE	
C1C5: C1C5 C1C5: A5 25	170 NDI 171	EQU	* CV	COPY USER CV
C1C7: 8D FB 05 C1CA: 4C FF C1	172 173	STA JMP	OURCY B. VECTOR	; TO DURS ; CONTINUE
C1CD: C1CD:	174 *			
C1CD: C1CD:	175 * NOTE: TH 176 * ALL IN 177 *	THE CIO	O PAGE	INES
C1CD:	178 #			
C1CD: C1CD C1CD: 20 A4 CC	179 B. SCROLL 180	EQU	* SCROLLUP	DO IT FOR CALLER
C1DO: 4C EB C2 C1D3	181 182 *	JMP	F. RETURN	AND RETURN DIRECTLY
C1D3: C1D3 C1D3: 20 48 CD	183 B. CLREOL 184	EQU	* X. GS	CLEAR TO EOL
C1D6: 4C EB C2 C1D7:	185	JMP	F. RETURN	RETURN DIRECTLY TO CALLER
C1D7: C1D7	186	EQU	*	
C1D9:A4 1F C1DB:20 4E CD	188 189	LDY JSR	YSAV1 X. GSEOLZ	RESTORE HORIZ POSITION
C1DE: 4C EB C2 C1E1:	190	JMP	F. RETURN	
CIE1: CIE1 CIE1: 20 23 CD	192 B. CLREOP 193	EQU	* x. vt	
C1E4: 4C EB C2	194	JMP	F. RETURN	CLEAR TO EDB RETURN DIRECTLY TO CALLER
C1E7: C1E7: 4C 19 C2 C1EA: 4C 34 C2	195 196 B. SETWND	JMP	B. SETWNDX	
C1EA: 4C 34 C2 C1ED:	197 B. RESET 198	JMP	B. RESETX	MUST BE IN BFUNC PAGE
C1ED: C1ED C1ED: 20 42 CD	199 B. HOME 200	EQU	* X.FF	HOME & CLEAR
C1FO: AD 78 05 C1F3: 85 24	201 202	LDA	OURCH	COPY CH/CV FOR CALLER
C1F5:8D 7B 04	203	STA	OLDCH	REMEMBER WHAT WE SET
C1FB: AD FB 05 C1FB: 85 25	204 205	LDA STA	OURCV CV	
C1FD: 10 2F C22E C1FF:	206 207 *	BPL	GOBACK	; (ALWAYS TAKEN)
C1FF: C1FF:	208 * COPY USE 209 * FROM QU	R'S CUR	SOR IF IT DI	FFERS
C1FF: C1FF:	210 * IN 80-C	OLUMN M	ODE). IF WE (ARE
GAFF.	ali * NUI (N	ov-nuné	THEN ALWAYS	5 UGE

C1FF:						
	2	212	* THE USER	S CH V	ALUE SINCE C	URS
C1FF:		213		BLY IN	ALID.	
C1FF:		214				
C1FF:	C1FF 2		B. VECTOR	LOU	* BASCALC	
C1FF: 20 51 C202: A5 24		216		LDA		GET USER CH VALUE
C202: A5 24		218		BIT		DISPLAYING BO-COLS?
C207:10 05	C20E 2	217		BPL	B. GETCH	STAREATING DUCCESS
C209: CD 7B	04 2	220		CMP	OLDCH	; =>ND, USER CH IS IT ; IS IT DIFFERENT?
C20C: FO 03		221		BEG	B. FUNC1	=>ND, USE OURS
C20E:		222	B. GETCH	EQU	*	
C20E: 80 78	05 3	223		STA	OURCH	; USE HIS CH
C211:			B. FUNCI	EQU	*	
C211: A9 C1		225		LDA	# <bfuncpg< td=""><td>TRANSFER TO ROUTINE</td></bfuncpg<>	TRANSFER TO ROUTINE
C213: 48		226		PHA		; VIA RTS-TRICK
C214: 89 F3		227		LDA	B. TABLE, Y	GET LO ADDRESS
C217:48		228		PHA		
C218: 50 C219:		229		RTS		
C219:		230	B. SETWNDX	EQU	*	
C219: A9 50		232	B. SE IWNUX	LDA		ASSUME BO-COLS
C218:2C 1F	0 5	233		BIT	RDBOVID	WHICH MODE?
C21E: 30 01	C221 2	234		BMI	B. SETWND2	;=>IT'S 80
C220: 4A		235		LSR	A	MAKE IT 40
C221			B. SETWND2	EQU	*	
C221:85 21		237		STA	WNDWDTH	
C223: A9 18		238		LDA	#24	SET BOTTOM
C225: 85 23	i	239		STA	WNDBTM	
C227: A9 17		240		LDA		VTAB TO BOTTOM
C229:80 FB		241		STA	OURCV	
C22C:85 25		242		STA	CV	
C22E: 20 51			GOBACK	JSR	BASCALC	
		244		JMP	F. RETURN	
C234:	4	245	*			
C234:			HANDLE RES	SET FOR	MONITOR:	
C234:					*	
C234:		248 1	B. RESETX	EQU		DESTROY MODE BYTE
C234: A9 FF C236: 80 FB		250		STA	MODE	DESTRUT MUDE BYTE
C239: AD 5D	04	250		LDA		SETUP
C23C: AD 5F	CO 2	252		LDA		ANNUNCIATORS
C23F:	- CQ	253		LDM	\$COOF	, ANNONCIATORS
C23F:		254			E KEY	
C23F:		255	ALIAS PA	ADDLE F	UTTONS () IS	
C23F:		256	+ DEPRESSEL	, COLL	START THE SY	STEM
C23F:		257 .			G MEMORY:	
C23F:	2	258	*			
C23F: AD 62		257		LDA		GET BUTTON ((SOLID)
C242: 30 1D	C261 2	260		BMI	DIAGS	;=>DOWN, DO DIAGS
C244: AD 61	CO 2	261			\$C061	
				LDA		GET BUTTON O (OPEN)
C247:10 1B	C264 a	262		BPL		GET BUTTON O (OPEN)
C247:10 18 C247:	C264 a	263		BPL.	RESETRET	
C247:10 1B C249: C249:	C264 a	263	* BLAST 2 BY	BPL OF	RESETRET	
C247:10 1B C249: C249: C249:	C264 a	263 · 264 · 265 ·	* BLAST 2 BY * INCLUDING	BPL OF	RESETRET	
C247:10 18 C249: C249: C249: C249: C249:	C264	263 · 264 · 265 · 266 ·	* BLAST 2 BY * INCLUDING	BPL TES OF	RESETRET F EACH PAGE, RESET VECTOR:	;=>NOT JIVE OR DIAGS
C247: 10 18 C249: C249: C249: C249: C249: C249: C249: A0 B0	C264	263 · 264 · 265 · 266 ·	* BLAST 2 BY * INCLUDING	BPL TES OF THE F	RESETRET F EACH PAGE, RESET VECTOR: #\$B0	
C247:10 18 C249: C249: C249: C249: C249: C249: A0 B0 C248:A9 00	C264	263 · 264 · 265 · 266 · 267 · 268 ·	* BLAST 2 BY * INCLUDING	BPL TES OF THE F LDY LDA	RESETRET EACH PAGE, RESET VECTOR: #\$B0 #0	;=>NOT JIVE OR DIAGS
C247: 10 18 C249: C249: C249: C249: C249: C249: C249: A0 B0	C264	263 · 264 · 265 · 266 ·	* BLAST 2 BY * INCLUDING	BPL TES OF THE F	RESETRET EACH PAGE, RESET VECTOR: #\$BO #0 A1L	;=>NOT JIVE OR DIAGS
C247:10 1B C249: C249: C249: C249: C249: C249:A0 B0 C248:A9 00 C240:85 3C C244:A9 BF	C264	263 264 265 266 267 268 268 269	* BLAST 2 BY * INCLUDING	BPL TES OF THE F LDY LDA STA	RESETRET F EACH PAGE, RESET VECTOR: #5BO #0 A1L. #5BF	;=>NOT JIVE OR DIAGS ;LET IT PRECESS DOWN
C247:10 1B C249: C249: C249: C249: C249: C249: C249:A0 B0 C24B:A9 00 C24B:A9 00 C24D:B5 3C C24F:A9 BF C251:3B	C264	263 264 265 266 267 268 267 268 269 270 271 272	* BLAST 2 BY * INCLUDING	BPL TES OF THE F LDY LDA STA LDA	RESETRET F EACH PAGE, RESET VECTOR: #\$BO #0 A1L. #\$BF #	;=>NGT JIVE OR DIAGS ;LET IT PRECESS DOWN ;START FROM BFXX DOWN
C247:10 18 C249: C249: C249: C249: C249: C249:A0 B0 C248:A9 00 C240:85 3C C247:A9 BF C251:38 C252: C252:B5 3D	C264 C252	263 264 265 266 267 268 267 268 267 270 271 272 272 272	* BLAST 2 B) * INCLUDING *	BPL THE OF THE F LDY LDA STA LDA SEC EQU STA	RESETRET F EACH PAGE, RESET VECTOR: #8B0 #0 A1L #8BF # A1H	;=>NGT JIVE OR DIAGS ;LET IT PRECESS DOWN ;START FROM BFXX DOWN
C247: 10 18 C249: C249: C249: C249: C249: C249: A0 80 C248: A9 85 C246: A9 85 C251: 38 C252: C252: 85 3D C254: 71 3C	C264 C252	263 264 265 266 267 268 267 270 271 272 271 272 273 273	* BLAST 2 B) * INCLUDING *	BPL THE OF LDY LDA STA LDA SEC EQU STA STA	RESETRET F EACH PAGE, RESET VECTOR: #\$BO #0 A1L. #\$BF #	;=>NOT JIVE OR DIAGS ;LET IT PRECESS DOWN ;START FROM BEXX DOWN
C247:10 18 C249: C249: C249: C249: C249: C249:A0 B0 C240:B5 3C C240:B5 3C C240:A9 BF C251:3B C252:B5 3D C252:B5 3D C254:91 3D C254:91 3D	C264 C252	263 264 265 266 267 268 267 270 271 272 273 273 274 275	* BLAST 2 B) * INCLUDING *	BPL THES OF LDY LDA STA LDA SEC EQU STA STA DEY	RESETRET F EACH PAGE, RESET VECTOR: #5B0 #0 A1L #5BF # A1H (A1L),Y	;=>NOT JIVE OR DIAGS ;LET IT PRECESS DOWN ;START FROM BEXX DOWN
C247:10 1B C249: C249: C249: C249: C249: C249:A0 B0 C248:A9 00 C248:A9 00 C244:A9 00 C244:A9 00 C244:A9 00 C244:A9 00 C244:A9 00 C252: C252:B5 3D C255:B5 3D C255:B5 3D C255:91 3C	C264	263 · 264 · 265 · 266 · 267 · 268 · 271 · 272 · 273 · 273 · 275 · 276 ·	* BLAST 2 B) * INCLUDING *	BPL TTES OF THE F LDY LDA STA LDA STA EQU STA DEY STA	RESETRET F EACH PAGE, RESET VECTOR: #00 A1L #9BF # A1H (A1L),Y (A1L),Y	;=>NOT JIVE OR DIAGS ;LET IT PRECESS DOWN ;START FROM BFXX DOWN ;FOR SUBTRACT
C247:10 18 C249: C249: C249: C249: C249:A9 C249:A9 00 C248:A9 00 C248:A9 00 C248:A9 00 C248:A9 00 C248:A9 00 C251:38 C252:65 3D C254:91 3C C254:91 3C C255:91 3C C257:91 3C	C264 C252	263 - 264 - 265 - 266 - 268 - 271 - 272 - 271 - 272 - 273 - 275 - 276 - 277 -	* BLAST 2 B) * INCLUDING *	BPL (TES OF THE F LDY LDA STA SEC EQU STA STA SEC STA SBC	RESETRET FEACH PAGE, RESET VECTOR: #80 A1L. #88F * A1H (A1L),Y (A1L),Y #1	;=>NOT JIVE OR DIAGS ;LET IT PRECESS DOWN ;START FROM BEXX DOWN ;FOR SUBTRACT ;BACK DOWN TO NEXT PAGE
C247: 10 18 C249: C249: C249: C249: C249: A0 80 C248: A9 00 C248: A9 00 C248: A9 00 C248: A9 00 C248: A9 00 C248: A9 00 C252: 138 C252: 138 C252: 138 C252: 138 C252: 10 00 C254: 19 00 C255: 19 10 C255: 10 10 C2	C264	263 264 265 266 267 268 267 270 271 272 273 274 275 276 277 278	* BLAST 2 B) * INCLUDING *	BPL (TES OF THE F LDY LDA STA LDA STA SEC EQU STA DEY STA DEY STA CMP	RESETRET F EACH PAGE, HESET VECTOR: #50 A1L #55 # A1H (A1L),Y (A1L),Y #1	;=>NOT JIVE OR DIAGS ;LET IT PRECESS DOWN ;START FROM BFXX DOWN ;FOR SUBTRACT
C247:10 18 C249: C249: C249: C249: C249: C249:A0 B0 C241:A9 BF C251:38 C252:45:49 BF C251:38 C252:45 30 C254:49 BF C255:49 BC C257:41 3C C259:47 10 C C258:C9 01 C258:C9 01 C258:C9 01	C264 C252 C252	263 264 265 266 267 268 267 271 272 273 274 275 276 277 278 277 278 279	* BLAST 2 B) * INCLUDING *	BPL TTES OF THE F LDY LDA STA LDA SEC EQU STA DEY STA STA STA STA STA STA STA STA	RESETRET F EACH PAGE, RESET VECTOR: #\$B0 AlL *\$BF * AlH (ALL),Y %1 %1 #1 BLAST	;=>NOT JIVE OR DIAGS ;LET IT PRECESS DOWN ;START FROM BEXX DOWN ;FOR SUBTRACT ;BACK DOWN TO NEXT PAGE ;STAY AWAY FROM STACK!
C247:10 18 C249: C249: C249: C249: C249: A0 BO C248: A9 BO C240: B5 3C C240: B5 3C C251: 38 C252: 85 3D C255: 85 3D C255: 91 3C C257: 91 3C C258: 69 C259: 67 01 3C C259: 67 01 3C C259: 67 01 3C	C264 C252 C252 C254	243 244 245 2467 2467 2468 2467 271 272 273 2774 2775 2776 2778 2779 278 279 278	* BLAST 2 B) * INCLUDING *	BPL (TES OF THE F LDY LDA STA LDA STA SEC EQU STA DEY STA DEY STA CMP	RESETRET F EACH PAGE, RESET VECTOR: #\$B0 AlL *\$BF * AlH (ALL),Y %1 %1 #1 BLAST	;=>NOT JIVE OR DIAGS ;LET IT PRECESS DOWN ;START FROM BEXX DOWN ;FOR SUBTRACT ;BACK DOWN TO NEXT PAGE
C247:10 18 C249: C249: C249: C249: C249: C249: C249: A0 80 C248: A9 00 C248: A9 00 C248: A9 85 C251: 38 C252: 85 30 C254: 49 85 C252: 85 30 C254: 49 30 C259: 57 01 C259: 57 01 C259: 50 0 F3 C256: F0 03 C256: F0 03	C264 C252 C252 C254	263 264 265 266 266 271 273 273 273 275 275 275 277 275 277 275 277 275 277 275 277 275 277 275 277 275 277 275 277 275 277 275 277	* BLAST 2 B' INCLUDING BLAST	BPL THES OF THE F LDY LDA STA LDA STA STA STA STA STA STA STA ST	RESETRET F EACH PAGE, RESET VECTOR: #\$B0 AlL *\$BF * AlH (ALL),Y %1 %1 #1 BLAST	;=>NOT JIVE OR DIAGS ;LET IT PRECESS DOWN ;START FROM BEXX DOWN ;FOR SUBTRACT ;BACK DOWN TO NEXT PAGE ;STAY AWAY FROM STACK!
C247:10 18 C249: C249: C249: C249: C249: A0 BO C240:85 3C C240:85 3C C251:38 C252:85 3D C252:85 3D C255:49 1 3C C255:49 1 3C C255:49 1 3C C257:91 3C C259:47 1 3C C259:57 1 3C C259:57 1 3C C259:57 50 1 3C C250:57 50 50 50 50 50 50 50 50 50 50 50 50 50	C264 C252 C252 C264 C264	243 2245 2245 2245 2245 2245 2246 2270 2272 2773 2775 2775 2775 2775 2775 2775	* BLAST 2 B) * INCLUDING *	BPL THE F LDY LDA STA LDA STA LDA STA SEC EQU STA SEC STA DEY STA BNE BEQ EQU EQU	RESETRET F EACH PAGE, RESET VECTOR: *0 A1L *0 A1H (A1L),Y (A1L),Y *1 BLAST RESETRET *	:=>NOT JIVE OR DIAGS :LET IT PRECESS DOWN :START FROM BEXX DOWN :FOR SUBTRACT :BACK DOWN TO NEXT PAGE :STAY AWAY FROM STACK! :(ALWAYS)
C247:10 18 C249: C249: C249: C249: C249: C249: C249: B5 3C C248: A9 00 C248: A9 00 C248: A9 00 C248: A9 00 C248: A9 00 C252: C252: C252: C1 30 C254: A9 10 C255: A9 11 C255: C1 11 C255: C1 01 C255: C1 01 C255: C1 01 C256: C	C264 C252 C252 C254 C264 C261	2223 - 2224 2224 2225 - 2225 2226 2227 2227 2273 2273 2274 2273 2274 2275 2275 2275 2275 2275 2275 2275	* BLAST 2 B' INCLUDING BLAST	BPL THES OF THE F LDY LDA STA LDA STA STA STA STA STA STA STA ST	RESETRET F EACH PAGE, RESET VECTOR: *0 A1L *0 A1H (A1L),Y (A1L),Y *1 BLAST RESETRET *	;=>NOT JIVE OR DIAGS ;LET IT PRECESS DOWN ;START FROM BEXX DOWN ;FOR SUBTRACT ;BACK DOWN TO NEXT PAGE ;STAY AWAY FROM STACK!
C247:10 18 C249: C249: C249: C249: C249: A0 BO C240:85 3C C240:85 3C C251:38 C252:85 3D C252:85 3D C255:49 1 3C C255:49 1 3C C255:49 1 3C C257:91 3C C259:47 1 3C C259:57 1 3C C259:57 1 3C C259:57 50 1 3C C250:57 50 50 50 50 50 50 50 50 50 50 50 50 50	C264 C252 C252 C264 C261 C4	2223 · 2224 · 2224 · 2224 · 2224 · 2224 · 2224 · 2224 · 2224 · 2224 · 2273 · 2274 · 2275 · 22	* BLAST 2 B' INCLUDING * BLAST DIAOS *	BPL THE F LDY LDA STA LDA STA LDA STA SEC EQU STA SEC STA DEY STA BNE BEQ EQU EQU	RESETRET F EACH PAGE, RESET VECTOR: *0 A1L *0 A1H (A1L),Y (A1L),Y *1 BLAST RESETRET *	:=>NOT JIVE OR DIAGS :LET IT PRECESS DOWN :START FROM BEXX DOWN :FOR SUBTRACT :BACK DOWN TO NEXT PAGE :STAY AWAY FROM STACK! :(ALWAYS)
C247:10 18 C249: C249: C249: C249: C249: C249: A0 80 C248: A9 00 C248: A9 85 C252: C252: B3 C252: B3 C252: B3 C252: B3 C254: F1 3C C254: F1 3C C257: F7 0 3 C257: F7 0 3 C257: F0 03 C254: C261: C261: C261: C264: C264: C264: C264:	C264 C252 C252 C254 C264 C261 C4 C264	2223 · 2224 · 2224 · 2224 · 2224 · 2224 · 2224 · 2224 · 2224 · 2224 · 2273 · 2274 · 2275 · 22	* BLAST 2 B' INCLUDING BLAST	BPL TTES OF THE F LDA LDA STA LDA STA LDA SEC EQU STA DEY STA DEY STA DEY STA DEY STA EQU JMP	RESETRET F EACH PAGE, RESET VECTOR: #880 AlL #88F * A1H (AlL),Y ((AlL),Y * BLAST RESETRET * \$C5401 *	:=>NOT JIVE OR DIAGS :LET IT PRECESS DOWN :START FROM BEXX DOWN :FOR SUBTRACT :BACK DOWN TO NEXT PAGE :STAY AWAY FROM STACK! :(ALWAYS)
C247:10 18 C249: C249: C249: C249: C249: C249: A B C C249: C240: B 5 3C C240: B 5 3C C252: B 5 3D C252: B 5 3D C252: B 7 1 3C C255: B 9 C255: B 9 C255: B 9 C255: C 9 01 C258: C 9 01 C258: C 9 01 C258: C 9 03 C261: C 0 33 C261: C 0 33 C264: C 0 24 C 264: C 0 24 C 265: C 0 24 C 265: C 0 10 C 255: C 0 10 C 256: C 0 10 C 0 10	C264 C252 C264 C264 C264 C264	2223 - 22	* BLAST 2 B' INCLUDING * BLAST DIAOS *	BPL THE F LDY LDA STA SEC EQU STA SBC STA SBC CMP BNE BEQ EQU JMP EQU	RESETRET F ACH PAGE, IESET VECTOR: #580 #0 All #585 * AlH (AlL),Y #1 BLAST RESETRET * \$CA01 * TESTCARD	;=>NOT JIVE OR DIAGS ;LET IT PRECESS DOWN ;START FROM BEXX DOWN ;FOR SUBTRACT ;BACK DOWN TO NEXT PAGE ;STAY AWAY FROM STACK! ;(ALWAYS) ;RUN DIAGS
C247:10 18 C249: C249: C249: C249: C249: C249: A0 80 C248: A9 00 C248: A9 85 C252: C252: B3 C252: B3 C252: B3 C252: B3 C254: F1 3C C254: F1 3C C257: F7 0 3 C257: F7 0 3 C257: F0 03 C254: C261: C261: C261: C264: C264: C264: C264:	C252 C252 C252 C254 C264 C4 C261 C4 C261 C27D C27D C27D	2263 · 2264 · 2265 · 2266 · 2267 · 2268 · 2267 · 2268 · 2277 · 2273 · 2276 · 2277 · 2278 · 2276 · 2278 · 2278 · 2280 · 2281 · 2282 · 2283 · 2288 · 22	* BLAST 2 B' INCLUDING * BLAST DIAOS *	BPL (TES OF THE F LDA STA STA STA SEQU STA STA STA STA STA STA STA STA	RESETRET F ACH PAGE, RESET VECTOR: #580 #0 AlL #58F * AlH (ALL),Y (ALL),Y (ALL),Y (ALL),Y (ALL),Y (ALL),Y * #1 BLAST RESETRET * # \$ \$CA01 * TESTCARD CORETN	<pre>;=>NOT JIVE OR DIAGS ;LET IT PRECESS DOWN ;START FROM BFXX DOWN ;FOR SUBTRACT ;BACK DOWN TO NEXT PAGE ;STAY AWAY FROM STACK? ;(ALWAYS) ;RUN DIAGS ;CARD PLUGGED IN? ;=>YES</pre>
C247:10 18 C249: C249: C249: C249: C249: C249: C249: A0 B0 C24B: A9 B5 C251:3B C252:85 3D C252:85 3D C254:47 13C C254:47 13C C254:47 13C C254:47 13C C254:47 13C C254:57 133 C254:57 133 C254:57 133 C254:57 14 C254:57 14 C255 14 C	C264 C252 C252 C264 C4 C4 C270 C270 C270 C270 C270 C270	2233 - 2254 - 2254 - 2255 - 2257 - 2275 - 22	* BLAST 2 B' INCLUDING * BLAST DIAOS *	BPL TTES OF THE F LDY LDA STA STA STA DEY STA STA DEY STA BNE BEQ EQU JSR BEQ	RESETRET F ACH PAGE, IESET VECTOR: #580 #0 All #585 * AlH (AlL),Y #1 BLAST RESETRET * \$CAO1 * TESTCARD GORETN SETSLOTCORD	:=>NOT JIVE OR DIAGS :LET IT PRECESS DOWN :START FROM BEXX DOWN :FOR SUBTRACT :BACK DOWN TO NEXT PAGE :STAY AWAY FROM STACK! :(ALWAYS) :RUN DIAGS :CARD PLUGGED IN?
C247:10 18 C249: C249: C249: C249: C249: C249: C249:A0 B0 C248:A9 00 C248:A9 00 C248:A9 00 C252:B5 3D C252:B5 3D C252:B5 3D C252:B7 01 C258:C9 01 C9 01 C1 C258:C9 01 C258:C9 01	C252 C252 C252 C264 C4 C264 C8 C270 C0 C270 C0 C270	2263 - 2264 - 2265 - 2267 - 2268 - 2267 - 2268 - 2271 - 2272 - 2273 - 2273 - 2275 - 2277 - 2275 - 2277 - 2277 - 2277 - 2277 - 2277 - 2280	* BLAST 2 B' INCLUDING * BLAST DIAOS *	BPL (TES OF THE F LDY LDA STA LDA SEC STA DEY STA DEY SBC CMP BEQ EQU JMP EQU JSR BEQ STA	RESETRET F ACH PAGE, IESET VECTOR: #580 #0 All #585 * AlH (AlL),Y #1 BLAST RESETRET * \$CAO1 * TESTCARD GORETN SETSLOTCORD	:=>NOT JIVE OR DIAGS :LET IT PRECESS DOWN :START FROM BEXX DOWN :FOR SUBTRACT :BACK DOWN TO NEXT PAGE :STAY AWAY FROM STACK: :(ALWAYS) :RUN DIAGS :CARD PLUGGED IN? :=>YES :NO. DISABLE ROM
C247:10 18 C249: C249: C249: C249: C249: C249: A B C C249: C249: A B C C249: C249: C249: C252: C252: C252: C252: C252: C253: C254: C253: C254: C	C264 C252 C252 C264 C264 C2 C264 C27D C27D C27D C27D C264	2263 2264 2265 2267 2268 2267 2272 2272 2272 2273 2274 2275 2276 2277 2278 2278 2278 2278 2278 2278	* BLAST 2 B' INCLUDING * BLAST DIAOS *	BPL (TES OF THE F LDY STA LDA STA LDA STA EQU JSR BRE EQU JSR BNE EQU EQU COMP BNE EQU COMP BNE EQU COMP COM	RESETRET F EACH PAGE, RESET VECTOR: #\$D0 All #\$DF (AlL),Y (AlL),Y (AlL),Y (AlL),Y (AlL),Y #1 BLAST RESETRET # \$C401 # TESTCARD GORETN SETSLOTC3ROM GORETN	<pre>;=>NOT JIVE OR DIAGS ;LET IT PRECESS DOWN ;START FROM BFXX DOWN ;FOR SUBTRACT ;BACK DOWN TO NEXT PAGE ;STAY AWAY FROM STACK! ;(ALWAYS) ;RUN DIAGS ;CARD PLUGGED IN? ;=>YES ; (ALWAYS)</pre>
C247:10 18 C249: C249: C249: C249: C249: C249: C249: C249: C249: S 52 C249: S 52 C249: S 52 C249: S 53 C254: S 53 C252: S 53 C254: S 53 C254: S 53 C254: S 79 C254: S 79 C254: S 79 C254:	C264 C252 C252 C264 C264 C264 C264 C264 C270 C270 C270 C270 C265	2233 - 2254 - 2254 - 2255 - 2267 - 2267 - 2267 - 2270 - 2271 - 2273 - 2275 - 2277 - 2275 - 2277 - 2275 - 22	* BLAST 2 BY INCLUDING BLAST DIAGS * RESETRET	BPL THES OF THE F LDY LDA STA LDA SEC EQU STA BEQU JSR EQU JSR BEQU STA BREQ STA BREQ STA BREQ STA BREQ STA BREQ STA	RESETRET F EACH PAGE, RESET VECTOR: #0 #0 A1L #8BF * A1H (A1L),Y #1 A1H, (A1L),Y #1 BLAST RESETRET * \$CA01 * \$CA01 * \$CA01 * * \$CA01 * * * * * * * * * * * * *	<pre>;=>NOT JIVE OR DIAGS ;LET IT PRECESS DOWN ;START FROM BFXX DOWN ;FOR SUBTRACT ;BACK DOWN TO NEXT PAGE ;STAY AWAY FROM STACK: ;(ALWAYS) ;RUN DIAGS ;CARD PLUGGED IN? ;=>YES ;(ALWAYS TAKEN) ;(ALWAYS TAKEN) ;FORCE TO UPPERCASE</pre>
C247:10 18 C249: C249: C249: C249: C249: C249: A B C C249: C249: A B C C249: C251:38 C252:85 C252:85 C252:85 C253:89 C253:19 C253:19 C253:19 C253:10 C253:10 C253:10 C253:10 C253:10 C253:10 C253:10 C253:10 C254:10 C255:10 C	C264 C252 C252 C264 C264 C27D C27D C27D C27D C27D	2234 2264 2266 2267 2268 2273 2273 2273 2274 2275 2274 2275 2275 2275 2275 2275	* BLAST 2 BY * INCLUDING BLAST BLAST * RESETRET B. ESCFIX	BPL (TES OF THE F LDY LDA STA LDA STA STA STA STA STA STA BLOY STA STA BEQ BEQ BEQ SRC EQU JSR BEQ STA BEQ LDY LDA BEQ LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA STA STA STA STA STA STA ST	RESETRET F EACH PAGE, RESET VECTOR: #0 #0 A1L #8BF * A1H (A1L),Y #1 A1H, (A1L),Y #1 BLAST RESETRET * \$CA01 * \$CA01 * \$CA01 * * \$CA01 * * * * * * * * * * * * *	<pre>;=>NOT JIVE OR DIAGS ;LET IT PRECESS DOWN ;START FROM BFXX DOWN ;FOR SUBTRACT ;BACK DOWN TO NEXT PAGE ;STAY AWAY FROM STACK! ;(ALWAYS) ;RUN DIAGS ;CARD PLUGGED IN? ;=>YES ; (ALWAYS)</pre>
C247:10 18 C249: C249: C249: C249: C249: C249:A0 80 C248:A7 00 C248:A9 00 C248:B5 3C C252:B5 3D C252:B5 3D C252:B5 3D C254:F1 3C C255:F7 01 C258:C9 01 C258:C9 01 C258:F0 03 C257:F0 13 C254:C264:C264: C264:C264:C264:C264:C264:C264:C264:C264:	C252 C252 C252 C264 C264 C264 C270 C270 C264 C270 C270 C265 C272	2234 2264 2266 2267 2268 2270 2271 2273 2274 2275 2274 2275 2275 2274 2275 2274 2275 2275	* BLAST 2 BY INCLUDING BLAST DIAGS * RESETRET	REPL STHE F LDY STATE F LDY STA LDA STA LDA STA LDA STA STA SBC CMP EQU JSR CCMP EQU JSR EQU JSR EQU LDY STA STA STA CMP EQU LDY STA STA SBC CMP EQU LDY STA STA SBC CMP EQU LDY STA STA SBC CMP EQU LDY STA STA SBC CMP EQU LDY STA STA SBC CMP EQU LDY STA STA SBC CMP EQU LDY STA STA SBC CMP EQU LDY STA STA SBC CMP EQU LDY STA STA SBC CMP EQU LDY STA STA SBC CMP EQU LDY STA STA SBC CMP EQU LDY STA STA SBC CMP EQU LDY STA STA SBC CMP EQU LDY STA STA SBC CMP EQU LDY STA STA SBC CMP EQU LDY STA SCA STA SBC CMP EQU LDY STA STA SBC CMP EQU LDY STA STA SBC CMP EQU LDY STA STA SBC CMP STA SBC SCA SCA STA SBC STA SDC STA SCA STA SCA STA SCA STA SCA STA SCA STA SCA STA STA STA STA STA STA STA STA STA ST	RESETRET F ACH PAGE, RESET VECTOR: #580 #0 A1L #585 * A1H (A1L),Y #1 #1 #1 #1 #1 #1 #1 #1 #1 #1	:=>NOT JIVE OR DIAGS :LET IT PRECESS DOWN :START FROM BEXX DOWN :FOR SUBTRACT :BACK DOWN TO NEXT PAGE :STAY AWAY FROM STACK: :(ALWAYS) :RUN DIAGS :CARD PLUGGED IN? :=>YES :(ALWAYS TAKEN) :(ALWAYS TAKEN) :FORCE TO UPPERCASE :SCAN FOR A MATCH
C247:10 18 C249: C249: C249: C249: C249: C249: A B C C249: C249: A B C C C C C C C C C C C C C C C C C C	C252 C252 C252 C254 C264 C264 C270 C270 C270 C265 C272 C272 C2	2434 2244 2244 2244 2244 2245 2243 2243	* BLAST 2 BY * INCLUDING BLAST BLAST * RESETRET B. ESCFIX	ATES OF THE F LDY LDA STA STA STA STA STA STA STA STA STA	RESETRET F EACH PAGE, RESET VECTOR: #550 #0 All #55F * AlH (AlL),Y (AlL),Y (AlL),Y * (AlL),Y * (AlL),Y * * (AlL),Y * * * * * * * * * * * * *	<pre>;=>NOT JIVE OR DIAGS ;LET IT PRECESS DOWN ;START FROM BFXX DOWN ;FOR SUBTRACT ;BACK DOWN TO NEXT PAGE ;STAY AWAY FROM STACK! ;(ALWAYS) ;RUN DIAGS ;CARD PLUGGED IN? ;>>YES ;NO. DISABLE ROM ;(ALWAYS TAKEN) </pre>
C247:10 18 C249: C249: C249: C249: C249: C249: C249: C249: C249: C249: C249: C251: C251: C252: C252: C252: C252: C254: C	C264 C252 C252 C264 C264 C264 C264 C264 C27D C27D C27D C272 C272 C272 C277	2434 2244 2264 2267 2268 2267 2272 2273 2274 2275 2274 2275 2275 2275 2275 2275	* BLAST 2 BY * INCLUDING BLAST BLAST * RESETRET B. ESCFIX	TTES OF 6 THE S OF 6 LDY LDY STAC STAC STAC STAC STA STA SEC STA STA STA STA STA STA STA STA STA STA	RESETRET F ACH PAGE, RESET VECTOR: #\$B0 #0 A1L #\$BF * A1H (A1L),Y (A1L),Y (A1L),Y (A1L),Y (A1L),Y * * CAC1 * * * * * * * * * * * * *	<pre>;=>NOT JIVE OR DIAGS ;LET IT PRECESS DOWN ;START FROM BEXX DOWN ;FOR SUBTRACT ;BACK DOWN TO NEXT PAGE ;STAY AWAY FROM STACK! ;(ALWAYS) ;RUN DIAGS ;CARD PLUGGED IN? ;=>YES ;IND. DISABLE ROM ;(ALWAYS TAKEN) ;FORCE TO UPPERCASE ;SCAN FOR A MATCH ;IS IT? ;=>NAM</pre>
C247:10 18 C249: C249: C249: C249: C249: C249: C249: C249: C251:38 C252:85 3D C254:91 3C C254:91 3C C254:19 C254:19 C254:10 73 C254:10 73 C257:10 73 C257:10 73 C257:10 14 C254:10 74 C254:10 74 C254:10 74 C254:10 75 C254:10 75 C255:10 75 C257:10 75 C277:10 75 C275 C275 C275 C275 C275 C275 C275 C2	C252 C252 C252 C254 C264 C3 C264 C3 C270 C270 C270 C270 C272 C2 C2 C2 C2 C2 C2 C2 C2 C2 C2 C2 C2 C2	2233 2244 2244 2244 2244 2244 2244 2244	* BLAST 2 BY * INCLUDING BLAST BLAST * RESETRET B. ESCFIX B. ESCFIX2	BPL OF THE FF CTTES OF THE FF LDA STA STA STA STA STA STA STA STA STA STA STA	RESETRET F ACH PAGE, RESET VECTOR: #\$B0 #0 A1L #\$BF * A1H (A1L),Y (A1L),Y (A1L),Y (A1L),Y (A1L),Y * * CAC1 * * * * * * * * * * * * *	<pre>;=>NOT JIVE OR DIAGS ;LET IT PRECESS DOWN ;START FROM BFXX DOWN ;FOR SUBTRACT ;BACK DOWN TO NEXT PAGE ;STAY AWAY FROM STACK! ;(ALWAYS) ;RUN DIAGS ;CARD PLUGGED IN? ;>>YES ;NO. DISABLE ROM ;(ALWAYS TAKEN) </pre>
C247:10 18 C249: C249: C249: C249: C249: C249: C249: C249: C249: C249: C249: C251: C251: C252: C252: C252: C252: C254: C	C264 C252 C252 C264 C264 C264 C264 C270 C270 C270 C270 C272 C272 C272 C272	2434 2244 2246 2246 2246 2247 2248 2249 2277 2273 2274 2275 2274 2275 2274 2275 2274 2275 2277 2280 2277 2280 2281 2283 2279 2283 2284 2285 2285 2285 2285 2285 2285 2285	* BLAST 2 BY * INCLUDING BLAST BLAST * RESETRET B. ESCFIX	TTES OF 6 THE S OF 6 LDY LDY STALE STA SEC SEC STA SEC SEC STA SEC STA SEC SEC STA SEC SEC STA SEC SEC STA SEC SEC STA SEC SEC STA SEC SEC STA SEC STA SEC SEC STA SEC SEC STA SEC SEC STA SEC SEC STA SEC SEC STA SEC SEC STA SEC SEC STA SEC SEC STA SEC SEC STA SEC SEC STA SEC SEC STA SEC STA SEC STA SEC STA SEC STA SEC STA SEC STA SEC STA SEC STA SEC STA SEC STA SEC STA SEC STA SEC STA STA STA STA STA STA STA STA STA STA	RESETRET F ACH PAGE, RESET VECTOR: #\$B0 #0 A1L #\$BF * A1H (A1L),Y (A1L),Y (A1L),Y (A1L),Y (A1L),Y * * CAC1 * * * * * * * * * * * * *	<pre>;=>NOT JIVE OR DIAGS ;LET IT PRECESS DOWN ;START FROM BEXX DOWN ;FOR SUBTRACT ;BACK DOWN TO NEXT PAGE ;STAY AWAY FROM STACK! ;(ALWAYS) ;RUN DIAGS ;CARD PLUGGED IN? ;=>YES ;IND. DISABLE ROM ;(ALWAYS TAKEN) ;FORCE TO UPPERCASE ;SCAN FOR A MATCH ;IS IT? ;=>NAM</pre>
C247:10 18 C249: C249: C249: C249: C249: C249: C249:A0 80 C240:B5 3C C251:38 C252:85 3D C254:71 3C C254:11 3C C254:11 3C C257:11 3C C254:12 3C C251:12 3C C270:12 3C	C252 C252 C252 C254 C254 C264 C3 C270 C270 C270 C270 C272 C277 C2 C277 C2 C277 C277	2434 - 2264 - 2264 - 2266 - 2267 - 2268 - 2267 - 2273 - 2273 - 2274 - 2275 - 2277 - 2278 - 2277 - 22	* BLAST 2 BY * INCLUDING BLAST BLAST * RESETRET B. ESCFIX B. ESCFIX2	BPL OF THE OF TH	RESETRET F EACH PAGE, RESET VECTOR: #550 #0 A1L #557 * A1H (A1L),Y (A1L),Y (A1L),Y (A1L),Y (A1L),Y (A1L),Y * (A1L),Y * * * * * * * * * * * * *	<pre>;=>NOT JIVE OR DIAGS ;LET IT PRECESS DOWN ;START FROM BEXX DOWN ;FOR SUBTRACT ;BACK DOWN TO NEXT PAGE ;STAY AWAY FROM STACK! ;(ALWAYS) ;RUN DIAGS ;CARD PLUGGED IN? ;=>YES ;IND. DISABLE ROM ;(ALWAYS TAKEN) ;FORCE TO UPPERCASE ;SCAN FOR A MATCH ;IS IT? ;=>NAM</pre>
C247:10 18 C249: C249: C249: C249: C249: C249: C249: C249: C249: C249: C249: C249: C251: C252: C252: C252: C252: C252: C254: C272: C	C264 C252 C252 C264 C264 C264 C264 C264 C270 C270 C270 C270 C272 C272 C272 C274 C272 C274 C272 C274 C272	2634 2264 2265 2266 2267 2272 2273 2274 2275 2277 2278 2275 2277 2278 2275 2277 2278 2283 2284 2283 2284 2285 2285 2285 2285 2285 2297 2298 2299 2299 2299 2299 2299 2299	* BLAST 2 BY * INCLUDING BLAST BLAST BLASS * BLASS * BLESCFIX BLESCFIX2 BLESCFIX3	BPL TTES OF THE T LDY LDY LDA STAL DA STAL	RESETRET F ACH PAGE, RESET VECTOR: #\$B0 #0 A1L #\$BF * A1H (A1L),Y #1 BLAST RESETRET * \$C401 * TESTCARD GORETN * # BLSCFIX3 ESCUN,Y B.ESCFIX3 ESCUT,Y * B.ESCFIX3	<pre>;=>NOT JIVE OR DIAGS ;LET IT PRECESS DOWN ;START FROM BFXX DOWN ;FOR SUBTRACT ;BACK DOWN TO NEXT PADE ;STAY AWAY FROM STACK! ;(ALWAYS) ;RUN DIAGS ;CARD PLUGGED IN? ;=>YES ;NOL DISABLE ROM ;(ALWAYS TAKEN) ;FORCE TO UPPERCASE ;SCAN FOR A MATCH ;IS IT? ;=>NAK ;STANSLATE IT</pre>
C247:10 18 C249: C249: C249: C249: C249: C249:A0 80 C2418:A9 00 C2418:A9 00 C2418:A9 00 C251:38 C252:85 C252:85 C252:85 C252:87 C253:47 13 C255:47 13 C255:47 13 C255:47 13 C255:47 13 C255:47 13 C255:47 13 C257:47 13 C258:47 13 C258:47 13 C258:47 13 C258:47 13 C254:47 13 C275:47 13 C275	C252 C252 C252 C254 C254 C264 C270 C270 C270 C270 C2772 C2772 C2772 C2772 C2772 C2772 C2772 C2772	2634 2264 2265 2266 2267 2272 2273 2274 2275 2277 2278 2275 2277 2278 2275 2277 2278 2283 2284 2283 2284 2285 2285 2285 2285 2285 2297 2298 2299 2299 2299 2299 2299 2299	* BLAST 2 BY * INCLUDING BLAST BLAST * RESETRET B. ESCFIX B. ESCFIX2	BPL OF THE OF TH	RESETRET F ACH PAGE, RESET VECTOR: #\$B0 #0 A1L #\$BF * A1H (A1L),Y #1 BLAST RESETRET * \$C401 * TESTCARD GORETN * # BLSCFIX3 ESCUN,Y B.ESCFIX3 ESCUT,Y * B.ESCFIX3	<pre>;=>NOT JIVE OR DIAGS ;LET IT PRECESS DOWN ;START FROM BEXX DOWN ;FOR SUBTRACT ;BACK DOWN TO NEXT PAGE ;STAY AWAY FROM STACK! ;(ALWAYS) ;RUN DIAGS ;CARD PLUGGED IN? ;=>YES ;IND. DISABLE ROM ;(ALWAYS TAKEN) ;FORCE TO UPPERCASE ;SCAN FOR A MATCH ;IS IT? ;=>NAM</pre>
C247:10 18 C249: C249: C249: C249: C249: C249: C249: C249: C249: C249: C249: C249: C251: C252: C252: C252: C252: C252: C254: C272: C	C252 C252 C252 C254 C254 C264 C270 C270 C270 C270 C2772 C2772 C2772 C2772 C2772 C2772 C2772 C2772	2633 - 2264 - 2264 - 2265 - 2266 - 2267 - 2268 - 2272 - 2273 - 2273 - 2273 - 2273 - 2274 - 2275 - 2274 - 2275 - 2274 - 2280 2282 - 2274 - 2282 - 2274 - 2282 - 2274 - 2282 - 2274 - 2282 - 2274 - 2282 - 2274 - 2283 - 2284 2284	* BLAST 2 BY * INCLUDING BLAST BLAST BLASS * BLASS * BLESCFIX BLESCFIX2 BLESCFIX3	BPL TTES OF THE T LDY LDY LDA STAL DA STAL	RESETRET F ACH PAGE, RESET VECTOR: #\$B0 #0 A1L #\$BF * A1H (A1L),Y (A1L),Y (A1L),Y (A1L),Y (A1L),Y (A1L),Y (A1L),Y * SC401 * * SC401 * * SC401 * * SC401 * * SC501	<pre>;=>NOT JIVE OR DIAGS ;LET IT PRECESS DOWN ;START FROM BFXX DOWN ;FOR SUBTRACT ;BACK DOWN TO NEXT PADE ;STAY AWAY FROM STACK! ;(ALWAYS) ;RUN DIAGS ;CARD PLUGGED IN? ;=>YES ;NOL DISABLE ROM ;(ALWAYS TAKEN) ;FORCE TO UPPERCASE ;SCAN FOR A MATCH ;IS IT? ;=>NAK ;STANSLATE IT</pre>

80-Column Firmware Listing

C280:88 95 C284:CA CB						
	BA BB	303	ESCIN	DFB	\$88, \$75, \$8A,	•88
C288:	CD C9	304	ESCOUT	ASC	'JKMI'	THE ARROWS
C288:	C288		B. KEYIN	EQU	*	
C289: 8D 78		307	2. M2. M	STA	TEMP1	SAVE ORIGINAL CHAR
C28B: 68		308		PLA		HOLD ONTO
C28C: A8		309		TAY		CXBANK STATUS
C28D: 68 C28E: 48		310		PLA		OET USER'S
C28F: 6A		311		PHA ROR	A	; IRQ STATE ; MOVE IRQ BIT TO
C290: 6A		313		ROR	A	THE
C291:6A		314		ROR	A	CARRY
C292: 98		315		TYA		PUT CXBANK STATUS
C293: 48 C294: 8A		316		PHA TXA		BACK ON STACK
C275:48		318		PHA		; SAVE ; XREQ
C296:		319	*	T DE		/ AREW
C296: 88		320		CLV		ASSUME NOT INTERRUPTIBLE
C297: BO 03	C54C	321		BCS	B. KEYIN2	: THE WERE RIGHT
C299:2C 00 C29C:	CF C290	322	-	BIT	SEV	SAY "INTERRUPTIBLE"
C29C: A9 FF	C54C	323	B. KEYIN2	EQU	* #\$FF	CURSOR=NORMAL DELETE
C29E: A4 24		325		LDY	CH	CORSUR-NURMAL DELETE
C2A0: 91 28		326		STA	(BASL), Y	
C2A2: 20 C6		327		JSR	KEYDLY	WAIT FOR A KEY
C2A5: BO OE	C2B5	328		BCS	COTKEY	=>GOT ONE
C2A7: AD 78	04	329		LDA		REPLACE ORIG CHAR
C2AA: A4 24 C2AC: 91 28		330 331		LDY	СН	
C2AE: 20 C6	62	332		STA	(BASL), Y	HATT FOR A KEY
C2B1: B0 02	C285	333		BCS		;WAIT FOR A KEY ;=>GOT ONE
C283: 90 E7	C29C	334		BCC		(ALWAYS TAKEN)
C285:		335	*			100 C
C285:	C285		GOTKEY	EQU	*	
C285: AD 78	04	337		LDA		RESTORE DRIGINAL
C288: A4 24 C28A: 91 28		338 337		LDY	(BASL), Y	ALIABACTER
C28C: 68		340		PLA		; CHARACTER ; RESTORE
C2BD: AA		341		TAX		i XREQ
C2BE: AD OO	CO	342		LDA		
C2C1:8D 10		343		STA	KBDSTRB	GET THE NEW KEYSTROKE
C2C4: 30 25	C2EB	344		BMI	F. RETURN	(ALWAYS TAKEN)
C2C6: C2C6:			*** *** INPUT: "			
C2C6:	C2C6		KEYDLY	EQU	SET IF INTERR	UPTIBLE
C2C6: A2 0C	6260	348	NETUL 1	LDX		SHORT DELAY FOR IRG
C2C8: 70 02	CSCC	349		BVS		->INTERRUPTIBLE
C2CA: A2 31		350		LDX	#\$31	LONG DELAY FOR NO IRG
C2CC:	C2CC		IK1	EQU	*	
C2CC: A0 00 C2CE:		352		LDY	#0	
		-				
	C2CE	353	IK2	EQU	*	
C2CE: 50 05	C2CE C2D5	353 354	IK2	EQU BVC	* IK2A	ANDT INTERRUPTIBLE
C2CE: 50 05 C2D0: 08	C2D5	353 354 355	IK2	EQU BVC PHP	* *	SAVE OFLOW
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D4: 28	C2D5 FC	353 354 355 356 357		EQU BVC PHP JSR PLP	* IK2A SNIFFIRQ	
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D4: 28 C2D5:	C2D5	353 354 355 356 357 358	IK2	EQU BVC PHP JSR PLP EQU	* IK2A SNIFFIRG *	;SAVE OFLOW ;ALLOW IRG
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D4: 28 C2D5: C2D5: E6 4E	C2D5 FC C2D5	353 354 355 356 357 358 358		EQU BVC PHP JSR PLP EQU INC	* IK2A SNIFFIRG * RNDL	;SAVE OFLOW ;ALLOW IRG
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D4: 28 C2D5: C2D5: E6 4E C2D7: D0 02	C2D5 FC	353 354 355 356 357 358 359 359		EQU BVC PHP JSR PLP EQU INC BNE	* IK2A SNIFFIRO * RNDL IK3	;SAVE OFLOW ;ALLOW IRG
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D4: 28 C2D5: C2D5: E6 4E C2D7: D0 02 C2D7: E6 4F	C2D5 FC C2D5 C2D8	353 354 355 356 357 358 359 360 361	IK2A	EQU BVC PHP JSR PLP EQU INC BNE INC	* IK2A SNIFFIRG * RNDL	;SAVE OFLOW ;ALLOW IRG
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D4: 28 C2D5: E6 4E C2D7: D0 02 C2D9: E6 4F C2D8: C2D8: AD 00	C2D5 FC C2D5 C2D8 C2D8	353 354 355 356 357 358 359 360 361 362	IK2A	EQU BVC PHP JSR PLP EQU INC BNE	* IK2A SNIFFIRG * RNDL IK3 RNDH *	;SAVE OFLOW ;ALLOW IRG
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D4: 28 C2D5: C2D5: E6 4E C2D7: D0 02 C2D7: E6 4F	C2D5 FC C2D5 C2D8 C2D8	353 354 355 356 357 358 359 360 361	IK2A	EQU BVC PHP JSR PLP EQU INC BNE INC EQU	* IK2A SNIFFIRG * RNDL IK3 RNDH * KBD	;SAVE OFLOW ;ALLOW IRQ ;RESTORE OFLOW
C2CE: 50 05 C2D1:08 C2D1:20 75 C2D4:28 C2D5: C2D5: C2D5: C2D5: C2D7: D0 02 C2D7: D0 02 C2D7: C2D8: A4F C2D8: AD 00 C2DE: 30 09 C2E0: 88	C2D5 FC C2D5 C2D8 C2D8 C2D8 C2D8 C2C8	353 354 355 356 357 358 357 360 361 362 363 364 365	IK2A	EQU BVC PHP JSR PLP EQU INC BNC EQU LDA BMI DEY	* IK2A SNIFFIRG * RNDL IK3 RNDH * KBD KDRETY	; SAVE OFLOW ; ALLOW IRG ; RESTORE OFLOW ; KEYPRESS?
C2CE: 50 05 C2D0:08 C2D1:20 75 C2D4:28 C2D5: C2D5:E6 4E C2D7:D0 02 C2D9:E6 4F C2D8: C2D8:AD 00 C2DE:30 09 C2DE:30 09 C2E0:88 C2E1:D0 E8	C2D5 FC C2D5 C2D8 C2D8 C0	353 354 355 356 357 358 357 360 361 362 363 364 365 365 366	IK2A	EQU BVC PHP JSR PLP EQU INC EQU LDA BMI DEY BNE	* IK2A SNIFFIRG * RNDL IK3 RNDH * KBD	; SAVE OFLOW ; ALLOW IRG ; RESTORE OFLOW ; KEYPRESS?
C2CE: 50 05 C2D0:08 C2D1:20 75 C2D4:28 C2D5: C2D5: C2D5: C2D7: D0 02 C2D7: D0 02 C2D7: C2D7: C2D	C2D5 FC C2D5 C2D8 C2D8 C2C2E9 C2CE	353 354 355 356 357 358 357 358 357 360 361 362 363 364 365 365 366 367	IK2A	EQU BVC PHP JSR, PLP EQU INC BNE LDA BMI DEY BNE DEX	* IK2A SNIFFIRG * NDDL IK3 RNDH * KBD KDRETY IK2	; SAVE OFLOW ; ALLOW IRG ; RESTORE OFLOW ; KEYPRESS?
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D4: 28 C2D5: E6 4E C2D7: D0 02 C2D9: E6 4F C2D8: AD 00 C2D8: AD 00 C2DE: 30 09 C2E0: 88 C2E1: D0 E8 C2E3: CA C2E4: D0 E6	C2D5 FC C2D5 C2D8 C2D8 C2D8 C2D8 C2C8	353 354 355 356 357 358 357 360 361 362 363 364 365 364 365 365 366 367 368	IK3	EQU BVC PHP JSR PLP EQU INC EQU LDA BMI DEY BNE	* IK2A SNIFFIRG * RNDL IK3 RNDH * KBD KDRETY	; SAVE OFLOW ; ALLOW IRG ; RESTORE OFLOW ; KEYPRESS?
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D4: 28 C2D5: E6 42 C2D5: E6 47 C2D8: 30 00 C2D8: 30 00 C2D8: 30 07 C2E0: 88 C2E1: D0 E8 C2E4: D0 E6 C2E4: 10 E6 C2E6: 18	C2D5 FC C2D5 C2D8 C2D8 C0 C2E7 C2CE C2CC	353 354 355 356 357 358 357 360 361 362 363 364 365 364 365 365 366 367 368	IK2A	EQU BVC PHP JSR EQU INC EQU LDA BMI DEY BNE DEX BNE	* IK2A SNIFFIRG * RNDL IK3 RNDH * KDD KDRETY IK2 IK1	; SAVE OFLOW ; ALLOW IRG ; RESTORE OFLOW ; KEYPRESS?
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D4: 28 C2D5: E6 4E C2D5: E0 02 C2D7: E0 47 C2D8: AD 00 C2D8: AD 00	C2D5 FC C2D5 C2D8 C2D8 C2E9 C2C5 C2C6 C2C6 C2E4	353 354 355 355 357 358 357 358 357 360 361 362 364 365 364 365 364 365 366 367 367 367 367 371	IK3 IK3 KDRETN	EQU BYC PHP JSR, PLP EQU INC EQU LDA BNE EQU LDA BMI DEY BNE EQU CLC BCC	* IK2A SNIFFIRG * RNDL IK3 RNDH * KDD KDRETY IK2 IK1	; SAVE OFLOW ; ALLOW IRG ; RESTORE OFLOW ; KEYPRESS?
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D4: 28 C2D5: E6 4E C2D5: E6 4E C2D5: E6 4E C2D5: AD 00 C2DE: 30 07 C2E0: 8B C2E1: D0 EB C2E4: D0 E6 C2E4: 10 E6 C2E6: 18 C2E6: 18 C2E7: 79 01	C2D5 FC C2D5 C2D8 C2D8 C2D8 C2E9 C2CE C2CC C2E5	353 354 355 355 357 357 357 357 357 357 360 361 362 362 364 365 366 366 366 366 366 367 366 367 371 372	IK3	EQU BVC JSR, PLP EQU INC BNC EQU LDA BNC EQU DEY BNE EQU CLC EQU	* IK2A SNIFFIRG * RNDL IK3 RNDH * KBD KDRETY IK2 IK1 *	; SAVE OFLOW ; ALLOW IRG ; RESTORE OFLOW ; KEYPRESS?
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D3: 28 C2D3: 26 4E C2D3: E6 4E C2D7: D0 02 C2D7: E6 4F C2D8: AD 00 C2D8: AD 00 C2D8: AD 00 C2D2: 30 07 C2E0: 88 C2E3: CA C C2E4: D0 E8 C2E4: D8 C2E4: D8 C2E5: D8 C2E5	C2D5 FC C2D5 C2D8 C2D8 C2D8 C2C8 C2C4 C2C4 C2C4 C2C4 C2C4 C2C4 C2C	353 354 355 355 357 359 360 361 362 364 365 366 365 366 366 367 366 367 371 372 373	IK3 IK3 KDRETN	EQU BVC PHP JSR PLP EQU INC BNE INC EQU LDA BMI DEY BNE DEX BNE CLC BCC BCC BCC SEC	* IK2A SNIFFIRG * RNDL IK3 RNDH * KDD KDRETY IK2 IK1 * KDRET *	; SAVE OFLOW ; ALLOW IRG ; RESTORE OFLOW ; KEYPRESS?
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D4: 28 C2D5: E6 4E C2D7: D0 02 C2D8: E6 4E C2D7: D0 02 C2D8: AD 00 C2DE: AD 00 C2DE: AD 00 C2DE: CA C2D8: CA CA C2D8: CA CA CA CA CA CA CA CA CA CA CA CA CA C	C2D5 FC C2D5 C2D8 C2D8 C2E9 C2C5 C2C6 C2C6 C2E4	353 354 355 355 357 359 360 361 365 365 366 365 366 365 366 366 366 366	IK3 IK3 KDRETN	EQU BVC PHP JSR FLP EQU INC EQU ABMI BNE EQU BNE EQU CLCC EQU CLCC EQU SEC EQU	* IK2A SNIFFIRG * RNDL IK3 RNDH * KBD KDRETY IK2 IK1 *	; SAVE OFLOW ; ALLOW IRG ; RESTORE OFLOW ; KEYPRESS?
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D3: 28 C2D3: 26 4E C2D3: E6 4E C2D7: D0 02 C2D7: E6 4F C2D8: AD 00 C2D8: AD 00 C2D8: AD 00 C2D2: 30 07 C2E0: 88 C2E3: CA C C2E4: D0 E8 C2E4: D8 C2E4: D8 C2E5: D8 C2E5	C2D5 FC C2D5 C2D8 C2D8 C2D8 C2C8 C2C4 C2C4 C2C4 C2C4 C2C4 C2C4 C2C	353 354 355 356 357 358 357 360 361 362 363 364 363 364 365 366 364 368 367 370 371 372 373 374 375	IK3 IK3 KDRETN	EQU BVC PHP JSR PLP EQU INC BNE INC EQU LDA BMI DEY BNE DEX BNE CLC BCC BCC BCC SEC	* IK2A SNIFFIRG * RNDL IK3 RNDH * KDD KDRETY IK2 IK1 * KDRET *	; SAVE OFLOW ; ALLOW IRG ; RESTORE OFLOW ; KEYPRESS?
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D3: 28 C2D3: E6 4E C2D7: D0 02 C2D9: E6 4F C2D8: AD 00 C2D9: AD 00 C2D8: AD 00 C2D8: AD 00 C2D8: AD 00 C2D2: 30 07 C2E0: 88 C2E1: D0 E8 C2E4: D0 E6 C2E6: 18 C2E7: 90 01 C2E7: 38 C2E7: 38 C2E7: 38 C2E7: 40	C2D5 FC C2D5 C2D8 C2D8 C2D8 C2C8 C2C4 C2C4 C2C4 C2C4 C2C4 C2C4 C2C	353 354 355 356 357 358 359 360 361 362 364 365 364 365 364 365 364 365 367 371 372 371 372 373 374	IK3 IK3 KDRETN KDRETY KDRET * * EXIT. EITH	EQU BVC PHP JSR, PLP EQU INC BNC EQU BNC EQU BNE BNE BNE EQU CLCC EQU SEC EQU RTS	* IK2A SNIFFIRG * RNDL IK3 RNDH * KDRETY IK2 IK4 * * KDRET * * X T WITH OR WI	; SAVE OFLOW ; ALLOW IRG ; RESTORE OFLOW ; KEYPRESS? ; =>YES
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D4: 28 C2D5: E6 4E C2D7: D0 02 C2D8: E6 4E C2D7: D0 02 C2D8: AD 00 C2DE: 30 09 C2E0: 8B C2E1: D0 EB C2E4: D0 EB C2E4: 10 EB C2E4: 10 E6 C2E6: 18 C2E6: 18 C2E7: 90 01 C2E7: 90 C2E7: 90 C2E7: 90 C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8:	C2D5 FC C2D5 C2D8 C2D8 C2D8 C2C8 C2C4 C2C4 C2C4 C2C4 C2C4 C2C4 C2C	353 354 355 356 357 358 359 360 361 362 363 364 365 364 365 364 367 370 371 372 373 374 375 374 377 378	IK2A IK3 KDRETN KDRETY KDRET * EXIT. EITH * ENABLING	EQU BVC PHP JSR, PLP EQU INC BNC EQU BNC EQU BNE BNE BNE EQU CLCC EQU SEC EQU RTS	* IK2A SNIFFIRG * RNDL IK3 RNDH * KDRETY IK2 IK4 * * KDRET * * X T WITH OR WI	; SAVE OFLOW ; ALLOW IRG ; RESTORE OFLOW ; KEYPRESS? ; =>YES
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D3: 28 C2D3: 28 C2D3: 26 C2D3: 26 C2D3: 26 C2D3: 26 C2D3: 20 C2D9: 26 4F C2D8: AD 00 C2DE: 30 07 C2E0: 88 C2E1: 10 E8 C2E4: 10 E8 C2E4: 10 E8 C2E4: 10 E8 C2E4: 10 E8 C2E4: 10 E8 C2E4: 10 C8 C2E4: 10 C8 C2E8: 10 C8 C8 C8 C8 C8 C8 C8 C8 C8 C8 C8 C8 C8 C	C2D5 FC C2D5 C2D8 C2D8 C2C7 C2C4 C2C4 C2C4 C2C4 C2C4 C2C4 C2C4	353 354 355 356 357 358 357 359 360 361 362 363 364 365 364 365 364 365 364 367 377 373 374 377 377 378 377	IK2A IK3 KDRETN KDRETY KDRET * EXIT. EITH * ENABLING *	EQU BVC PHP JSR PLP EQU INC EQU INC EQU LDA BMI EQU CLCA BME EQU CLC EQU SEC EQU SEC EQU SEC EQU SEC EQU SEC EQU SEC SEC EQU SEC SEC SEC SEC SEC SEC SEC SEC SEC SEC	* IK2A SNIFFIRG * RNDL IK3 RNDH * KDRET IK2 IK1 * * KDRET * * Y WITH OR WI ACE.	; SAVE OFLOW ; ALLOW IRG ; RESTORE OFLOW ; KEYPRESS? ; =>YES
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D4: 28 C2D5: E6 4E C2D7: D0 02 C2D9: E6 4E C2D9: E6 4F C2D8: 30 09 C2E0: 88 C2E1: 10 EB C2E2: CA C2E4: 10 EB C2E4: 10 EC C2E4: 10 C2 C2E4: 10 C2 C2E4: 10 C1 C2E7: 38 C2E7: 38 C2E8:	C2D5 FC C2D5 C2D8 C2D8 C2D8 C2C8 C2C4 C2C4 C2C4 C2C4 C2C4 C2C4 C2C	353 354 355 357 358 357 359 360 361 362 363 364 364 364 364 364 364 364 364 364	IK2A IK3 KDRETN KDRETY KDRET * EXIT. EITH * ENABLING	EQU BYC PHP JSR, PLP EQU INC INC EQU LDA BNE EQU LDA BNE EQU DEY BNE EQU SEC EQU SEC EQU SEC EQU SEC EQU	* IK2A SNIFFIRG * RNDL IK3 RNDH * KDRETY IK2 IK1 KIR * * * * * * * * * * * * * * * * * * *	; SAVE OFLOW ; ALLOW TRG ; RESTORE OFLOW ; KEYPRESS? ; =>YES
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D3: 8 C2D5: 8 C2D5: 8 C2D5: 8 C2D5: 8 C2D5: 8 C2D8: AD 00 C2D8: AD 00 C2D8: 30 07 C2E0: 88 C2E1: 10 E8 C2E4: 10 E8 C2E4: 10 E8 C2E4: 10 E8 C2E4: 10 E1 C2E7: 70 01 C2E7: 70 01 C2E7: 70 01 C2E7: 8 C2E4: 40 C2E8: 4	C2D5 FC C2D5 C2D8 C2D8 C0 C2E7 C2CC C2E6 C2E6 C2E6 C2E7 C2E8	353 354 355 355 359 359 360 361 362 363 364 365 364 365 364 365 364 365 370 371 372 373 374 377 377 377 377 377 377 377 377	IK2A IK3 KDRETN KDRETY KDRET * EXIT. EITH * ENABLING *	EQU BYC BYC BYC BYC JSR FLP EQU INC EQU EQU EQU EQU EQU EQU SEC EQU SEC EQU SEC EQU SEC EQU SEC EQU SEC EQU FLP	* IK2A SNIFFIRG * RNDL IK3 RNDH * KDD KORETY IK2 IK1 * * KDRET * * Y WITH OR WI ACE. *	; SAVE OFLOW ; ALLOW IRG ; RESTORE OFLOW ; KEYPRESS? ; >>YES THOUT ; GET PRIOR I/O DISABLE
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D4: 28 C2D5: E6 4E C2D7: D0 02 C2D9: E6 4E C2D9: E6 4F C2D8: 30 09 C2E0: 88 C2E1: 10 EB C2E2: 10 EB C2E3: CA C2E4: 10 E6 C2E4: 10 E6 C2E4: 10 E6 C2E4: 10 C2 C2E4: 10 C1 C2E7: 90 01 C2E7: 90 01 C2E7: 90 01 C2E7: 38 C2E4: 40 C2E8: C2E8: C2E8: C2E8:	C2D5 FC C2D5 C2D8 C2D8 C2E7 C2C6 C2E7 C2C6 C2E4 C2E9 C2E4 C2E5 C2E4 C2E5 C2E5 C2E5	353 354 355 355 358 359 350 361 362 363 364 365 364 365 364 367 370 371 372 373 374 377 377 377 377 377 377 377 377	IK2A IK3 KDRETN KDRETY KDRET * EXIT. EITH * ENABLING *	EQU BYC BYC JSR JSR FLP EQU INC INC INC EQU LDA BNE EQU EQU BNE EQU BEC EQU SEC CLC EQU SEC EQU SEC EQU FLP FLP EQU BNE EQU SEC EQU FLP FLP EQU EQU SEC EQU SEC EQU FLP FLP EQU INC EQU SEC EQU SEC EQU SEC EQU EQU SEC E EQU SEC E EQU SEC E EQU SEC E EQU SEC E EQU SEC E EQU SEC E E EQU SEC E E E E E E E E E E E E E E E E E E	* IK2A SNIFFIRG * RNDL IK3 RNDH * KDRETY IK2 IK1 * KORET * * Y WITH OR WI ACE. * F. RET1	; SAVE OFLOW ; ALLOW IRG ; RESTORE OFLOW ; KEYPRESS? ; =>YES THOUT ; GET PRIOR I/O DISABLE ; =>LEAVE IT DISABLED
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D4: 28 C2D5: E6 4E C2D7: D0 02 C2D5: E6 4E C2D8: AD 00 C2D8: AD 00 C2D8: AD 00 C2D8: AD 00 C2E0: 88 C2E1: D0 EB C2E4: D0 EB C2E4: 10 EB C2E5: 28 C2E8: 28 C2E8: 28 C2E8: 28 C2E8: 28 C2E8: 28 C2E8: 28 C2E8: 28 C2E5:	C2D5 FC C2D5 C2D8 C2D8 C2C8 C2C4 C2C4 C2C4 C2C4 C2C4 C2C4 C2C	353 354 355 355 359 359 360 361 362 362 364 365 364 365 364 365 364 365 370 371 373 374 377 378 377 378 377 378 377 380 381 382 383	IK2A IK3 KDRETN KDRETY KDRET * EXIT. EITH * ENABLING *	EQU BYC BYC BYC BYC JSR FLP EQU INC EQU EQU EQU EQU EQU EQU SEC EQU SEC EQU SEC EQU SEC EQU SEC EQU SEC EQU FLP	* IK2A SNIFFIRG * RNDL IK3 RNDH * KDD KORETY IK2 IK1 * * KDRET * * T WITH OR WI ACE. * F.RET1 FUNCEXIT	; SAVE OFLOW ; ALLOW IRG ; RESTORE OFLOW ; KEYPRESS? ; >>YES THOUT ; OET PRIOR I/O DISABLE ; >>LEAVE IY DISABLED ; >>EAVE S ENABLE I/O
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D4: 28 C2D5: E6 4E C2D7: D0 02 C2D7: E6 4E C2D8: AD 00 C2D8: AD 00 C2D8: AD 00 C2D8: AD 00 C2E0: 08 C2E1: D0 E8 C2E4: D0 E6 C2E4: 10 E6 C2E4: 10 E6 C2E4: 10 E6 C2E4: 10 E6 C2E4: 18 C2E7: 90 01 C2E7: 90 01 C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8:	C2D5 FC C2D5 C2D8 C2D8 C0 C2E7 C2CC C2E6 C2E6 C2E6 C2E7 C2E8 C2E8 C2E1 FD	333 354 355 355 356 357 358 357 361 362 363 364 364 364 364 364 364 364 364 364	IK2A IK3 KDRETN KDRETY KDRET * EXIT. EIT * ENABLING * F. RETI	EQU PHP PLP JSR EQU EQU EQU EQU EQU EQU BNE EQU EQU EQU EQU EQU EQU EQU EQU EQU EQ	* IK2A SNIFFIRG * RNDL IK3 RNDH * KDD KORETY IK2 IK1 * * KDRET * * T WITH OR WI ACE. * F.RET1 FUNCEXIT	; SAVE OFLOW ; ALLOW IRG ; RESTORE OFLOW ; KEYPRESS? ; =>YES THOUT ; GET PRIOR I/O DISABLE ; =>LEAVE IT DISABLED
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D4: 28 C2D5: E6 4E C2D7: D0 02 C2D9: E6 4E C2D9: E6 4F C2D8: 30 09 C2E0: 88 C2E1: 10 EB C2E2: CA C2E4: 10 EB C2E4: 10 EB C2E4: 10 EB C2E4: 10 EB C2E4: 10 EB C2E5: 10 C2E C2E4: 10 C2 C2E4: 10 C2 C2E8: C2 C2E4: C2 C2E4: C2 C2E4: C2 C2E4: C2 C2E4: C2 C2E4: C2 C2E4: C2 C2E4: C2 C2E8: C2 C2E8: C2 C2E8: C2 C2E8: C2 C2E8: C2 C2E7: C2 C2E4: C2 C2E4: C2 C2E4: C2 C2E8: C2 C2E1: C2 C2E7: C2	C2D5 FC C2D5 C2D8 C2D8 C2C8 C2C4 C2C4 C2C4 C2C4 C2C4 C2C4 C2C	353 354 355 355 358 357 358 357 360 362 363 364 364 364 364 364 364 364 364 364	IK2A IK3 KDRETN KDRETY KDRET * * ENABLING F. RETURN	EQU PHP PLP EQU EQU EQU EQU EQU EQU EQU EQU EQU EQU	* IK2A SNIFFIRG * RNDL IK3 RNDH * KDRETY IK2 IK1 * * KDRET * * T WITH OR WI ACE. * F.RET1 FUNCEXIT+3 *	; SAVE OFLOW ; ALLOW IRG ; RESTORE OFLOW ; KEYPRESS? ; >>YES THOUT ; OET PRIOR I/O DISABLE ; >>LEAVE IY DISABLED ; >>EAVE S ENABLE I/O
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D4: 28 C2D5: E6 4E C2D5: E6 4E C2D5: E6 4E C2D5: E6 4E C2D5: E6 4E C2D8: AD 00 C2DE: 30 07 C2E0: 8E C2E1: D0 EB C2E4: D0 E6 C2E4: 10 E6 C2E4: 10 E6 C2E4: 10 E6 C2E4: 10 E6 C2E4: 18 C2E7: 90 01 C2E7: 90 01 C2E7: 90 01 C2E7: 90 01 C2E7: 90 01 C2E7: 8E C2E8: C2E8: C2E8: C2E8: C2E8: C2E	C2D5 FC C2D5 C2D8 C2D8 C2C2 C2E7 C2CC C2E5 C2E4 C2E5 C2E5 C2E5 C2E5 C2E5 C2E5 C2E5 C2E5	353 355 355 355 356 357 358 357 361 362 363 364 365 364 365 364 365 364 370 371 373 374 377 377 377 377 377 377 377 377	IK2A IK3 KDRETN KDRETY KDRET * ENABLING F. RETI X. CLEDLZ	EQU PHP JSP, JSP, EQU BNE EQU BNE EQU BNE EQU BNE EQU BNE EQU BNE EQU EQU FIS EQU FIS EQU FIS EQU EQU EQU EQU EQU EQU EQU EQU EQU EQU	* IK2A SNIFFIRG * RNDL IK3 RNDH * KDR KDRETY IK2 IK1 * * KDRET * * Y WITH OR WI ACE. * F.RET1 FUNCEXIT FUNCEXIT * * SAO	; SAVE OFLOW ; ALLOW IRG ; RESTORE OFLOW ; KEYPRESS? ; >>YES THOUT ; OET PRIOR I/O DISABLE ; >>LEAVE IY DISABLED ; >>EAVE S ENABLE I/O
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D4: 28 C2D5: E6 4E C2D7: D0 02 C2D9: E6 4E C2D9: E6 4F C2D8: AD 00 C2E0: 88 07 C2E0: 88 07 C2E0: 88 07 C2E0: 10 E8 C2E3: CA C2E4: 10 E6 C2E4: 10 E6 C2E4: 10 E6 C2E5: 10 C2 C2E4: 10 C1 C2E7: 38 C2E4: 40 C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: 28 C2E8: 28 C2E1: 4C 27 C2E74: 47 C2E74: 47 C2E76: 57 C2E76: 57 C2E766:	C2D5 FC C2D5 C2D8 C2D8 C0 C2E7 C2CC C2E6 C2E6 C2E6 C2E7 C2E8 C2E8 C2E1 FD	353 354 355 355 356 357 358 359 360 361 362 363 364 365 364 365 364 365 364 370 372 373 374 377 377 377 377 377 377 377 377	IK2A IK3 KDRETN KDRETY KDRET * EXIT. EIT * ENABLING * F. RETI	EQU PHP PLP EQU EQU EQU EQU EQU EQU EQU EQU EQU EQU	* IK2A SNIFFIRG * RNDL IK3 RNDH * KDRET * * KDRET * * KDRET * * T WITH OR WI ACE. * F.RET1 FUNCEXIT+3 * *	; SAVE OFLOW ; ALLOW IRG ; RESTORE OFLOW ; KEYPRESS? ; >>YES THOUT ; OET PRIOR I/O DISABLE ; >>LEAVE IY DISABLED ; >>EAVE S ENABLE I/O
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D4: 28 C2D5: E6 4E C2D5: E6 4E C2D5: E6 4E C2D5: E6 4E C2D5: E6 4E C2D8: AD 00 C2DE: 30 07 C2E0: 8E C2E1: 100 EB C2E4: 100 EB C2E5: 28 C2E8: 28 C2E4: 40 C2 C2E4: 40 C2 C2E4: 47 A0 C2F4: 47 A0 C2F4: 47 A0 C2F4: 47 A9	C2D5 FC C2D5 C2D8 C2D8 C2C2 C2E7 C2CC C2E5 C2E4 C2E5 C2E5 C2E5 C2E5 C2E5 C2E5 C2E5 C2E5	353 354 355 355 357 358 357 358 357 361 362 364 365 364 365 364 365 364 365 372 373 374 377 377 377 377 377 377 377 377	IK2A IK3 KDRETN KDRETY KDRET * ENABLING F. RETI X. CLEDLZ	EQU PHP SVC PHP JSR, JPLP ENC EQU ENC EQU EQU EQU EQU EQU EQU EQU EQU EQU EQU	* IK2A SNIFFIRG * RNDL IK3 RNDH * KDR KDRETY IK2 IK1 * * KDRET * * Y WITH OR WI ACE. * F.RET1 FUNCEXIT FUNCEXIT * * SAO	; SAVE OFLOW ; ALLOW IRG ; RESTORE OFLOW ; KEYPRESS? ; >>YES THOUT ; OET PRIOR I/O DISABLE ; >>LEAVE IY DISABLED ; >>EAVE S ENABLE I/O
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D4: 28 C2D5: E6 4E C2D7: D0 02 C2D7: E6 4E C2D7: D0 02 C2D7: E6 4E C2D8: AD 00 C2D8: 30 07 C2E0: 8E C2E1: D0 E8 C2E4: D0 E6 C2E4: 10 E8 C2E4: 10 E6 C2E4: 10 E6 C2E4: 10 E6 C2E4: 18 C2E7: 70 01 C2E7: 38 C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E4: C2F4: C2F4: C2 C2F4: C2 C2F4: C2 C2F4: C2 C2F4: C1 28 C2F5: C1 28 C2F	C2D5 FC C2D5 C2D8 C2D8 C2C2 C2E7 C2CC C2E5 C2E4 C2E5 C2E5 C2E5 C2E5 C2E5 C2E5 C2E5 C2E5	353 354 355 355 356 357 358 359 360 361 362 363 364 365 364 365 364 365 364 370 372 373 374 377 377 377 377 377 377 377 377	IK2A IK3 KDRETN KDRETY KDRET * ENABLING F. RETI X. CLEDLZ	EQU PHP PLP EQU EQU EQU EQU EQU EQU EQU EQU EQU EQU	* IK2A SNIFFIRG * RNDL IK3 RNDH * KDRET * * KDRET * * KDRET * * T WITH OR WI ACE. * F.RET1 FUNCEXIT+3 * *	; SAVE OFLOW ; ALLOW IRG ; RESTORE OFLOW ; KEYPRESS? ; >>YES THOUT ; OET PRIOR I/O DISABLE ; >>LEAVE IY DISABLED ; >>EAVE S ENABLE I/O
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D4: 28 C2D5: E6 4E C2D7: D0 02 C2D9: E6 4E C2D9: E6 4F C2D8: AD 00 C2E0: 88 07 C2E0: 88 07 C2E0: 88 07 C2E0: 88 07 C2E0: 88 07 C2E0: 10 E8 C2E3: CA C2E4: 10 E8 C2E4: 10 E6 C2E4: 10 E6 C2E4: 10 E6 C2E5: 10 E8 C2E7: 38 C2E4: 40 C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: 28 C2E8: 28 C2E8: 28 C2E8: 28 C2E8: 28 C2E8: 28 C2E8: 28 C2E8: 28 C2E8: 28 C2E6: 30 03 C2E1: 4C 27 C2E7: 47 A0 C2E76: 71 28 C2E76: 78 C2E76: 78	C2D5 FC C2D5 C2D8 C2D8 C2C2 C2E7 C2CC C2E5 C2E4 C2E5 C2E5 C2E5 C2E5 C2E5 C2E5 C2E5 C2E5	333 355 3356 3357 3358 3357 3359 3357 3359 3357 3359 340 340 340 340 340 340 340 340 340 340	IK2A IK3 KDRETN KDRETY KDRET * ENABLING F. RETI X. CLEDLZ	EQU PHP JSR, PLP LENC ENC ENC EQU ENC EQU EQU EQU EQU EQU EQU EQU EQU EQU EQU	* IK2A SNIFFIRG * RNDL IK3 RNDH * KDRET IK2 IK1 KDRET * * KDRET * * KDRET * * KDRET * * KDRET * * KDRET * * KDRET * * KDRET * * KDRET * * KDRET * * KDRET * * KDRET * * * KDRET * * * KDRET * * KDRET * * * KDRET * * * KDRET * * * KDRET * * * KDRET * * * KDRET * * * KDRET * * * KDRET * * * KDRET * * * KDRET * * * KDRET * * * * * * * * * * * * * * * * * * *	; SAVE OFLOW ; ALLOW IRG ; RESTORE OFLOW ; KEYPRESS? ; =>YES THOUT ; GET PRIOR I/O DISABLE ; =>LEAVE IY DISABLED ; =>LEAVE IY DISABLED ; =>EXIT & ENABLE I/O ; EXIT DISABLED
C2CE: 50 05 C2D0: 08 C2D1: 20 75 C2D4: 28 C2D5: E6 4E C2D7: D0 02 C2D7: E6 4E C2D7: D0 02 C2D7: E6 4E C2D8: AD 00 C2D8: 30 07 C2E0: 8E C2E1: D0 E8 C2E4: D0 E6 C2E4: 10 E8 C2E4: 10 E6 C2E4: 10 E6 C2E4: 10 E6 C2E4: 18 C2E7: 70 01 C2E7: 38 C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E8: C2E4: C2F4: C2F4: C2 C2F4: C2 C2F4: C2 C2F4: C2 C2F4: C1 28 C2F5: C1 28 C2F	C2D5 FC C2D5 C2D8 C2D8 C2E7 C2CC C2E6 C2E6 C2E7 C2E8 FD FD C2F1 FD C2F4 C2F4 C2F6	333 354 355 355 357 358 357 358 357 358 357 358 360 361 362 364 362 364 364 364 364 364 364 364 364 364 364	IK2A IK3 KDRETN KDRETY KDRET * ENABLING F. RETI X. CLEDLZ	EQU PHP JSR, PLP JSR, FLP JSR,	* IK2A SNIFFIRG * RNDL IK3 RNDH * KDRET IK2 IK1 KDRET * * KDRET * * KDRET * * KDRET * * KDRET * * KDRET * * KDRET * * KDRET * * KDRET * * KDRET * * KDRET * * KDRET * * * KDRET * * * KDRET * * KDRET * * * KDRET * * * KDRET * * * KDRET * * * KDRET * * * KDRET * * * KDRET * * * KDRET * * * KDRET * * * KDRET * * * KDRET * * * * * * * * * * * * * * * * * * *	; SAVE OFLOW ; ALLOW IRG ; RESTORE OFLOW ; KEYPRESS? ; >>YES THOUT ; OET PRIOR I/O DISABLE ; >>LEAVE IY DISABLED ; >>EAVE S ENABLE I/O

C2FE: 0002 C300:	375 5		DS INCLU	\$C300-*, 0 DE C3SPACE	
C300: C300:	2 3	*			
C300:	- 4		HE \$C3	XX ROM SPACE	
C300: C300:	5	*			
C300: C300	7	CN00	EQU	*	
C300: C300 C300: 2C 58 FF	8	BASICINT	EQU	* IORTS	SET VFLAG (INIT)
C303: 70 12 C317	10		BVS	BASICENT	ALWAYS TAKEN)
C305: C305	11	BASICIN	EQU	*	
C305: 38 C306: 90	12		SEC	\$70	BCC OPCODE (NEVER TAKEN)
C307: C307		BASICOUT	EQU	*	
C307: 18 C308: 88	15		CLC		CLEAR VELAG (NOT INIT)
C309: 50 0C C317	17		BVC	BASICENT	(ALWAYS TAKEN)
C30B: C30B:	18		1 FIRM	WARE PROTOCO	L TABLE:
C30B:	20	*			
C30B: 01 C30C: 88	21 22		DFB	\$01 \$88	GENERIC SIGNATURE BYTE
C30D:	23	*			
C30D: 48 C30E: 51	24 25		DFB	>JPINIT >JPREAD	PASCAL INIT
C30F: 57	26		DFB	>JPWRITE	PASCAL WRITE
C310: 5D C311:	27		DFB	>JPSTAT	PASCAL STATUS
C311:	29	*			
C311: C311:	30	* 128K SUPP	ORT RO	UTINE ENTRIE	B:
C311:4C 63 C3	32		JMP	MOVE	MEMORY MOVE ACROSS BANKS
C314:4C B0 C3 C317:	33		JMP	XFER	; TRANSFER ACROSS BANKS
C317:		* BASIC I/O	ENTRY	POINT:	
C317: C317: C317	36	BASICENT	EQU		
C317: 80 78 06	38	BHOICENT	STA	CHAR	SAVE CHARACTER
C31A: 48 C31B: 98	39 40		PHA		SAVE AC
C31C: 48	40		PHA		, and t
C31D: 8A	42 43		TXA		; AND X
C31E: 48 C31F: 08	43		PHP		SAVE CARRY & VELAG
C320:	45	*			
C320: C320:	46	* SET IRGMO	DE:		
C320: AD FB 04	48		LDA	MODE	ASSUME IRQ IS DISABLED
C323: 29 FE C325: 80 FB 04	49 50		AND	#255-M. IRG MODE	
C328: 68	51		PLA		GET PSTATUS
C329: 48 C32A: 29 04	52 53		PHA	#\$04	; AND LEAVE ON STACK ; IS 'I' BIT SET?
C32C: DO OB C336	54		BNE.	BASICENT2	=>YES, DISABLED
C32E: AD FB 04 C331: 09 01	55 56		LDA	MODE #M. IRG	
C333: 8D FB 04	57		STA	MODE	SET IT ENABLED
C336: C336 C336: AD FF CF	58 59	BASICENT2	EQU	* \$CFFF	KICK OUT ALL CB ROMS
C339: A5 25	60		LDA	cv	GET USER CV AND
C338: 8D F8 05 C33E: 20 E8 C3	61		JSR	SURCV SETC8	; STUFF IT FOR US ; SETUP CB INDICATOR
C341:28	63		PLP	02100	GET VELAG (INIT)
C342: 08 C343: 70 03 C348	64		PHP	JBASINIT	;=>DD THE INIT
C345: 4C 66 C8	66		JMP	CBBASIC	GET OUT OF CN SPACE
C348: 4C 03 CB C348:	67 68	JBASINIT	JMP	BASICINIT	;=>GOTO C8 SPACE
C34B: C34B	69	JPINIT	EQU	#	
C34B: 20 EB C3	70		JSR JMP	SETCO	SETUP CE INDICATOR
C34E:4C 4F CA C351: C351	71 72	JPREAD	EQU	PINIT *	XFER TO PASCAL INIT
C351:20 EB C3	73		JSR	SETCB	SETUP CB INDICATOR
C354:4C 74 CA C357: C357	74 75	JPWRITE	JMP	PREAD	XFER TO PASCAL READ
C357: 20 EB C3	76		JSR	SETCH	SETUP CE INDICATOR
C35A: 4C 8E CA C35D: C35D	77	JPSTAT	JMP	PWRITE *	XFER TO PASCAL WRITE
C35D: 20 EB C3	79		JSR	SETCB	SETUP CE INDICATOR
C360:4C 94 C9 C363:	80 82		JMP	PSTATUS	XFER TO PASCAL STATUS
C363:	83	* NAME :	MOVE		
C363: C363:	84		PERFO	IRM CROSSBANK	MEMORY MÖVE
C363:	86			URCE ADDRESS	
C363: C363:	87 88	* :	A4=DE CARRV	STINATION ST SET=MAIN>	ART CARD
C363	89	*		CLR=CARD>	MAIN
C363: C363:	90 91	* OUTPUT : * VOLATILE:	NONE	NG	
sama (3		STREET, STREET, STR	at section of	0050	

80-Column Firmware Listing

C363:	92 * CALLS :	NOTHI	NG	
C363:	93			
C363: C363: C363	94 * 95 MOVE	EQU		
C363: 48	96	PHA	#	SAVE AC
C364: 98	97	TYA		AND Y
C365: 48	98	PHA		
C366: AD 13 CO C369: 48	99 100	LDA	RORAMRD	SAVE STATE OF
C36A: AD 14 C0	101	LDA	RDRAMWRT	/ HEADAT FERGS
C36D: 48	102	PHA		
C36E: C36E:	103 * 104 * SET FLAGS			-
CBAE	105 # SET PLAGE	FURC	RUSSBANK HUV	E :
C36E: 90 08 C378	106	BCC	MOVEC2M	; =>CARD>MAIN
C370: 80 02 C0	107	STA	RDMAINRAM	SET FOR MAIN
C373:8D 05 C0 C376:80 06 C37E	109	STA	WRCARDRAM	; TO CARD ;=>(ALWAYS TAKEN)
C378:	110 *			
C378: C378 C378:80 04 C0	111 MOVEC2M 112	EQU	* WRMAINRAM	SET FOR CARD
C378:80 03 C0	113	STA	RDCARDRAM	TO MAIN
C37E:	114 *			
C37E: C37E	115 MOVESTRT	EQU	*	
C37E: A0 00 C380:	116 117 *	LDA	# 0	DUMMY INDEX
C380: C380	118 MOVELOOP	EQU	*	
C380: B1 3C	119	LDA	(A1L), Y	GET A BYTE
C382: 91 42 C384: E6 42	120	STA	(A4L), Y A4L	;MOVE IT
C386: D0 02 C38A	122	BNE	NXTA1	
C388; E6 43	123	INC	A4H	
C38A: A5 3C C38C: C5 3E	124 NXTA1 125	LDA	A1L A2L	
C38E: A5 3D	125	LDA	ALH	
C390: E5 3F	127	SBC	A2H	
C392:E6 3C C394:D0 02 C398	128	INC	A1L CO1	
C394:D0 02 C398 C396:E6 3D	127 130	BNE	AIH	
C398: 90 E6 C380	131 CO1	BCC	MOVELOOP	; =>MORE TO MOVE
C39A:	132 *			
C39A: C39A:	133 * RESTORE 0 134 *	RIGINA	L FLAGS:	
C39A: 8D 04 CO	135	STA	WRMAINRAM	CLEAR FLAG2
C39D: 68	136	PLA	APPERATION (GET ORIGINAL STATE
C39E: 10 03 C3A3 C3A0: BD 05 C0	137 138	BPL STA	CO3 WRCARDRAM	;=>IT WAS OFF
C3A3: C3A3	137 003	EQU	*	
C3A3: 80 02 CO	140	STA	RDMAINRAM	CLEAR FLAG1
C3A6:68 C3A7:10 03 C3AC	141 142	PLA BPL	MOVERET	;GET DRIGINAL STATE ;=>IT WAS OFF
C3A9: 80 03 CO	143	STA	RDCARDRAM	
C3AC: C3AC	144 MOVERET 145	EQU	*	RESTORE Y
C3AC: 68 C3AD: AB	145	PLA TAY		RESTORE Y
CJAE: 68	147	PLA		; AND AC
COAF: 60	148	RTS		
C3B0: C3B0:	149	XFER		
COBO	151 * FUNCTION:	TRANS	FER CONTROL	CROSSBANK
C3BO:	151 * FUNCTION: 152 * INPUT	\$03ED	TRANSFER AD	DR
C3BO: C3BO:	153 * : 154 *	CARRY	SET=XFER TO CLR=XFER TO	MATN -
C3BO:	155 * :	VFLAG	CLR=USE STD	ZP/STK
C3BO:	156 * :		SET-USE ALT	ZP/STK
C3B0: C3B0:		NONE	OJEE IN DES	T BANK
C3B0:		NOTHI	NG	/ DINT.
C3BO:	160 * NOTE :	ENTER	ED VIA JMP,	NOT JSR
C3BO: C3BO:	161			
C3BO: C3BO	162 * 163 XFER	EQU	*	
C3BO: 48	164	PHA		SAVE AC ON CURRENT STACK
C3B1:	165 *			7.15
C3B1: C3B1:	166 * COPY DEST 167 * OTHER BA	INALIO	N ADDRESS TO THAT WE HAVE	IT
C3B1:	168 * IN CASE	WE DO	A SWAP:	
C381:	167 *		40350	OFT VERADE 10
C3B1: AD ED 03 C3B4: 48	170 171	LDA PHA	\$03ED	; GET XFERADOR LO ; SAVE ON CURRENT STACK
C3B5: AD EE 03	172	LDA	\$03FE	GET XFERADOR HI
C3B8: 48	173	PHA		SAVE IT TOD
C3B9: C3B9:	174 * 175 * SWITCH TO		PRIATE BANK:	
C387:	176 *			
C387: 90 0A C3C5	177	BCC	XFERC2M	=>CARD>MAIN
C388:80 03 CO C38E:80 05 CO	178 179	STA	RDCARDRAM WRCARDRAM	SET FOR RUNNING
C3C1: 50 19 C3DC	180	BVC	XFERSZP	;=>USE STD ZP/STK
C3C3:70 08 C3CD C3C5: C3C5	181	BVS	XFERAZP	=>USE ALT ZP/STK
	182 XFERC2M	EQU	*	

3C5:8D 02 C0 3C8:8D 04 C0	183				
308.80 04 00			STA	RDMAINRAM	SET FOR RUNNING
	184		STA	WRMAINRAM	; IN MAIN RAM
3CB: 50 OF C31	C 185		BVC	XFERSZP	;=>USE STD ZP/STK
BCD:	186	*			
3CD: C30	D 187	XFERAZP	EQU	*	SWITCH TO ALT ZP/STK
3CD: 68	188		PLA		STUFF XFERADDR
3CE: 8D EE 03	189		STA	\$03EE	; HI AND
3D1:68 3D2:80 ED 03	190		PLA		
302:80 ED 03	191		STA	\$03ED	; LO
3D5:68	192		PLA		RESTORE AC
3D6:8D 09 CO	193		STA	SETALTZP	SWITCH TO ALT ZP/STK
3D9:6C ED 03	194		JMP	(\$03ED)	;=>OFF WE GO!
3DC:	195	*			
3DC: C31	C 196	XFERSZP	EQU	*	
3DC: 68	197		PLA		STUFF XFERADDR
300:8D EE 03	198		STA	\$03EE	; HI AND
3E0: 68	199		PLA		
3E1:8D ED 03	200		STA	\$03ED	; LO
3E4: 68	201		PLA		RESTORE AC
365: 80 08 CO	202		STA	SETSTDZP	;=>SWITCH TO STD ZP/STK
3E8:6C ED 03	203		JMP	(\$03ED)	OFF WE GO!
3EB:	204				And the operand the set of the Set
JEB:	205	* NAME :	SETCB		
BEB:	206	* FUNCTION:	SETUP	IRQ \$CBOO P	ROTOCOL
3EB:	207	* INPUT : * OUTPUT :	NONE		
3EB:	208	* OUTPUT :	NONE		
3EB:	209	* VOLATILE:	NOTHI	NG	
GEB:	210	* CALLS :	NOTHI	NG	
3EB:					
3EB:	212				
	EB 213		EQU	*	5445 40
3EB: 48	214		PHA		SAVE AC
BEC: A9 C3	215		LDA	#CCN00	SLUI NUMBER
BEE: 80 F8 07	216		STA	CBSLOT	STUFF IT
3F1:68	217		PLA		RESTORE AC
3F2: 60	218		RTS		
3F3:	6		INCLU	DE COSPACE	
3F3:	2				
3F3:		* THIS IS T			
3F3:					
3F3: 00			DO	TEST	
S	6		ORG	\$D800	
3F3:	7		ELSE		
NEXT DBJE				1	
800: C8			ORC	\$C800	
800:	9		FIN	NAMES OF TAXABLE POST	
800:4C 4A CA	10		JMP		PASCAL 1.0 INIT
803:	11	* BASIC INI	TIALIZ	ATION:	
800:4C 4A CA 803: 803:	11	* BASIC INI	TIALIZ	ATION:	;PASCAL 1.0 INIT
803: 803: 803: C8	11 12 03 13	* BASIC INI BASICINIT	EQU	ATION: *	
803: 803: 803: C8 803: A7 06	11 12 03 13 14	* BASIC INI BASICINIT	EQU EDA	ATION: * #G00DF8	CHECK F8 RDM
803: 803: 803: C8 803: A7 06 805: CD 83 FB	11 12 03 13 14 15	* BASIC INI BASICINIT	EQU LDA CMP	ATION: * #GOODF8 F8VERSION	;CHECK F8 ROM ; IS IT OK?
803: 803: 803: C8 803: A7 06 805: CD 83 FB	11 12 03 13 14 15 16 16	* BASIC INI BASICINIT	EQU LDA CMP BEQ	ATION: * #GDODF8 F8VERSION BINIT1	;CHECK F8 ROM ; IS IT OK? ;=>YES
803: 803: 803: CB 803: A7 06 805: CD B3 FB 808: F0 0C CB 808: 20 78 CF	11 12 03 13 14 15 16 16 17	* BASIC INI BASICINIT	EQU LDA CMP BEQ JSR	ATION: * #COODF8 F8VERSION BINIT1 COPYROM	;CHECK F8 ROM ; IS IT OK?
803: 803: CB: 803: A7 06 805:CD B3 FB 808:F0 0C CB 80A:20 78 CF 80D:CD B3 FB	11 12 03 13 14 15 16 16 17 18	* BASIC INI BASICINIT	EQU EQU EDA CMP BEQ JSR CMP	ATION: * #GOODF8 F8VERSION BINIT1 COPYROM F8VERSION	;CHECK F8 ROM ; 15 IT DK? ;=>YES ;TRY COPYING TD RAMCARD
803: 803: 803: C8 803: A7 06 805: CD 83 FB 808: F0 0C C8 806: CD 83 F8 800: CD 83 F8 810: F0 04 C8	11 12 03 13 14 15 16 16 17 18 16 19	* BASIC INI BASICINIT	FIALIZ EQU LDA CMP BEQ JSR CMP BEQ BEQ	ATION: * #COODF8 F8VERSION BINIT1 COPYROM	;CHECK F8 ROM ; IS IT OK? ;⇒?YES ;≈?YEOPYING TO RAMCARD ;≈?NOW IT'S COOD
803: 803: C8: 803: A7 06 805: CD 83 FB 808: F0 0C C8 808: F0 0C C8 808: C0 78 CF 800: CD 83 FB 810: F0 04 C8 812: 78	11 12 13 14 15 16 16 17 18 14 17 18 14 19 20	* BASIC INI BASICINIT	FIALIZ EQU LDA CMP BEQ JSR CMP BEQ SEI	ATION: * #GOODF8 F8VERSION BINIT1 COPYROM F8VERSION	;CHECK F8 ROM ; 15 IT DK? ;=>YES ;TRY COPYING TD RAMCARD
903: 803: 803: 803: 803: 803: 808: 808: 808: 808: 808: 800: 800: 800: 800: 803: 810: 812: 78 813: 80	11 12 13 14 15 16 16 16 17 18 16 17 18 16 17 20 13 21	* BASIC INI BASICINIT HANG	TIALIZ EQU LDA CMP BEQ JSR CMP BEQ SEI EQU	ATION: * #CDDDF8 F8VERSION BINIT1 COPYROM F8VERSION BINIT1 *	;CHECK F8 ROM ; IS IT OK? ;=>YES ;TRY COPYING TO RAMCARD ;=>NOW IT'S GOOD ;CRASH THE SYSTEM!
BO3: BO3: <td< td=""><td>11 12 13 14 15 16 16 17 18 14 17 18 14 19 20</td><td>* BASIC INI BASICINIT HANG</td><td>FIALIZ EQU LDA CMP BEQ JSR CMP BEQ SEI</td><td>ATION: * #GOODF8 F8VERSION BINIT1 COPYROM F8VERSION</td><td>;CHECK F8 ROM ; IS IT OK? ;⇒?YES ;≈?YCE OPYING TD RAMCARD ;≈?NOW IT'S GOOD</td></td<>	11 12 13 14 15 16 16 17 18 14 17 18 14 19 20	* BASIC INI BASICINIT HANG	FIALIZ EQU LDA CMP BEQ JSR CMP BEQ SEI	ATION: * #GOODF8 F8VERSION BINIT1 COPYROM F8VERSION	;CHECK F8 ROM ; IS IT OK? ;⇒?YES ;≈?YCE OPYING TD RAMCARD ;≈?NOW IT'S GOOD
303: 203: 2803: 2	11 12 13 14 14 15 16 16 17 18 16 17 18 16 17 18 16 17 20 13 21 22 23	* BASIC INI BASICINIT HANC *	TIALIZ EQU LDA CMP BEQ JSR CMP BEQ SEI EQU JMP	ATION: * #CDDDF8 F8VERSION BINIT1 COPYROM F8VERSION BINIT1 *	;CHECK F8 ROM ; IS IT OK? ;=>YES ;TRY COPYING TO RAMCARD ;=>NOW IT'S GOOD ;CRASH THE SYSTEM!
303: 303: CB 303: A A A 303: A A A A 303: A A A B	11 12 13 14 15 16 16 17 18 16 17 18 16 20 13 21 22 23 16 24	* BASIC INI BASICINIT HANG * BINIT1	TIALIZ EQU LDA CMP BEQ JSR CMP BEQ SEI EQU JMP	ATION: * #COODF8 FBVERSION BINIT1 COPYROM FBVERSION BINIT1 * HANC *	;CHECK F8 ROM ; IS IT OK? ;=>YES ;TRY COPYING TO RAMCARD ;=>NOW IT'S COOD ;CRASH THE SYSTEM! ;HANG FOREVER
303: 303: 28 303: 706 303: 28 303: 706 83 78 305: 70 83 78 308: 70 76 26 304: 70 78 26 304: 70 78 78 310: 70 83 78 310: 70 83 78 312: 78 312: 78 313: C8 313: 78 313: C8 314: 78 314: C8 314: 78 314: 70 78 314:	11 12 13 14 14 16 16 16 16 17 18 16 17 18 16 20 13 21 23 16 23	* BASIC INI BASICINIT HANG * BINIT1	TIALIZ EQU LDA CMP BEQ JSR CMP BEQ SEI EQU LDA	ATION: * #CDODF8 FRVERSION BINIT1 COPYROM FSVERSION BINIT1 * HANG * *CCN00	;CHECK F8 ROM ; IS IT OK? ;=>YES ;TRY COPYING TO RAMCARD ;=>NOW IT'S GOOD ;CRASH THE SYSTEM!
903: 903: CB 903: A 9 06 903: A 9 06 905: CD B3 FB 906: CD B3 FB 900: CD B3 FB 910: F0 0C CB 910: F0 04 CB 911: CB 913: 4C 13 CB 916: 916: CB 916: CB 9	11 12 13 14 15 16 16 16 17 18 16 17 20 13 21 22 23 16 24 25 26 26	* BASIC INI BASICINIT HANG * BINIT1	TIALIZ EQU LDA CMP BEG JSR CMP BEG SEI EQU JMP EQU LDA STA	* CODF8 #CCODF8 F8VERSION BINIT1 COPYROM F8VERSION BINIT1 * HANG * * CCN00 CSWH	;CHECK F8 ROM ; 15 IT OK? ;=>YES ;TRY COPYING TO RAMCARD ;=>NOW IT'S COOD ;CRASH THE SYSTEM! ;HANG FOREVER ;SET HOOKS FOR
303: 303: CB 303: A 06 305: CD B3 FB 306: CO CB 306 310: CD B3 FB 310: CD B3 FB 313: CB 313: CB 314: C1 CB 314: 314: B 37 314:	11 12 13 14 14 16 16 16 16 17 18 16 17 18 16 20 13 21 23 16 23	* BASIC INI BASICINIT HANG * BINIT1	TIALIZ EQU LDA CMP BEQ JSR CMP BEQ SEI EQU LDA	* #GDDDF8 F8VERSION BINIT1 COPYROM F8VERSION BINIT1 * HANG * * * * CONOO CSWH KSWH	;CHECK F8 ROM ; IS IT OK? ;=>YES ;TRY COPYING TO RAMCARD ;=>NOW IT'S COOD ;CRASH THE SYSTEM! ;HANG FOREVER
303: 303: CB 303: A 06 305: CD B3 FB 306: CO CB 306 310: CD B3 FB 310: CD B3 FB 313: CB 313: CB 314: C1 CB 314: 314: B 37 314:	11 12 13 14 14 15 16 16 16 16 17 18 16 17 20 21 22 23 16 24 25 26 26 27 27 27	* BASIC INI BASICINIT HANG * BINIT1	TIALIZ EQU LDA CMP BEG JSR CMP BEG SEI EGU JMP EQU LDA STA LDA	* CODF8 #CCODF8 F8VERSION BINIT1 COPYROM F8VERSION BINIT1 * HANG * * CCN00 CSWH	;CHECK F8 ROM ; 15 IT OK? ;=>YES ;TRY COPYING TO RAMCARD ;=>NOW IT'S COOD ;CRASH THE SYSTEM! ;HANG FOREVER ;SET HOOKS FOR
BO3: BO3: CB BO3: A 06 BO3: CB BO3: A BO3: CD B3 FB BO5: CD B3 FB BO5: CD C CB BO3: A C C B10: FO B3 FB B10: FO O4 CB B13: C 13 CB B14: S S S B16: A C3 S B16: S S S B16: S S S B14: S S S B14: S S S B14: S S S B14: S S S	111 12 13 14 15 15 16 16 17 18 20 21 22 23 24 25 24 24 25 28 27 28 29 29 29 29	+ BASIC INI BASICINIT HANG * BINIT1	TIALIZ EQU LDA CMP BEQ JSR CMP BEQ SEI EQU LDA STA STA STA	ATION: * #CCODF8 FBVERSION BINITI COPYROM FBVERSION BINITI * HANG * * * *CNOO CSWH *>BASICIN KSWL	;CHECK F8 ROM ; 15 IT OK? ;=>YES ;TRY COPYING TO RAMCARD ;=>NOW IT'S COOD ;CRASH THE SYSTEM! ;HANG FOREVER ;SET HOOKS FOR
BO3: CB BO3: CB BO3: CB BO3: A BO3: A BO3: A BO3: A BO3: A BO3: A BO3: CB BO0: CO BI0: CO B10: CO B13: CB B14: CB B14: CB B14: CB B14: S7 B14: S7 B14: S37 B1	111 12 13 14 15 16 16 16 17 18 18 19 20 13 21 22 23 16 24 25 26 27 28 29 300	* BASIC INI BASICINIT HANG * BINIT1	TIALIZ EQU LDA CMP BEG JSR CMP BEG SEI JMP EQU LDA STA STA LDA	ATION: * #GCODF8 FBVERSION BINIT1 COPYROM FBVERSION BINIT1 * HANG * HANG * KSWH KSWH	;CHECK F8 ROM ; 15 IT OK? ;=>YES ;TRY COPYING TO RAMCARD ;=>NOW IT'S COOD ;CRASH THE SYSTEM! ;HANG FOREVER ;SET HOOKS FOR
BO3: BO3: CB BO3: A 06 BO3: CB BO3: A BO3: CD B3 FB BO3: A 06 BO3: A BO3: A C B S FB BO3: A C C CB S FB BO0: CD B FB S FB S CB S	11 12 13 14 15 16 16 16 17 17 18 16 17 17 28 29 28 26 26 27 28 29 29 30 31	* BASIC INI BASICINIT HANG * BINIT1	TIALIZ EQU LDA CMP BEQ STA LDA STA LDA STA LDA STA	ATION: * #CCODF8 FBVERSION BINITI COPYROM FBVERSION BINITI * #ANG * * * * * * * * * * * * *	: CHECK F8 ROM : IS IT GK? :=>YES :TRY COPYING TD RAMCARD ;=>MOW IT'S GOOD :CRASH THE SYSTEM! :HANG FOREVER :SET HOOKS FOR : IN & OUT
803: 803: CB 803: A 0.6 805: CD B.3 FB 805: CO B.3 FB 805: CO C.7 CF 800: FO 0C CO 810: FO 0C CB 810: FO 0C CB 813: CB B12: TS 814: CS ST B14: 814: CS ST B14: 814: ST ST B14: ST 814: ST ST ST ST 814: ST ST ST ST 814: ST ST ST<	111 12 13 14 15 16 16 16 17 18 18 19 20 13 21 22 23 16 24 25 26 27 28 29 300	* BASIC INI BASICINIT HANG * BINIT1	TIALIZ EQU LDA CMP BEG JSR CMP BEG SEI JMP EQU LDA STA STA LDA	ATION: * #GCODF8 FBVERSION BINIT1 COPYROM FBVERSION BINIT1 * HANG * # HANG * # CNOO CSWH KSWL W>BASICOUT	;CHECK F8 ROM ; 15 IT OK? ;=>YES ;TRY COPYING TO RAMCARD ;=>NOW IT'S COOD ;CRASH THE SYSTEM! ;HANG FOREVER ;SET HOOKS FOR
803: 810: 600: 803: 810: 600: 803: 810: 600: 80: 810: 600: 60: 813: 40: 813: 40: 813: 40: 813: 40: 814: 814: 814: 814: 814: 814: 814: 814: 83: 814: 83: 814: 83: 814: 83: 814: 83: 814: 83: 814: 83: 814: 83: 82: <	11 12 13 13 14 15 16 16 16 17 18 17 18 17 20 21 22 23 16 24 24 24 27 28 29 30 31 32 23 31 31 31 31 31 31 31 31 31 3	* BASIC INI BASICINIT HANG * BINIT1	TIALIZ EQU LDA CMP BEQU JSR CMP BEQU JSR EQU JMP EQU LDA STA STA LDA STA LDA STA	ATION: * #GCODF8 FBVERSION BINIT1 COPYROM FBVERSION BINIT1 * HANG * HANG * HANG CSWH KSWL * BASICOUT CSWL * * *	; CHECK F8 ROM ; IS IT GK? ;=>YES ;TRY COPYING TD RAMCARD ; CASH THE SYSTEM! ;HANG FOREVER ;SET HOOKS FOR ; IN & OUT ;SET FULL 40-COL WINDOW
903: 903: CB 903: Y 06 803: Y 06 803: Y 06 805: CD B3 FB 805: FO 0C CF 805: TF 0C 0C 805: TF 0C 0C 810: FO 0C 0C 812: TF 0C 0C 813: CB 0C 0C 814: CS 37 0C 814: SG 37 0C 820: AP 0C 0C 820: AP 0C 0C 824: AP 0C 0C<	11 12 13 14 15 14 15 16 16 17 18 20 21 22 23 16 24 25 24 27 27 27 28 29 30 31 32 33 33 34 34	* BASIC INI BASICINIT HANG * BINIT1	TIALIZ EQU LDA CMP JSR CMP SEI SEI LDA STA STA LDA STA LDA STA LDA	ATION: * #CCODF8 F8VERSION BINIT1 COPYROM F8VERSION BINIT1 * HANG * HANG * HANG SWH KSWL WSBASICIN KSWL * * * 0 WNDLFT * 0	; CHECK F8 ROM ; IS IT OK? ;=>YES :TRY COPYING TO RAMCARD :CRASH THE SYSTEM! :HANG FOREVER ; SET HOOKS FOR ; IN & OUT ;SET FULL 40-COL WINDOW ;ASSUME TEXT MODE
303: 300: CD B3: 300: CD 30: 300: CD 30:	11 12 13 13 14 14 15 16 16 16 16 17 18 20 23 24 25 26 27 27 27 27 28 30 31 32 33 34 35	<pre>* BASIC INI BASICINIT HANG * BINIT1</pre>	TIALIZ EQU LDA CMP BEG SEI JSR EGU LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA	ATION: * #CCODF8 FBVERSION BINITI COPYROM FBVERSION BINITI * * * * * * * * * * * * *	<pre>; CHECK F8 ROM ; IS IT DK? ;=>YES ;TRY COPYING TD RAMCARD ; CASH THE SYSTEM! ;HANG FOREVER ; SET HOOKS FOR ; IN & DUT ;SET FULL 40-COL WINDOW ;ASSUME TEXT MODE ; IN TEXT MODE?</pre>
303: 303: CB 303: 404 305: CD 83 FB 305: CD 83 FB 305: CC CB 305: CD 83 FB 305: CD B3 FD 305: CD B3 FD 305: CD CD CD FD FD 305: CD B3 FD 50: CD B3 FD 50: CD B3 FD 50: CD B3 FD 50: CD B3 FD CD B3 FD 50: CD B3 FD 50: FD FD 50: FD	11 12 13 13 14 14 15 16 16 16 16 17 18 20 23 24 25 26 27 27 27 27 28 30 31 32 33 34 35	<pre>* BASIC INI BASICINIT HANG * BINIT1</pre>	TIALIZ EQU LDA CMP BEG SEI EGU EGU EQU EQU EQU EQU EQU EQU EQU EQU EQU EQ	ATION: * #CCODF8 F8VERSION BINIT1 COPYROM F8VERSION BINIT1 * HANG * HANG * HANG SWH KSWL WSBASICIN KSWL * * * 0 WNDLFT * 0	; CHECK F8 ROM ; IS IT OK? ;=>YES :TRY COPYING TD RAMCARD ;=>NOW IT'S GOOD :CRASH THE SYSTEM! ;HANG FOREVER ;SET HOOKS FOR ; IN & OUT ;SET FULL 40-COL WINDOW ;ASSUME TEXT MODE ;IN TEXT MODE? ;=>YES
303: 300: CD B3: FB 300: 50: 50: 30: </td <td>11 12 13 14 15 16 16 16 16 17 18 20 17 27 27 27 27 27 27 27 27 27 2</td> <td><pre>* BASIC INI BASICINIT HANG * BINIT1</pre></td> <td>TIALIZ EQU LDA CMP BEG SEI JSR EGU LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA</td> <td>ATION: * #CCODF8 F8VERSION BINIT1 COPYROM F8VERSION BINIT1 * HANG * # HANG * # CSWH * * * * * * * * * * * * *</td> <td><pre>:CHECK F8 ROM : IS IT OK? :=>YES :TRY COPYING TO RAMCARD ;CRASH THE SYSTEM! :HANG FOREVER :SET HOOKS FOR ; IN & OUT ;SET FULL 40-COL WINDOW ;ASSUME TEXT MODE :IN TEXT MODE?</pre></td>	11 12 13 14 15 16 16 16 16 17 18 20 17 27 27 27 27 27 27 27 27 27 2	<pre>* BASIC INI BASICINIT HANG * BINIT1</pre>	TIALIZ EQU LDA CMP BEG SEI JSR EGU LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA	ATION: * #CCODF8 F8VERSION BINIT1 COPYROM F8VERSION BINIT1 * HANG * # HANG * # CSWH * * * * * * * * * * * * *	<pre>:CHECK F8 ROM : IS IT OK? :=>YES :TRY COPYING TO RAMCARD ;CRASH THE SYSTEM! :HANG FOREVER :SET HOOKS FOR ; IN & OUT ;SET FULL 40-COL WINDOW ;ASSUME TEXT MODE :IN TEXT MODE?</pre>
303: 303: 303: 303: 303: 303: 303: 40: 303: 40: 303: 40: 303: 40: 303: 40: 305: 10: 50:	11 12 13 14 15 16 16 16 16 17 18 20 17 27 27 27 27 27 27 27 27 27 2	* BASIC INI BASICINIT HANG * BINITI	TIALIZ EQU LDA CMP BEG JSRP BEG JSRP BEG JMP EQU LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA	ATION: * #CCODF8 F8VERSION BINIT1 COPYROM F8VERSION BINIT1 * HANG * # HANG * # CSWH * * * * * * * * * * * * *	; CHECK F8 ROM ; IS IT OK? ;=>YES :TRY COPYING TD RAMCARD ;=>NOW IT'S GOOD :CRASH THE SYSTEM! ;HANG FOREVER ;SET HOOKS FOR ; IN & OUT ;SET FULL 40-COL WINDOW ;ASSUME TEXT MODE ;IN TEXT MODE? ;=>YES
303: 203: 803: CB 803: CB 803: CB 803: CD 803: CD 805: CO 805: CO 800: CO 800: CO 800: CO 800: CO 800: CO 800: CO 810: FO 810: FO 810: CO 811: CS 813: CS 813: CS 814: FO 816: FO 816: FO 816: FO 816: FO 820: FO 824: FO 824: <td>11 12 13 14 15 14 15 16 16 17 17 17 17 20 19 20 21 21 23 24 25 27 27 27 27 27 27 27 27 27 27</td> <td>+ BASIC INI BASICINIT HANC * BINITI</td> <td>TIALIZ EQU LDA CMP BEG JSP BEG JCMP BEG LDA STA STA LDA LDA STA BHI BMI BMI BMI BMI BMI BMI BMA CMA</td> <td>ATION: * #CCODF8 F8VERSION BINIT1 COPYROM F8VERSION BINIT1 * HANG * * HANG * * * CSWH *SBASICOUT CSWL *D * * * * * * * * * * * * *</td> <td>; CHECK F8 ROM ; IS IT OK? ;=>YES :TRY COPYING TD RAMCARD ;=>NOW IT'S GOOD :CRASH THE SYSTEM! ;HANG FOREVER ;SET HOOKS FOR ; IN & OUT ;SET FULL 40-COL WINDOW ;ASSUME TEXT MODE ;IN TEXT MODE? ;=>YES</td>	11 12 13 14 15 14 15 16 16 17 17 17 17 20 19 20 21 21 23 24 25 27 27 27 27 27 27 27 27 27 27	+ BASIC INI BASICINIT HANC * BINITI	TIALIZ EQU LDA CMP BEG JSP BEG JCMP BEG LDA STA STA LDA LDA STA BHI BMI BMI BMI BMI BMI BMI BMA CMA	ATION: * #CCODF8 F8VERSION BINIT1 COPYROM F8VERSION BINIT1 * HANG * * HANG * * * CSWH *SBASICOUT CSWL *D * * * * * * * * * * * * *	; CHECK F8 ROM ; IS IT OK? ;=>YES :TRY COPYING TD RAMCARD ;=>NOW IT'S GOOD :CRASH THE SYSTEM! ;HANG FOREVER ;SET HOOKS FOR ; IN & OUT ;SET FULL 40-COL WINDOW ;ASSUME TEXT MODE ;IN TEXT MODE? ;=>YES
BO3: CB BO3: A CA BO3: CD B3 FB BO5: CD B3 FB BO5: CD B3 FB BO5: CD CB FB BO5: CD B3 FB BO5: CD B3 FB BO0: CD B3 FB BO0: CD B3 FB BI0: FD CH CB B12: TB CB B13: CB B14: CS 37 B14: S3 B14: S3 S4 CS B22: S4 B14: S4 S5 CD B22: S4 CD B22: </td <td>11 12 13 14 15 14 15 16 16 16 16 17 18 18 10 19 20 21 23 13 24 25 26 27 27 28 29 30 31 32 29 30 31 34 35 29 31 34 35 29 31 34 35 35 35 35 35 35 35 35 35 35</td> <td>* BASIC INI BASICINIT HANG * BINITI</td> <td>TIALIZ EQU LDA CMP BEG JSR BEG JCMP BEG JCMP BEG LDA STA STA STA STA LDA STA STA LDA STA BMI LDA CMP EQU EQU STA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA STA LDA STA LDA STA LDA STA STA LDA STA STA LDA STA STA LDA STA STA STA STA STA STA STA STA STA ST</td> <td>ATION: * #CCODF8 FBVERSION BINIT1 COPYBOM FBVERSION BINIT1 * + HANG * * COND CSWH * * * * * * * * * * * * *</td> <td>; CHECK F8 ROM ; IS IT OK? ;=>YES :TRY COPYING TD RAMCARD ;=>NOW IT'S GOOD :CRASH THE SYSTEM! ;HANG FOREVER ;SET HOOKS FOR ; IN & OUT ;SET FULL 40-COL WINDOW ;ASSUME TEXT MODE ;IN TEXT MODE? ;=>YES</td>	11 12 13 14 15 14 15 16 16 16 16 17 18 18 10 19 20 21 23 13 24 25 26 27 27 28 29 30 31 32 29 30 31 34 35 29 31 34 35 29 31 34 35 35 35 35 35 35 35 35 35 35	* BASIC INI BASICINIT HANG * BINITI	TIALIZ EQU LDA CMP BEG JSR BEG JCMP BEG JCMP BEG LDA STA STA STA STA LDA STA STA LDA STA BMI LDA CMP EQU EQU STA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA STA LDA STA LDA STA LDA STA STA LDA STA STA LDA STA STA LDA STA STA STA STA STA STA STA STA STA ST	ATION: * #CCODF8 FBVERSION BINIT1 COPYBOM FBVERSION BINIT1 * + HANG * * COND CSWH * * * * * * * * * * * * *	; CHECK F8 ROM ; IS IT OK? ;=>YES :TRY COPYING TD RAMCARD ;=>NOW IT'S GOOD :CRASH THE SYSTEM! ;HANG FOREVER ;SET HOOKS FOR ; IN & OUT ;SET FULL 40-COL WINDOW ;ASSUME TEXT MODE ;IN TEXT MODE? ;=>YES
803: 603: 603: 603: 603: 605: <td< td=""><td>11 12 13 13 14 15 14 14 15 16 16 16 16 17 18 18 10 10 21 20 21 20 21 20 21 20 21 20 21 20 21 20 21 20 21 20 20 21 20 20 21 20 20 21 20 20 21 20 20 21 20 20 21 20 20 21 20 20 20 20 20 20 20 20 20 20</td><td>+ BASIC INI BASICINIT HANG * BINITI BINITIA</td><td>TIALIZ EQU LDA CMP BEGR BEGR J GUA STA LDA STA LDA STA BITI LDA STA BITI LDA STA STA STA STA STA STA STA ST</td><td>ATION: * #CCODF8 FBVERSION BINIT1 COPYROM FBVERSION BINIT1 * HANG * * HANG * * * * * * * * * * * * *</td><td>; CHECK F8 ROM ; IS IT OK? ;=>YES :TRY COPYING TD RAMCARD ;=>NOW IT'S GOOD :CRASH THE SYSTEM! ;HANG FOREVER ;SET HOOKS FOR ; IN & OUT ;SET FULL 40-COL WINDOW ;ASSUME TEXT MODE ;IN TEXT MODE? ;=>YES</td></td<>	11 12 13 13 14 15 14 14 15 16 16 16 16 17 18 18 10 10 21 20 21 20 21 20 21 20 21 20 21 20 21 20 21 20 21 20 20 21 20 20 21 20 20 21 20 20 21 20 20 21 20 20 21 20 20 21 20 20 20 20 20 20 20 20 20 20	+ BASIC INI BASICINIT HANG * BINITI BINITIA	TIALIZ EQU LDA CMP BEGR BEGR J GUA STA LDA STA LDA STA BITI LDA STA BITI LDA STA STA STA STA STA STA STA ST	ATION: * #CCODF8 FBVERSION BINIT1 COPYROM FBVERSION BINIT1 * HANG * * HANG * * * * * * * * * * * * *	; CHECK F8 ROM ; IS IT OK? ;=>YES :TRY COPYING TD RAMCARD ;=>NOW IT'S GOOD :CRASH THE SYSTEM! ;HANG FOREVER ;SET HOOKS FOR ; IN & OUT ;SET FULL 40-COL WINDOW ;ASSUME TEXT MODE ;IN TEXT MODE? ;=>YES
803: 603: 603: 603: 603: 605: <td< td=""><td>11 12 13 14 15 14 15 16 16 16 16 17 18 18 10 19 20 21 23 13 24 25 26 27 27 28 29 30 31 32 29 30 31 34 35 29 31 34 35 29 31 34 35 35 35 35 35 35 35 35 35 35</td><td>* BASIC INI BASICINIT HANG * BINITI</td><td>TIALIZ EQU LDA CMP BEG JSR BEG JCMP BEG JCMP BEG LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA LDA CMP BEG U JMP EQU STA LDA STA</td><td>ATION: * #CCODF8 FBVERSION BINIT1 COPYBOM FBVERSION BINIT1 * HANG * #CCN00 CSWH * *CCN00 CSWH * * * * * COV CSWH * * * * * * * * * * * * *</td><td><pre>:CHECK F8 ROM : IS IT OK? :=>YES :TRY COPYING TO RAMCARD ;=>NOW IT'S GOOD :CRASH THE SYSTEM! :HANG FOREVER :SET HOOKS FOR ; IN & OUT :SET FULL 40-COL WINDOW :ASSUME TEXT MODE :IN TEXT MODE? :=>YES ;IF GR, SET 4 LINES</pre></td></td<>	11 12 13 14 15 14 15 16 16 16 16 17 18 18 10 19 20 21 23 13 24 25 26 27 27 28 29 30 31 32 29 30 31 34 35 29 31 34 35 29 31 34 35 35 35 35 35 35 35 35 35 35	* BASIC INI BASICINIT HANG * BINITI	TIALIZ EQU LDA CMP BEG JSR BEG JCMP BEG JCMP BEG LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA LDA CMP BEG U JMP EQU STA LDA STA	ATION: * #CCODF8 FBVERSION BINIT1 COPYBOM FBVERSION BINIT1 * HANG * #CCN00 CSWH * *CCN00 CSWH * * * * * COV CSWH * * * * * * * * * * * * *	<pre>:CHECK F8 ROM : IS IT OK? :=>YES :TRY COPYING TO RAMCARD ;=>NOW IT'S GOOD :CRASH THE SYSTEM! :HANG FOREVER :SET HOOKS FOR ; IN & OUT :SET FULL 40-COL WINDOW :ASSUME TEXT MODE :IN TEXT MODE? :=>YES ;IF GR, SET 4 LINES</pre>
BO3: BO3: CB BO3: A 06 BO3: CB BO3: A BO3: A 06 BO3: A BO3: A 06 BO3: A BO3: A 06 BO3: A BO3: A C B B BO1: FD OA CB B B10: FO OA CB B B C B B13: A 13 CB B CB CB CB B CB CB B CB B CB B CB B CB B CB CB B CB CB CB B CB CB B CB CB CB CB <t< td=""><td>11 12 13 14 15 16 17 18 19 10 11 14 15 16 17 18 19 21 21 22 23 24 25 26 27 28 29 30 31 35 31 36 31 32 41 42</td><td>* BASIC INI BASICINIT HANG * BINITI</td><td>TIALIZ EQU LDA BEG SEJ SR CMP BEG SEJ EGU LDA STA STA LDA STA LDA BIT LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA</td><td>ATION: * # GCODF8 # F8VERSION BINIT1 COPYROM F8VERSION BINIT1 * HANG * # CCN00 CSWH KSWL * * * CCN00 CSWH KSWL * * * * * * * * * * * * *</td><td><pre>:CHECK F8 ROM : IS IT OK? :=>YES :TRY COPYING TO RAMCARD ;=>NOW IT'S GOOD :CRASH THE SYSTEM! :HANG FOREVER :SET HOOKS FOR ; IN & OUT :SET FULL 40-COL WINDOW :ASSUME TEXT MODE :IN TEXT MODE? :=>YES ;IF GR, SET 4 LINES</pre></td></t<>	11 12 13 14 15 16 17 18 19 10 11 14 15 16 17 18 19 21 21 22 23 24 25 26 27 28 29 30 31 35 31 36 31 32 41 42	* BASIC INI BASICINIT HANG * BINITI	TIALIZ EQU LDA BEG SEJ SR CMP BEG SEJ EGU LDA STA STA LDA STA LDA BIT LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA	ATION: * # GCODF8 # F8VERSION BINIT1 COPYROM F8VERSION BINIT1 * HANG * # CCN00 CSWH KSWL * * * CCN00 CSWH KSWL * * * * * * * * * * * * *	<pre>:CHECK F8 ROM : IS IT OK? :=>YES :TRY COPYING TO RAMCARD ;=>NOW IT'S GOOD :CRASH THE SYSTEM! :HANG FOREVER :SET HOOKS FOR ; IN & OUT :SET FULL 40-COL WINDOW :ASSUME TEXT MODE :IN TEXT MODE? :=>YES ;IF GR, SET 4 LINES</pre>
803: 803: <td< td=""><td>11 12 13 14 15 14 15 14 15 16 17 18 16 17 18 19 21 22 231 21 22 231 24 25 31 32 31 38 40 41 42 43</td><td>* BASIC INI BASICINIT HANG * BINITI</td><td>TIALIZ EQU LDA EQU LDA BEQ EQU LDA EQU LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA</td><td>ATION: * #CCODF8 FBVERSION BINITI DINITI BINITI * HANG * * HANG * * * CNCO CSWH * * * * * * * * * * * * *</td><td><pre>:CHECK F8 ROM : IS IT OK? :=>YES :TRY COPYING TO RAMCARD ;=>NOW IT'S GOOD :CRASH THE SYSTEM! :HANG FOREVER :SET HOOKS FOR ; IN & DUT ;SET FULL 40-COL WINDOW ;ASSUME TEXT MODE ;IN TEXT MODE? :=>YES ;IF GR, SET 4 LINES</pre></td></td<>	11 12 13 14 15 14 15 14 15 16 17 18 16 17 18 19 21 22 231 21 22 231 24 25 31 32 31 38 40 41 42 43	* BASIC INI BASICINIT HANG * BINITI	TIALIZ EQU LDA EQU LDA BEQ EQU LDA EQU LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA	ATION: * #CCODF8 FBVERSION BINITI DINITI BINITI * HANG * * HANG * * * CNCO CSWH * * * * * * * * * * * * *	<pre>:CHECK F8 ROM : IS IT OK? :=>YES :TRY COPYING TO RAMCARD ;=>NOW IT'S GOOD :CRASH THE SYSTEM! :HANG FOREVER :SET HOOKS FOR ; IN & DUT ;SET FULL 40-COL WINDOW ;ASSUME TEXT MODE ;IN TEXT MODE? :=>YES ;IF GR, SET 4 LINES</pre>
803: 803: <td< td=""><td>11 12 13 14 15 16 16 17 18 19 11 18 16 17 18 19 21 223 231 24 27 28 29 304 31 35 31 34 31 34 43 43 43</td><td>* BASIC INI BASICINIT HANG * BINITI</td><td>TIALIZ EQU LDA CMP BEQ CMP BEQ CMP EQU JSR CMP EQU JSR CMP EQU JSR CMP EQU STA STA STA LDA STA LDA STA LDA STA LDA STA</td><td>ATION: * #CCODF8 F8VERSION BINIT1 COPYROM F8VERSION BINIT1 * HANG * * HANG * * HANG * * * CNOO CSWH * * * * * * * * * * * * *</td><td><pre>:CHECK F8 ROM : IS IT OK? :=>YES :TRY COPYING TO RAMCARD ;=>NOW IT'S GOOD :CRASH THE SYSTEM! :HANG FOREVER :SET HOOKS FOR ; IN & OUT :SET FULL 40-COL WINDOW :ASSUME TEXT MODE :IN TEXT MODE? :=>YES ;IF GR, SET 4 LINES</pre></td></td<>	11 12 13 14 15 16 16 17 18 19 11 18 16 17 18 19 21 223 231 24 27 28 29 304 31 35 31 34 31 34 43 43 43	* BASIC INI BASICINIT HANG * BINITI	TIALIZ EQU LDA CMP BEQ CMP BEQ CMP EQU JSR CMP EQU JSR CMP EQU JSR CMP EQU STA STA STA LDA STA LDA STA LDA STA LDA STA	ATION: * #CCODF8 F8VERSION BINIT1 COPYROM F8VERSION BINIT1 * HANG * * HANG * * HANG * * * CNOO CSWH * * * * * * * * * * * * *	<pre>:CHECK F8 ROM : IS IT OK? :=>YES :TRY COPYING TO RAMCARD ;=>NOW IT'S GOOD :CRASH THE SYSTEM! :HANG FOREVER :SET HOOKS FOR ; IN & OUT :SET FULL 40-COL WINDOW :ASSUME TEXT MODE :IN TEXT MODE? :=>YES ;IF GR, SET 4 LINES</pre>
803: 803: CB 803: CB 803: CD B3 803: CD B3 804: CO CB 804: CO CB 804: CO CB 804: CO CB 810: FO CB 813: CB SI 814: CB SI 814: CB SI 814: S3 SP 824: S0 S2 824: S0 S2 824: S0 S2 824: S0 S2 833: S4 S3 833: S3 S2 837: S2 S37	11 12 13 14 15 14 15 14 15 14 15 14 15 16 17 18 19 21 22 23 21 22 231 24 25 31 32 31 32 41 42 43 443 445	* BASIC INI BASICINIT HANG * BINITI	TIALIZ EQU LDA CMP BEQ CMP EQU SEJ SR CMP EQU LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA STA LDA LDA	ATION: * #CCODF8 F8VERSION BINIT1 COPYROM F8VERSION BINIT1 * HANG * * HANG * * HANG * * * CNOO CSWH * * * * * * * * * * * * *	<pre>:CHECK F8 ROM ; IS IT GK? ;=>YES ;TRY COPYING TD RAMCARD ;=>NOW IT'S GOOD :CRASH THE SYSTEM! ;HANG FOREVER ;SET HOOKS FOR ; IN & OUT ;SET FULL 40-COL WINDOW ;ASSUME TEXT MODE ;IN TEXT MODE? ;=>YES ;IF GR, SET 4 LINES ;COPY USER CH ; AS 'OLD' SETTING ;GET READY TO CLEAR</pre>
803: 803: 76 803: 76 803: 807 803: 78 804: 70 805: 00 807: 78 810: 78 811: 70 813: 40 813: 40 814: 73 814: 73 814: 73 814: 73 814: 73 814: 83 815: 63 816: 73 814: 83 816: 83 816: 83 816: 83 822: 83 824: 80 824: 80 824: 80 824: 80 824: 80 831: 82 833: 82 833: 82 833: 82	11 12 13 14 15 14 15 16 17 18 16 17 18 19 21 22 23 16 24 25 26 27 31 32 31 331 34 43 43 43 43 45 47 47	* BASIC INI BASICINIT HANG * BINITI	TIALIZ EQU LDA CMP BEQ CMP BEQ CMP EQU JSR CMP EQU JSR CMP EQU JSR CMP EQU STA STA STA LDA STA LDA STA LDA STA LDA STA	ATION: * #COODF8 F8VERSION BINIT1 COPYROM F8VERSION BINIT1 * HANG * # CCN00 CSWH * * * CCN00 CSWH * * * * CCN00 CSWH * * * * * * * * * * * * *	<pre>; CHECK F8 RDM ; IS IT DK? ;=>YES ;TRY COPYING TD RAMCARD ;=>NOW IT'S COOD ; CRASH THE SYSTEM! ;HANG FOREVER ; SET HOOKS FOR ; IN & OUT ; SET FULL 40-COL WINDOW ;ASSUME TEXT MODE ; IN TEXT MODE? ;=>YES ; IF GR, SET 4 LINES ;COPY USER CH ; AS 'OLD' SETTING ;GET READY TO CLEAR ;PRESERVE ING STATUS</pre>
BO3: CB BO3: CB BO3: CB BO3: CB BO3: CB BO3: CD BO3: CD BO5: CO BO5: CO BO5: CO BO5: CO BO1: CD BO1: CO BO3: AF BI0: CO BI1: CB B13: CC B14: S B16: S B16: S B16: S B16: S B16: S B16: S B20: AF B20: S B31: S </td <td>11 12 13 14 15 15 16 17 18 19 11 15 16 17 18 19 11 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 22 230 31 34 43 443 445 446 47 48</td> <td>+ BASIC INI BASICINIT HANG * BINITI</td> <td>TIALLZ EQU LDA CMP BEG SEJ JSR CMP BEG SEJ JMP EQU JMP EQU JMP EQU JMP EQU LDA STA LDA LDA STA LDA STA STA STA AND STA</td> <td>ATION: * #COODF8 FBVERSION BINITI BINITI * HANG * * HANG * * CNOO CSWH * SASICIN KSWH * SASICIN KSWH * * * * * * * * * * * * *</td> <td><pre>:CHECK F8 ROM ; IS IT GK? ;=>YES ;TRY COPYING TD RAMCARD ;=>NOW IT'S COOD :CRASH THE SYSTEM! ;HANG FOREVER ;SET HOOKS FOR ; IN & OUT ;SET FULL 40-COL WINDOW ;ASSUME TEXT MODE ;IN & OUT ;SET FULL 40-COL WINDOW ;ASSUME TEXT MODE ;IN TEXT MODE? ;=>YES ;IF GR, SET 4 LINES ;COPY USER CH ;ASSUME TEXT MODE ;COPY USER CH ;ASSUME TEXT MODE ;CEAR MODES</pre></td>	11 12 13 14 15 15 16 17 18 19 11 15 16 17 18 19 11 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 22 230 31 34 43 443 445 446 47 48	+ BASIC INI BASICINIT HANG * BINITI	TIALLZ EQU LDA CMP BEG SEJ JSR CMP BEG SEJ JMP EQU JMP EQU JMP EQU JMP EQU LDA STA LDA LDA STA LDA STA STA STA AND STA	ATION: * #COODF8 FBVERSION BINITI BINITI * HANG * * HANG * * CNOO CSWH * SASICIN KSWH * SASICIN KSWH * * * * * * * * * * * * *	<pre>:CHECK F8 ROM ; IS IT GK? ;=>YES ;TRY COPYING TD RAMCARD ;=>NOW IT'S COOD :CRASH THE SYSTEM! ;HANG FOREVER ;SET HOOKS FOR ; IN & OUT ;SET FULL 40-COL WINDOW ;ASSUME TEXT MODE ;IN & OUT ;SET FULL 40-COL WINDOW ;ASSUME TEXT MODE ;IN TEXT MODE? ;=>YES ;IF GR, SET 4 LINES ;COPY USER CH ;ASSUME TEXT MODE ;COPY USER CH ;ASSUME TEXT MODE ;CEAR MODES</pre>
BO3: BO3: CB BO3: A 06 BO5: CD B3 FB BO5: CD B3 FB BO5: CD B3 FB BO0: CD B3 FB B10: FO A CB B12: TB SB SB B13: CB B12: TB B14: S3 S7 B12: B14: S3 S9 B12: B14: S3 S9 B2 B14: S3 S9 B2 B2: S4 O0 B2 B2: S4 O0 B2 B2: S4 O0 B2 B2: S2: B3 B3 B3: S2	11 12 13 14 15 14 15 16 17 18 16 17 18 19 201 12 21 22 23 24 25 26 27 30 28 29 30 31 35 31 34 43 43 44 45 46 47 48 47 48	* BASIC INI BASICINIT HANG * BINITI	TIALLZ EQU LDA CMP BEQ SEJ JSR CMP EQU JSR CMP EQU JSR EQU JMP EQU LDA STA STA STA LDA STA LDA STA LDA STA LDA STA LDA AND	ATION: * #COODF8 F8VERSION BINIT1 COPYROM F8VERSION BINIT1 * HANG * #CCNOO CSWH * * * * CSWOO CSWH * * * * * * * * * * * * *	<pre>; CHECK F8 RDM ; IS IT DK? ;=>YES ;TRY COPYING TD RAMCARD ;=>NOW IT'S COOD ; CRASH THE SYSTEM! ;HANG FOREVER ; SET HOOKS FOR ; IN & OUT ; SET FULL 40-COL WINDOW ;ASSUME TEXT MODE ; IN TEXT MODE? ;=>YES ; IF GR, SET 4 LINES ;COPY USER CH ; AS 'OLD' SETTING ;GET READY TO CLEAR ;PRESERVE ING STATUS</pre>
303: CB 303: CB 303: CB 303: CD 305: CD 305: CD 306: CO 307: CD 310: CD 313: CC 314: CC 316: SC 322: SC 324: AP 324: SC 324: SC 324: SC 325: SC 326: SC 327:	11 12 13 14 15 15 16 17 18 19 11 15 16 17 18 19 11 21 22 23 21 22 231 324 331 34 43 443 443 445 446 47 48 49 50	+ BASIC INI BASICINIT HANG * BINITI	TIALLZ EQU LDA CMP BEG SEJ JSR CMP BEG SEJ JMP EQU JMP EQU JMP EQU JMP EQU LDA STA LDA LDA STA LDA STA STA STA AND STA	ATION: * #COODF8 FBVERSION BINITI BINITI * HANG * * HANG * * CNOO CSWH * SASICIN KSWH * SASICIN KSWH * * * * * * * * * * * * *	<pre>:CHECK F8 ROM ; IS IT GK? ;=>YES ;=YES ;TRY COPYING TD RAMCARD ;=>NOW IT'S COOD :CRASH THE SYSTEM! ;HANG FOREVER ;SET HOOKS FOR ; IN & OUT ;SET FULL 40-COL WINDOW ;ASSUME TEXT MODE ;IN & OUT ;SET FULL 40-COL WINDOW ;ASSUME TEXT MODE ;IN TEXT MODE? ;=>YES ;IF GR, SET 4 LINES ;COPY USER CH ;AS 'OLD' SETTING ;GET READY TO CLEAR ;PRESERVE ING STATUS ;CLEAR MODES</pre>
303: CB 303: CB 303: CB 303: CD 300: CD 300: CD 310: CB 311: CD 313: CD 314: SD 316: SD 320: SD 320: SD 320: SD 331: SD 331: SD 331:	11 12 13 14 15 16 16 17 18 16 17 18 19 201 21 22 23 16 27 28 29 30 31 32 31 34 31 38 42 43 44 45 46 47 48 49 50	<pre>* BASIC INI BASICINIT HANG * BINITI BINITIA</pre>	TIALIZ EQU LDA CMP BEQ CMP BEQ CMP EQU JMP EQU JMP EQU LDA STA STA LDA STA STA LDA STA STA LDA STA STA LDA STA STA LDA STA STA STA STA STA STA STA STA STA ST	ATION: * #COODF8 F8VERSION BINIT1 COPYROM F8VERSION BINIT1 * HANG * HANG * HANG CSWH KSWL * * CSWH KSWL * * CSWH * * CSWH * * CSWH * * CSWH * * * * * * * * * * * * *	<pre>:CHECK F8 ROM ; IS IT GK? ;=>YES ;=YES ;TRY COPYING TD RAMCARD ;=>NOW IT'S COOD :CRASH THE SYSTEM! ;HANG FOREVER ;SET HOOKS FOR ; IN & OUT ;SET FULL 40-COL WINDOW ;ASSUME TEXT MODE ;IN & OUT ;SET FULL 40-COL WINDOW ;ASSUME TEXT MODE ;IN TEXT MODE? ;=>YES ;IF GR, SET 4 LINES ;COPY USER CH ;AS 'OLD' SETTING ;GET READY TO CLEAR ;PRESERVE ING STATUS ;CLEAR MODES</pre>
303: CB 303: CB 303: CB 303: CB 303: CD 303: F0 303: F0 305: CD B3 303: F0 CC 304: F0 CC 300: F0 CC 300: CD B3 300: CD B3 301: CD B3 301: C13 CB 311: C3 F3 314: F3 37 314: F3 38 320: A7 07 322: F3 50 320: A7 14 331: F3 23 </td <td>11 12 13 14 15 16 16 17 18 16 17 18 19 201 21 22 23 16 27 28 29 30 31 32 31 34 31 38 42 43 44 45 46 47 48 49 50</td> <td><pre>* BASIC INI BASICINIT HANG * BINITI BINITIA * * PASCAL 1.</pre></td> <td>TIALIZ EQU LDA CMP BEQ CMP BEQ CMP EQU JMP EQU JMP EQU LDA STA STA LDA STA STA LDA STA STA LDA STA STA LDA STA STA LDA STA STA STA STA STA STA STA STA STA ST</td> <td>ATION: * #COODF8 F8VERSION BINIT1 COPYROM F8VERSION BINIT1 * HANG * HANG * HANG CSWH KSWL * * CSWH KSWL * * CSWH * * CSWH * * CSWH * * CSWH * * * * * * * * * * * * *</td> <td><pre>:CHECK F8 ROM ; IS IT OK? ;=>YES ;TRY COPYING TD RAMCARD ;CRASH THE SYSTEM! ;HANG FOREVER ;SET HOOKS FOR ; IN & OUT ;SET FULL 40-COL WINDOW ;ASSUME TEXT MODE ;IN TEXT MODE? ;=>YES ;IF GR. SET 4 LINES ;COPY USER CH ;ASSUME TEXT MODE? ;=>YES ;IF GR. SET 4 LINES</pre></td>	11 12 13 14 15 16 16 17 18 16 17 18 19 201 21 22 23 16 27 28 29 30 31 32 31 34 31 38 42 43 44 45 46 47 48 49 50	<pre>* BASIC INI BASICINIT HANG * BINITI BINITIA * * PASCAL 1.</pre>	TIALIZ EQU LDA CMP BEQ CMP BEQ CMP EQU JMP EQU JMP EQU LDA STA STA LDA STA STA LDA STA STA LDA STA STA LDA STA STA LDA STA STA STA STA STA STA STA STA STA ST	ATION: * #COODF8 F8VERSION BINIT1 COPYROM F8VERSION BINIT1 * HANG * HANG * HANG CSWH KSWL * * CSWH KSWL * * CSWH * * CSWH * * CSWH * * CSWH * * * * * * * * * * * * *	<pre>:CHECK F8 ROM ; IS IT OK? ;=>YES ;TRY COPYING TD RAMCARD ;CRASH THE SYSTEM! ;HANG FOREVER ;SET HOOKS FOR ; IN & OUT ;SET FULL 40-COL WINDOW ;ASSUME TEXT MODE ;IN TEXT MODE? ;=>YES ;IF GR. SET 4 LINES ;COPY USER CH ;ASSUME TEXT MODE? ;=>YES ;IF GR. SET 4 LINES</pre>

80-Column Firmware Listing

C84C:00 C84D: S	,	0000	55 56 57		BRK IFNE FAIL	*-\$C84D 2, 'C84D	ERR IF WRONG ADDR Hook Alignment'
CB4D CB4D: 40	51	сз	58 59		FIN	JPREAD	=>00 TO STANDARD READ
C850: C850:			60	*			
C850:				* IS THERE	A CARD	?	
C850: C850: 20		C850		BINIT2	EGU	* TESTCARD	SEE IF CARD PLUGGED IN
C853: DC	90 (66 67		BNE	CLEARIT	JET S 40
C857: 80	01		68		STA	SETBOCOL	IENABLE 80 STORE
C85A: 80 C85D:	00	0	69 70	•	STA	SETSOVID	AND BO VIDED
C85D: C85D:			72	* HOME & CLI *			
C85D: C85D: 81) OF	C85D CO	74	CLEARIT	EQU	* SETALTCHAR	SET NORM/INV LCASE
CB60: 20 CB63: 28	3	CD	75 76		JSR PLP	X.FF	CLEAR IT
C864:16 C865:08			77 78		CLC		ACLC ASSURES THAT A WE PRINT THIS A INITIAL CHARACTER
C866: C866;			80		FEOR	INTEGER BASI	
C866: C866:			82	* HITTING	DF \$CO	DO ON INITIA	ENTRY:
C866:		6885		CBBASIC	EQU	*	BASIC IN/OUT
C866: 20 C869: 10	09	C874	86		BPL	RDBOVID CBB2	;=>40. LEAVE ALONE
C86B: 81 C86E:	0 01	0		*	STA	SETBOCOL	80. ENABLE STORE.
C86E: C86E:			90	* AN EVEN I	SCROLI	ING WINDOW	IS
C86E: A5	5 21		91 92	*	LDA	WNDWDTH	
C870: 29 C872: 85	FE		93 94		AND	#\$FE WNDWDTH	ROUND IT TO LOWER EVEN
C874:				* COPY USER	'S CH	IF IT DIFFER	5 FROM
C874: C874:			97	* WHAT WE I	LAST P	JT THERE:	
C874: C874: A5		C874		C882		* CH	GET IT
C876: CI C879: FC	79	04	101		CMP	OLDCH CBB3	; IS IT THE SAME? ;=>YES, USE OUR OWN
C878: 81		05	103		STA	OURCH	;=>NO, USE HIS
C87E: C87E: A9	06	C87E	105	C883	EQU	#GODDF8	CHECK F8 ROM
C880: CI C883: FC) 93) 08	FB C890	106 107		CMP BEG	F8VERSION C884	; IF DIFFERENT, USER ; HAS RELOADED RAMCARD
C885: C885:			109		ом то н	LANG CARD:	
C885: C885: 20			110 111	*	JSR	COPYROM	COPY IT AGAIN
C888: CI C888: FC) B3	FB C890	112 113		CMP BEQ	F8VERSION C884	; IS IT NOW CORRECT? ;=>GREAT
C88D: 40 C890:	: 13	CB	114 115	*	JMP	HANG	;=>WE HAVE WRONG ROM!
C890: C890: 28	3	C890	116	C884	EQU	*	;RECOVER CARRY (IN/DUT)
C891:90 C893:40	03	C896	118 119		BCC	BINPUT	; ≈>PRINT A CHAR ; =>INPUT A CHAR
C896: C896: AI		C896	120	BOUT	EQU	* MODE	SAY THAT WE'RE
C899:29 C898:81	BF		122		AND		JT ; PRINTING
C89E: 40	A1	C8	123		JMP	BPRINT	;->OUTPUT A CHAR
CBA1: CBA1:			2			DE BPRINT	
CBA1: CBA1:			4	* BASIC OUT	~~~~~		
CBA1: CBA1: AI) 7B	C8A1 06	6	BPRINT	EQU	* CHAR	GET CHARACTER
C8A4: C9			7		CMP BNE	#\$8D NOWAIT	;IS IT C/R? ;=>NOPE, NO VIDWAIT
CBAB: AC	00	C8C0	9 10		LDY BPL	KBD NOWAIT	IS KEY PRESSED?
CBAD: CO	93		11		CPY	#\$93 NOWAIT	IS IT CTL-S?
CBB1:20 CBB4:A0	: 10	CO	13	KBDWAIT	BIT	KBDSTRB	CLEAR STROBE
C887: 10 C889: C0) FB	C884	15		BPL	KBDWAIT #983	; IF CTL-C, LEAVE IT
C888: F0 C880: 20	03	C8C0	17		BEQ	#\$83 NOWAIT KBDSTRB	IN THE KED BUFFER
CBCO:	0.000	C8C0	19	NOWAIT	EQU	*	
C8C0: 25 C8C2: C5	9 20		20 21		AND CMP	#\$7F #\$20	; DROP POSSIBLE HI BIT ; IS IT CONTROL CHAR?
C8C4: B0	06	CBCC	22		BCS	BPNCTL	;=>NOPE

C8C6:20 99 CB C8C9:4C E2 C8	23 24		CTLCHAR BIORET	; EXECUTE POSSIBLE CTL CHAR ; =>EXECUTED OR IGNORED
CBCC:	25 *			
CBCC:	26 * NOT A C	TL CHAR. F	PRINT IT.	
CBCC:	27 *		_	
CBCC: CBCC	28 BPNCTL 29	EQU 1	NURCH	AFT CH
CBCC: AC 78 05 CBCF: AD 78 06	30		CHAR	GET CH GET CHAR (ALL 8 BITS)
C802: 20 F2 CE	31		STORCHAR	STUFF ONTO SCREEN
CBD5:	32 *			
CBD5:		E CURSOR H	HOR I ZONTAL:	
C8D5: C8D5: EE 78 05	34 * 35	INC (DURCH	BUMP IT
CEDS: AD 78 05	36		URCH	ARE WE PAST THE
CBDB: C5 21	37	CMP L	WNDWDTH	ARE WE PAST THE
CBDD: 90 03 CBE			BIORET	; =>NO, NO PROBLEM
CODF: 20 EC CB COE2:	39 40 *	JSR)	X. CR	; YES, DO C/R
CBE2: CBE2		EQU 1	*	
C8E2: AD 78 05	42		DURCH	SET CH AND CV
C8E5: 20 AF CE	43	JSR S	BETCH	FOR BASIC
CBEB: AD FB 05	44		BURCV	
C8EB: 85 25	45		cv	RESTORE
CBED: 68 CBEE: AA	46 47	PLA TAX		RESTORE
CBEF: 68	48	PLA		X AND Y
CBFO: AB	49	TAY		
C8F1: 68	50	PLA		; AND AC
C8F2: AD 7B 06 C8F5: 60	51 52	LDA (CHAR	RETURN TO BASIC
CBF5:60	32		E BINPUT	TRETORN TO BRAIL
CBF6:	2 * BASIC I	NPUT:		
CBF6:	3*			
CBF6: CBF6	4 BINPUT	EQU	*	
C8F6: AD FB 04	5	LDA I	MODE #M. BINPUT	; SAY THAT ; WE'RE INPUTTING
C8F9:09 40 C8FB:80 FB 04	6 7		MODE	WE RE INFOLLING
CBFE: AD 7B 06	8		CHAR	GET CHAR AT CURSOR AND
C701: A4 24	9	LDY	СН	GET CURSOR POSITION
C703: 91 28	10		(BASL), Y	; REPAIR MONITOR'S SILLY ATTEMPT
C905: C901		EQU	*	
C905:20 DD CE C908:20 15 CB	12 13		INVERT GETKEY	;CREATE DUR DWN CURSOR IMAGE ;GET A KEY
C908:20 15 CB	13	STA	CHAR	SAVE IT
C90E: 20 DD CE	15		INVERT	REMOVE CURSOR
C911:C9 98	16	CMP	#\$9B	
C913: FO 03 C91	3 17	BEG 1	ESCAPING	;=>YES IT IS
C713.F0 03 C71		N-4		
C915: 4C B7 C9	18	JMP I	NDESC	;=>ND, IT'S NORMAL
C915:4C B7 C9 C918	18 20 * START A	JMP I	NDESC SEQUENCE:	;=>ND, IT'S NORMAL
C915:4C B7 C9 C918 C918:	18 20 * START A 21 * WE HAN 22 * 0 -	JMP I N ESCAPE : DLE THE FI HDMF & CLI	NDESC SEQUENCE: OLLOWING ON EAR	;=>ND, IT'S NORMAL
C915:4C B7 C9 C918 C918: C918: C918: C918:	18 20 * START A 21 * WE HAN 22 * 0 -	JMP I N ESCAPE : DLE THE FI HDMF & CLI	NDESC SEQUENCE: OLLOWING ON EAR	;=>ND, IT'S NORMAL
C915:4C B7 C9 C918: C918: C918: C918: C918: C918:	18 20 * START A 21 * WE HAN 22 * & - 23 * E - 24 * F -	JMP I N ESCAPE : DLE THE F HOME & CLI CLR TO ED CLR TO ED	NDESC SEQUENCE: OLLOWING ON EAR	;=>ND, IT'S NORMAL
C915: 4C B7 C9 C918: C918: C918: C918: C918: C918: C918: C918:	18 20 * START A 21 * WE HAN 22 * & - 23 * E - 24 * F - 25 * I -	JMP I N ESCAPE : DLE THE F HOME & CLI CLR TO ED CLR TO ED CURSOR UP	NDESC SEQUENCE: OLLOWING ON EAR L S	;=>ND, IT'S NORMAL
C915:4C B7 C9 C918 C918: C918: C918: C918: C918: C918: C918: C918:	18 20 * START A 21 * WE HAN 22 * & - 23 * E - 24 * F - 25 * I - 26 * J -	JMP 1 N ESCAPE 3 DLE THE F1 HOME & CL1 CLR TO E0 CLR TO E0 CURSOR UP CURSOR LE1	NDESC SEQUENCE: OLLOWING ON EAR L S FT	;=>ND, IT'S NORMAL
C915: 4C B7 C9 C918: C918: C918: C918: C918: C918: C918: C918:	18 20 * START A 21 * WE HAN 22 * & - 23 * E - 24 * F - 25 * I - 26 * J - 27 * K - 28 * M -	JMP I N ESCAPE : DLE THE F HOME & CLI CLR TO EO CLR TO EO CURSOR UP CURSOR LEI CURSOR RII CURSOR RII	NDESC SEQUENCE: OLLOWING ON EAR L S FT GHT WN	;=>ND, IT'S NORMAL ES:
C915: 4C B7 C9 C918: C918: C918: C918: C918: C918: C918: C918: C918: C918: C918: C918: C918:	18 20 * START A 21 * WE HAN 22 * & - 23 * E - 24 * F - 25 * I - 26 * J - 27 * K - 28 * M -	JMP I N ESCAPE : DLE THE F HOME & CLI CLR TO EO CLR TO EO CURSOR UP CURSOR LEI CURSOR RII CURSOR RII	NDESC SEQUENCE: OLLOWING ON EAR L S FT GHT WN	;=>ND, IT'S NORMAL ES:
C913: 4C B7 C9 C918: C918: C918: C918: C918: C918: C918: C918: C918: C918: C918: C918: C918: C918: C918:	18 20 * START A 21 * WE HAN 22 * & - 23 * E - 24 * F - 25 * I - 26 * J - 27 * K - 28 * M - 29 * R - 30 * T -	JMP I N ESCAPE: DLE THE FI HOME & CLI CLR TO EOU CLR TO EOU CURSOR UP CURSOR LEI CURSOR DOR RESTRICT TURN DFF I	NDESC SEQUENCE: QLLOWING ONI EAR L S FT GHT WN TO UPPERCASI ESC-R	;=>ND, IT'S NORMAL ES:
C918: C918: C918: C918: C918: C918: C918: C918: C918: C918: C918: C918: C918: C918: C918: C918: C918:	18 20 * START A 21 * WE HAN 22 * & - 23 * E - 24 * F - 25 * I - 26 * J - 27 * K - 28 * M - 29 * R - 30 * T - 31 * 4 -	JMP I N ESCAPE : DLE THE FI HOME & CLI CLR TO EOI CLR TO EOI CURSOR LEI CURSOR RI CURSOR RI CURSOR DOI RESTRICT : TURN DFF I GOTO 40 CI	NDESC SEQUENCE: OLLOWING ON EAR L S FT GHT WN TO UPPERCAS ESC-R OLUMN MODE	;=>ND, IT'S NORMAL ES:
C913: 4C B7 C9 C918: C918: C918: C918: C918: C918: C918: C918: C918: C918: C918: C918: C918: C918: C918:	18 20 * START A 21 * WE HAN 22 * C - 23 * E - 24 * F - 25 * I - 26 * J - 27 * K - 28 * M - 27 * R - 30 * T - 31 * 4 - 31 * 4 -	JMP 1 N ESCAPE : DLE THE FI HOME & CLI CLR TO EDI CLR TO EDI CURSOR UP CURSOR LEI CURSOR DE CURSOR DE RESTRICT TURN DFF 1 GOTO 40 CI	NDESC SEQUENCE: OLLOWING ON EAR L S FT GHT WN TO UPPERCASI ESC-R OLUMN MODE	;=>ND, IT'S NORMAL ES:
C 913: 4C 87 C 9 C 918: C 918:	18 20 * START A 21 * WE HAN 22 * d - 23 * E - 24 * F - 25 * I - 25 * I - 26 * J - 27 * K - 28 * M - 29 * R - 30 * T - 31 * 4 - 32 * 8 - 33 * CTL-90- 34 * THE F	JMP 1 N ESCAPE : DLE THE FI HOME & CLI CLR TO ECI CURSOR UP CURSOR RI CURSOR RI CURSOR RI CURSOR RI CURSOR DO RESTRICT JURN DFF 1 GOTO 40 CI GOUT (PR#	NDESC SEQUENCE: OLLOWING ON EAR L S FT GHT WN TO UPPERCASI ESC-R OLUMN MODE	;=>ND, IT'S NORMAL ES: E
C918: C918:	18 20 * START A 21 * WE HAN 22 * & - 23 * E - 25 * I - 25 * I - 26 * J - 27 * K - 28 * M - 29 * R - 30 * T - 31 * 4 - 32 * CT-Q- 34 * THE F 35 *	JMP 1 N ESCAPE : DLE THE FI HOME & CLI CLR TO ECI CURSOR LE CURSOR NE CURSOR RI CURSOR RI CURSOR RI TURN DFF 1 GOTO 40 CI GUTO 40 CI GUTI 40 CI GUTI 40 CI	NDESC SEQUENCE: QLLOWING ONI EAR L S FT GHT WN PPPERCASI ESC-R OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE (JINHO) KEYS (AS I	;=>ND, IT'S NORMAL ES: E
C 913: 4C 87 C 9 C 918: C 91	18 20 * START A 21 * WE HAN 22 * C - 23 * E - 24 * F - 25 * I - 25 * I - 26 * J - 27 * K - 28 * M - 30 * T - 31 * 4 - 32 * 8 - 33 * CTL-Q- 34 * THE F 35 *	JMP 1 N ESCAPE : DLE THE FI HOME & CLLR TO ECI CLR TO ECI CURSOR LEI CURSOR LEI CURSOR DO RESTRICT TURN DFF 1 GOTO 40 CI GOTO 80 CL GUIT (PR# GUIT (PR# MSB 0	NDESC SEQUENCE: QLLOWING DNI EAR FT GHT WN TO UPPERCASI ESC-R OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE GEF	;=>ND, IT'S NORMAL ES: E
C 918: C	18 20 * START A 21 * WE HANN 22 * @ - 23 * E - 24 * F - 25 * F - 26 * J - 27 * K - 28 * M - 29 * R - 30 * T - 31 * 4 - 32 * CT-Q- 34 * THE F 35 * 36 37 ESCAPING	JMP I N ESCAPE DLE THE FI HOME & CLIC CLR TO EOI CLR TO EOI CURSOR LEI CURSOR LEI CURSOR NI CURSOR NI CURSOR NI CURSOR NI CURSOR OI GUTO 40 CI GUTO 40 CI GUTI (PR#) OUR ARROW MSB EQU	NDESC SEQUENCE: QLLOWING ONI EAR L S S GHT WN UPPERCASI ESC-R DLUMN MODE GLUMN MODE Q/INHO) KEYS (AS I GFF *	;=>ND, IT'S NORMAL ES: E
C 913: 4C 87 C 9 C 918: C 918:	18 20 * START A 21 * WE HAN 22 * @ - 23 * E -24 * F - -25 * I - 26 * J - 27 * K - 28 * M - 30 * T - 31 * 4 - 32 * CT-0- 34 * CT-0- 35 * CT-0- 36 37 ESCAPING 38	JMP 1 N ESCAPE : DLE THE FI HOME & CLL CLR TO ECI CLR TO ECI CURSOR LEI CURSOR NUP CURSOR DO RESTRICT GUTO 40 CI GOTO 40 CI GOTO 90 CI QUIT (PR# OUR ARROW MSB 1 EQU JSR 1	NDESC SEQUENCE: QLLOWING DNI EAR L S FT GHT WN TO UPPERCASI ESC-R QLUMN MODE QLUMN MODE QLUMN MODE QLUMN MODE QLUMN MODE SESCON	;=>ND, IT'S NORMAL ES: E JKM) ;ESCAPE CURSORON
C 913: 4C 87 C 9 C 918: C	18 20 * START A 21 * WE HAN 22 * te - 23 * E - 24 * F - 25 * I - 26 * J - 26 * J - 27 * K - 28 * M - 30 * T - 31 * 4 - 32 * CI - 34 * THE F 35 * 36 37 ESCAPING 38 39 40	JMP I N ESCAPE: DLE THE F: HOME & CLI CLR TO EDI CLR TO EDI CLR TO EDI CLRSOR LOE CURSOR LOE CURSOR NO CURSOR RI CURSOR NO RESTRICT TURN DFF I RESTRICT TURN DFF I GUIT (PRH OUR ARROW MSB I EQUIT (PRH JSR I JSR I	NDESC SEQUENCE: QLLOWING DNI EAR L S FT OHT WN TO UPPERCASI ESC-R OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE GLUMN MODE SECOFF * ESCON GETKEY ESCOF	;=>ND, IT'S NORMAL ES: E JKM) ;ESCAPE CURSORON ;GET ESCAPE FUNCTION ; REPLACE DRIGINAL CHARACTER
C 913: 4C 87 C 9 C 918: C 915 C 9 C 9 C 9 C 9 C 9 C 9 C 9 C 9 C 9 C 9	18 20 * START A 21 * WE HAN 22 * & - 23 * E - 24 * F - 25 * J - 25 * J - 27 * K - 28 * M - 27 * K - 28 * M - 30 * T - 31 * 4 - 32 * CT-Q- 34 * THE F 35 * 36 37 ESCAPING 38 39 40	JMP I N ESCAPE DLE THE FI HOME & CLI CLR TO EDI CLR TO EDI CURSOR UC CURSOR LEI CURSOR DOI RESTRICT TURN DFF I GUTU 40 CI GUTU 50 CI STR I JSR I JSR I AND	NDESC SEQUENCE: OLLOWING DNI EAR S TT OHT OHT OUPPERCASI ESC-R OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE SCON GETS ESCON GETKEY ESCOFF #\$77	;=>ND, IT'S NORMAL ES: JKM) ;ESCAPE CURSORON ;GET ESCAPE FUNCTION ;REPLACE DRIGINAL CHARACTER ;DROP H BIT
C 913: 4C 87 C 9 C 918: C 91	18 20 * START A 21 * WE HAN 22 * & H 23 * E 23 * E 24 * F 25 * I 25 * I 26 * J 27 * K 28 * M 29 * R 30 * T 31 * 4 32 * 8 - 33 * CTHE F 35 * 36 37 ESCAPING 38 39 40 41 42	JMP I JNE THE DLE THE DLE THE IND SCL CLR TO CLR TO CLR TO CURSOR CURSOR CURSOR CURSOR RESTRICT TURN GOTO 40 CI GOTO 40 CI GOTO 40 CI GOTO 40 CI JSR JSR JSR JSR JSR JSR JSR JSR CMP SSR	NDESC SEQUENCE: OLLOWING ONI EAR FT GHT WN TO UPPERCASI ESC-R OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE SECON KEYS (AS I OFF * SECON GETKEY ESCOFF #\$50	;=>ND, IT'S NORMAL ES: JKM) ;ESCAPE CURSORON ;GET ESCAPE FUNCTION ;REPLACE DRIGINAL CHARACTER ;DROP H BIT
C 913: 4C 87 C 9 C 918: C 91	18 20 * START A 21 * WE HANN 22 * @ - 23 * E - 24 * F - 25 * I - 25 * I - 26 * K - 28 * M - 29 * R - 30 * T - 31 * 4 - 32 * CT-Q- 34 * THE F 35 * 36 37 ESCAPING 38 39 40 41 42 42 43	JMP I JULE THE F F DLE THE F F DALE THE F F DALE THE F F CLR TO ECO CLR TO ECO CURSOR COLCARSOR VE CURSOR FLIC CURSOR FIL CURSOR FOO RESTRICT TURN DFF I GOTO 40 CI GOTO 40 CI GOTO 40 CI GOTO 40 CI JSR I JSR I JSR I JSR I JSR I SC CHP BCC	NDESC SEQUENCE: OLLOWING DNI EAR S S FT OHT WN UPPERCASI ESC-R OLUMN MDDE OLUMN MDDE OLUMN MDDE OLUMN MDDE OLUMN MDDE SCON GFF ESCON GETKEY ESCOFF #\$50	;=>ND, IT'S NORMAL ES: UKM) ;ESCAPE CURSORON ;GET ESCAPE FUNCTION ;REPLACE ORIGINAL CHARACTER ;DROP HI BIT ;IS IT LOWERCASE? ;=>NO, DM'T UPSHIFT
C 913: 4C 87 C 9 C 918: C 91	18 20 * START A 21 * WE HAN 22 * E - 23 * E - 24 * F - 25 * I - 26 * J - 27 * K - 28 * M - 30 * T - 31 * 4 - 32 * C - 34 * THE F 36 37 ESCAPING 38 37 40 41 42 43	JMP I JULE THE F F DLE THE F F DALE THE F F DALE THE F F CLR TO ECO CLR TO ECO CURSOR COLCARSOR VE CURSOR FLIC CURSOR FIL CURSOR FOO RESTRICT TURN DFF I GOTO 40 CI GOTO 40 CI GOTO 40 CI GOTO 40 CI JSR I JSR I JSR I JSR I JSR I SC CHP BCC	NDESC SEQUENCE: OLLOWING ONI EAR FT GHT WN TO UPPERCASI ESC-R OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE SECON KEYS (AS I OFF * SECON GETKEY ESCOFF #\$50	;=>ND, IT'S NORMAL ES: JKM) ;ESCAPE CURSORON ;GET ESCAPE FUNCTION ;REPLACE DRIGINAL CHARACTER ;DROP HI BIT
C 913: 4C 87 C 9 C 918: C 91	18 20 * START A 21 * WE HAN 22 * E - 23 * E - 24 * F - 25 * I - 26 * J - 27 * K - 28 * M - 30 * T - 31 * 4 - 32 * CL-Q- 34 * THE F 35 * 36 37 ESCAPING 38 39 40 41 42 43 44 44 44 44	JMP I JNLE THE F FN DLE THE F FN DRE SCALE CLR TO EDICURSOR UP CURSOR CURSOR CLE CURSOR CLE CURSOR DO RESTRICT RESTRICT TURN OFF I GOTO 40 CURSOR DO ROFT GOTO 40 CURSOR DO ROTO 40 CURSOR DO MESTRICT SOTO 40 CURSOR DO GUT 470 CURSOR DO SOTO 40 CURSOR DO GUT 40 CURSOR DO SOTO 40 CURSOR DO GUT 470 DO SOTO	NDESC SEQUENCE: OLLOWING DNI EAR S S FT OHT WN UPPERCASI ESC-R OLUMN MDDE OLUMN MDDE OLUMN MDDE OLUMN MDDE OLUMN MDDE SCON GFF ESCON GETKEY ESCOFF #\$50	;=>ND, IT'S NORMAL ES: JKM) ;ESCAPE CURSORON ;GET ESCAPE FUNCTION ;REPLACE ORIGINAL CHARACTER ;DROP HI BIT ;IS IT LOWERCASE? ;=>ND, DM'T UPSHIFT
C 913: 4C 87 C 9 C 918: C 91	18 20 * START A 21 * WE HAN 22 * E - 23 * E - 24 * F - 25 * I 25 * I 25 * I 25 * I 26 * J - 27 * K - 28 * M - 29 * R - 30 * T - 31 * 8 - 33 * CTL-Q- 34 * THE F 35 * 36 37 ESCAPING 38 39 40 41 42 43 44 45 ESC1 46	JMP I JME THE DLE THE DLE THE N ESCAPEE DLE THE INDE Sclint CLR TO CLR TO CURSOR CURSOR CURSOR CURSOR CURSOR CONSOR CURSOR CONSOR CURSOR GOTO CURSOR SOTO CURSOR SOTO CURSOR SOTO GOTO SOTO GOTO SOTO GOTO SOTO JUN ARROW MSB I EQUIT SOR JSR JSR JSR JSR JSR JSR AND COMP BCC AND COMP EQU LDY EQU	NDESC SEQUENCE: GLLOWING DNI EAR S S FT GHT TO UPPERCASI ESC-R DLUMN MODE GLUMN MODE SSC S S ESCON ESCON ESCON ESCON ESCON S S S S S S S S S S S S S S S S S S S	;=>ND, IT'S NORMAL ES: JKM) ;ESCAPE CURSORON ;GET ESCAPE FUNCTION ;REPLACE ORIGINAL CHARACTER ;DROP HI BIT ;IS IT LOWERCASE? ;=>NO, DN'T UPSHIFT ;UPSHIFT ;COUNT/INDEX
C 913: 4C 87 C 9 C 918: C 91	18 20 * START A 21 * WE HAN 22 * E - 23 * E - 24 * F - 25 * I - 26 * J - 27 * K - 28 * M - 29 * R - 30 * T - 32 * CL-0- 34 * THE F 35 * 36 37 ESCAPING 38 39 40 41 42 43 44 45 36 45 37 ESCAPING	JMP I JNLE THE F F DLE THE F F MOME & CL CLR TO EDIC CLRSOR UP CURSOR CUC CURSOR CUC CURSOR DO RESTRICT TURN DFF I GOTO 40 C GOTO 40 C GOTO 40 C GOTO 40 C GOTO 40 C JSR JSR JSR JSR JSR JSR BCC C AND CMP BCC AND C EQU CMP	NDESC SEQUENCE: OLLOWING DNI EAR S FT GHT WN TO UPPERCASI ESC-R OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE SCON KEYS (AS I COFF #877 #8500 ESC1 #350 ESC1 #5500 #8000 #80	;=>ND, IT'S NORMAL ES: JKM) ;ESCAPE CURSORON ;GET ESCAPE FUNCTION ;REPLACE ORIGINAL CHARACTER ;DROP HI BIT ;IS IT LOWERCASE? ;=>ND, DON'T UPSHIFT ; VOSHIFT ;COUNT/INDEX ;IS IT A VALID ESCAPE?
C 913: 4C B7 C9 C 918: C 918	18 20 * START A 21 * WE HAN 22 * C - 23 * E - 24 * F - 25 * J - 26 * J - 27 * K - 28 * M - 29 * R - 30 * T - 31 * C - 31 * C - 33 * CTL-Q- 34 * T THE F 35 * 36 37 ESCAPING 38 39 40 41 42 7 43 ESC1 46 47 ESC2 48	JMP I JNLE THE FI FILE DLE THE FI FILE NOTE SCLL CLR TO ECO CLR TO ECO CURSOR CO CURSOR CO CURSOR CO CURSOR CO CURSOR CO CURSOR CO CURSOR CO CURSOR CO CO JUN ARROW MSB LOUIT (FRANCON JSR JSR JSR JSR JSR JSR JSR JSR SC AND CO CHP EQU LDY EQU CMP EQU	NDESC SEQUENCE: GLLOWING DNI EAR S S FT GHT TO UPPERCASI ESC-R DLUMN MODE GLUMN MODE SSC S S ESCON ESCON ESCON ESCON ESCON S S S S S S S S S S S S S S S S S S S	;=>ND, IT'S NORMAL ES: JKM) ;ESCAPE CURSORON ;GET ESCAPE FUNCTION ;REPLACE ORIGINAL CHARACTER ;DROP HI BIT ;IS IT LOWERCASE? ;=>NO, DN'T UPSHIFT ;UPSHIFT ;COUNT/INDEX
C 913: 4C 87 C 9 C 918: C 91	18 20 * START A 21 * WE HAN 22 * E 23 * E 24 * F 25 * I 26 * J 27 * K 28 * M - 27 * K 28 * M - 30 * T 31 * 4 32 * CL 33 * THE F 36 37 ESCAPING 38 39 40 41 42 43 44 54 46 37 ESC1 46 47 ESC2 48 50	JMP 1 JNLE THE F FN DLE THE F FN NE SCAPE DL CLR TO ED CLRSOR CL CURSOR CL CURSOR CL CURSOR DO RESTRICT TURN DFF I GOTO 40 CL GOTO 40 CL GOTO 40 CL GOTO 40 CL GOTO 40 CL GUT 47 CL SR JSR JSR JSR JSR JSR GUT BCC AND CMP BCC AND CMP EQU CMP BEG DEY	NDESC SEQUENCE: OLLOWING DNI EAR S FT GHT WN TO UPPERCASI ESC-R OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE SCON KEYS (AS I COFF #877 #8500 ESC1 #350 ESC1 #5500 #8000 #80	;=>ND, IT'S NORMAL ES: JKM) ;ESCAPE CURSORON ;GET ESCAPE FUNCTION ;REPLACE ORIGINAL CHARACTER ;DROP HI BIT ;IS IT LOWERCASE? ;=>ND, DON'T UPSHIFT ; VOSHIFT ;COUNT/INDEX ;IS IT A VALID ESCAPE?
C 913: 4C 87 C 9 C 918: C 91	18 20 * START A 21 * WE HAN 22 * C - 23 * E - 24 * F - 25 * I - 26 * J - 26 * J - 27 * K - 28 * M - 30 * T - 32 * C - 33 * THE F 33 * THE F 34 * THE F 35 * 36 37 ESCAPING 38 37 ESCAPING 38 37 40 40 41 42 43 44 45 45 45 45 55 52 55 55 55 55 55 55 55 5	JMP I JULE THE FI FI DLE THE FI FI DLE THE FI FI CLR TO ECO CLR TO ECO CURSOR CUC CURSOR CUC CURSOR CO CURSOR FI CURSOR FI CURSOR FI CURSOR CO CURSOR CO CURSOR OF GOTO 40 CI GOTO 40 CI GOTO 40 CI GOTO 40 CI JUL ARROW MSB I EQUIT CRM JSR I JSR I	NDESC SEQUENCE: OLLOWING ONI EAR S FT GHT WN TO UPPERCASI ESC-R OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE SCON ESCI #8400 ESCI #8500 * #5500W * #5500W * ESC33 *	;=>ND, IT'S NORMAL ES: JKM) ;ESCAPE CURSORDN ;GET ESCAPE FUNCTION ;REPLACE ORIGINAL CHARACTER ;DROP HI BIT ;IS IT LOWERCASE? ;=>ND, DON'T UPSHIFT ;USSHIFT ;COUNT/INDEX ;IS IT A VALID ESCAPE? ;=>YES
C 913: 4C B7 C9 C 918: C 918	18 20 * START A 21 * WE HAN 22 * E - 23 * E - 24 * F - 25 * I - 25 * I - 26 * J - 27 * K - 28 * M - 30 * T - 31 * A - 32 * C - 33 * C TL- 34 * THE F 36 37 ESCAPING 38 39 40 41 42 43 44 45 55C1 46 47 45 46 47 48 47 51 52 53 *	JMP I JN F SCAPE DLE THE FI N ESCAPE DLE THE FI MOME & CLI CLR TO EOI CLR TO EOI CURSOR CLE CURSOR DO RESTRICT RESTRICT TURN DFF I GOTO 40 CURSOR DO ROTA 40 CURSOR MO MSB JSR MOUTI (PRH OUTI (PRH JSR MOUTI (PRH JSR MOUTI (PRH JSR MOUTI (PRH AND JSR MOUTI (PRH BCC AND EQU CMP MOUTI (PRH BCC AND BCQ DEFY BFE GU DEFY BFN I BHI I	NDESC SEQUENCE: GLLOWING DNI EAR S FT GHT TO UPPERCASI ESC-R DLUMN MODE GLUMN MODE SSC- SSC- SSC- SSC- SSC- SSC- SSC- SSC	;=>ND, IT'S NORMAL ES:
C913: 4C B7 C9 C918: C918: C918: C918: C916: C92: C918: C92: C92: C92: C92: C92: C92: C92: C92: C92: C92: C92: C92: C92: C92: C92: F0 05 C93: C93: C93: 01 C94: C933: 10 C94:	18 20 * START A 21 * WE HAN 22 * C - 23 * E - 24 * F - 25 * I - 26 * J - 26 * J - 27 * K - 28 * M - 30 * T - 31 * A - 32 * 8 - 33 * THE F 33 * THE F 34 * THE F 35 * 36 37 ESCAPING 38 37 ESCAPING 38 37 ESCAPING 38 37 ESCAPING 38 37 ESCAPING 38 37 40 40 41 42 43 44 45 45 45 55 55 55 55 55 55	JMP I JNE THE DLE THE N ESCAPE DL CURSOR CURSOR ED CURSOR ED CURSOR ED CURSOR ED CURSOR ED CURSOR ED GOTO 40 GOTO 40 GOTO 40 GOTO 40 GOTO 40 JSR JSR JSR JSR JSR JSR JSR AND AND AD EQU JSR EQU EQU BEG JSR BEG JE BE JE BH I BH EQU	NDESC SEQUENCE: OLLOWING ONI EAR S GHT WN TO UPPERCASI ESC-R OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE SCON ESC * #SCF #SCON ESC1 #SSCNUM * ESC1AB.Y ESC3 ESC2 ESC2 *	;=>ND, IT'S NORMAL ES: UKM) ;ESCAPE CURSORDN ;ESCAPE FUNCTION ;REPLACE ORIGINAL CHARACTER ;DROP HI BIT ;IS IT LOWERCASE? ;=>ND, DON'T UPSHIFT ;UPSHIFT ;COUNT/INDEX ;IS IT A VALID ESCAPE? ;=>YES ;TRY 'EM ALL ;=>MAYDE IT'S A SPECIAL ONE
C913: 4C B7 C9 C918: C921: C922: C923: C924: C925: C926: C927: C928: C928:<	18 20 * START A 21 * WE HAN 22 * E - 23 * E - 24 * F - 25 * I - 25 * I - 26 * J - 27 * K - 28 * M - 30 * T - 31 * A - 32 * C - 33 * C TL- 34 * THE F 36 37 ESCAPING 38 39 40 41 42 43 44 45 55C1 46 47 45 46 47 48 47 51 52 53 *	JMP I JN HE FIHE DLE THE FI N ESCAPE CLIC CLR TO CLIC CLIC CLR TO EGIC CLIC CURSOR CURSOR CURSOR CURSOR RESTRICT TURN DIF GOTO 40 CI GOTO 40 CI GOTO 40 CI GOTO 40 CI UJUT FRH JSR	NDESC SEQUENCE: GLLOWING DNI EAR S FT GHT TO UPPERCASI ESC-R DLUMN MODE GLUMN MODE SSC- SSC- SSC- SSC- SSC- SSC- SSC- SSC	; =>ND, IT'S NORMAL ES:
C 913: 4C 87 C 9 C 918: C 92: C 93: C 95: C 95:	18 20 * START A 21 * WE HAN 22 * E - 23 * E - 24 * F - 25 * I - 25 * I - 25 * I - 25 * I - 25 * M - 30 * T - 31 * A - 32 * B 34 * THE F 35 * CCP ING 36 37 ESCAP ING 38 39 40 41 42 7 43 36 37 36 37 36 37 36 37 40 41 42 43 44 55 50 37 51 52 53 53 53 54 55 54 55 54	JMP I JNE THE DLE THE N ESCAPE DLE THE NDLE CLR CLR TO EGUC CLR CURSOR CLR CURSOR CLR EGUTO CUC CURSOR CLR RESTRICT TURN OGITO 40 CL GOTO 40 CL GOTO 40 CL JSR JSR BEQU JSR BEQU BEQU BHI BEQU BHI EQU	NDESC SEQUENCE: OLLOWING ONI EAR S FT CHT WN TO UPPERCASI ESC-R OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE SCON ESC SCON SECTAB.Y ESCON ESC SECTAB.Y ESCS ESCS ESCS ESCS ESCS ESCS ESCS ESC	<pre>;=>ND, IT'S NORMAL ES: UKM) ;ESCAPE CURSORON ;GET ESCAPE FUNCTION ;REPLACE DRIGINAL CHARACTER ;DROP HI BIT ;IS IT LOWERCASE? ;=>NO, DON'T UPSHIFT ;UPSHIFT ;COUNT/INDEX ;IS IT A VALID ESCAPE? ;=>YES ;TRY 'EM ALL ;=>NAYDE IT'S A SPECIAL ONE ;OET CHAR TO "PRINT" ;DROP HI BIT (FLAG)</pre>
C913: 4C 87 C9 C918: C918: C918: C918: C918: C912: C918: C92: C92: C92: C92: C92: C93: C93: C93: C93: C93:	18 20 * START A 21 * WE HAN 22 * C - 23 * E - 24 * F - 25 * J - 25 * J - 25 * J * 27 * K - 28 * M - 32 * 8 - 33 * T - 33 * S - 34 * THE F 35 * 36 37 ESCAPING 38 39 40 41 42 7 43 34 45 ESC1 46 46 5 52 5 5 4 ESC3 5 5 5 4 ESC3 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	JMP I JMP I N ESCAPE DLE THE F DLE THE F INDECTION N ESCAPE DLE THE F CUR TO EDU CURSOR LEICURSOR LEICURSOR LEICURSOR DO CURSOR DO RESTRICT TURN DFF I GOTO 40 C GOTO 40 C GOTO 40 C GOTO 40 C GOTO 40 C JSR I JSR I JSR I JSR I JSR I JSR I CMP EQU CMP BCC BCQ I PHE I EQU JSR I JSR I BAND EQU LDA I AND JSR I EQU LDA I	NDESC SEQUENCE: OLLOWING DNI EAR S FT OHT WN TO UPPERCASI ESC-R OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE SCI ESCON GETKEY ESCON GETKEY ESCON ESCON ESCTAB.Y ESCTAB.Y ESCCHAR,Y #37 CTLCHAR ESCCHAR,Y	;=>ND, IT'S NORMAL ES: E JKM) ;ESCAPE CURSORON ;GET ESCAPE FUNCTION ;GET ESCAPE FUNCTION ;REPLACE ORIGINAL CHARACTER ;DROP HI BIT ;IS IT LOWERCASE? ;=>NA.DON'T UPSHIFT ;USHIFT ;COUNT/INDEX ;IS IT A VALID ESCAPE? ;=>YES ;TRY 'EM ALL ;=>MAYDE IT'S A SPECIAL ONE ;GET CHAR TO "PRINT" ;DROP HI BIT (FLAG) ;EXECUTE IT ;GET FLAG
C913: 4C B7 C9 C918: C929: C920: C920: C921: C922: C923: C933: C933:<	18 20 * START A 21 * WE HAN 22 * E - 23 * E - 24 * F - 25 * I 25 * I 26 * J - 27 * K 28 * M - 30 * T - 31 * A 32 * C - 34 * THE F 36 37 ESCAPING 38 37 40 41 42 43 44 5 5 5 5 5 6 7 8 7 28 40 41 42 43 5 5 5 6 6 7 8 7 8 7 8 7 8 7 8	JMP I JN F SCAPE DLE THE F F NE SCAPE CLC CLR TO ED CLC CURSOR UC CORSOR DO RESTRICT TURN DFF GOTO 40 C GOTO 40 C GOTO 40 C GOTO 40 C GOTO 40 C SOTO 40 C GOTO 40 C SOTO 40 C GOTO 40 C SOTO 40 C GOTO 40 C SOR	NDESC SEQUENCE: OLLOWING DNI EAR FT GHT WN TO UPPERCASI ESC-R OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE SCON ESCON	;=>ND, IT'S NORMAL ES:
C913: 4C B7 C9 C918: C921: C923: C923: C924: C925: C926: C927: C928: C928: C928: C930: C930: C930: C933: C933:<	18 20 * START A 21 * WE HAN 22 * E - 23 * E - 24 * F - 25 * I - 25 * I - 25 * I - 25 * I - 26 * J - 27 * K - 28 * M - 30 * T - 32 * C - 33 * C - 33 * T - 32 * C - 34 * THE F 35 * 36 37 ESCAPING 38 39 40 41 42 7 43 44 45 55 56 57 50 51 52 53 54 55 54 55 56 57 58 37 52 53 54 55 56 57 <t< td=""><td>JMP I JN F SCAPE DLE THE F F NE SCAPE CLC CLR TO ED CLC CURSOR UC CORSOR DO RESTRICT TURN DFF GOTO 40 C GOTO 40 C GOTO 40 C GOTO 40 C GOTO 40 C SOTO 40 C GOTO 40 C SOTO 40 C GOTO 40 C SOTO 40 C GOTO 40 C SOR GOTO 40 C SOR</td><td>NDESC SEQUENCE: OLLOWING DNI EAR S FT OHT WN TO UPPERCASI ESC-R OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE SCI ESCON GETKEY ESCON GETKEY ESCON ESCON ESCTAB.Y ESCTAB.Y ESCCHAR,Y #37 CTLCHAR ESCCHAR,Y</td><td>;=>ND, IT'S NORMAL ES: E JKM) ;ESCAPE CURSORON ;GET ESCAPE FUNCTION ;GET ESCAPE FUNCTION ;REPLACE ORIGINAL CHARACTER ;DROP HI BIT ;IS IT LOWERCASE? ;=>NA.DON'T UPSHIFT ;USHIFT ;COUNT/INDEX ;IS IT A VALID ESCAPE? ;=>YES ;TRY 'EM ALL ;=>MAYDE IT'S A SPECIAL ONE ;GET CHAR TO "PRINT" ;DROP HI BIT (FLAG) ;EXECUTE IT ;GET FLAG</td></t<>	JMP I JN F SCAPE DLE THE F F NE SCAPE CLC CLR TO ED CLC CURSOR UC CORSOR DO RESTRICT TURN DFF GOTO 40 C GOTO 40 C GOTO 40 C GOTO 40 C GOTO 40 C SOTO 40 C GOTO 40 C SOTO 40 C GOTO 40 C SOTO 40 C GOTO 40 C SOR	NDESC SEQUENCE: OLLOWING DNI EAR S FT OHT WN TO UPPERCASI ESC-R OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE SCI ESCON GETKEY ESCON GETKEY ESCON ESCON ESCTAB.Y ESCTAB.Y ESCCHAR,Y #37 CTLCHAR ESCCHAR,Y	;=>ND, IT'S NORMAL ES: E JKM) ;ESCAPE CURSORON ;GET ESCAPE FUNCTION ;GET ESCAPE FUNCTION ;REPLACE ORIGINAL CHARACTER ;DROP HI BIT ;IS IT LOWERCASE? ;=>NA.DON'T UPSHIFT ;USHIFT ;COUNT/INDEX ;IS IT A VALID ESCAPE? ;=>YES ;TRY 'EM ALL ;=>MAYDE IT'S A SPECIAL ONE ;GET CHAR TO "PRINT" ;DROP HI BIT (FLAG) ;EXECUTE IT ;GET FLAG
C913: 4C B7 C9 C918: C929: C920: C920: C921: C922: C923: C933: C933:<	18 20 * START A 21 * WE HAN 22 * E - 23 * E - 24 * F - 25 * I - 25 * I - 26 * J - 27 * K - 28 * M - 30 * T - 31 * A - 32 * C - 34 * THE F 36 37 ESCAPING 38 39 40 41 42 43 44 5 5 5 5 6 7 8 40 41 42 43 44 5 5 51 5 5 6 7 8 9 8 9 9 9 9 9 9 9	JMP I JN F SARA DLE THE N ESCAPE DLE THE IN ESCAPE CLR TO CLR TO CURSOR EO CURSOR CURSOR RESTRICT TORN GOTO 40 GOTO 40 GOTO 40 GUT ARM JSR JSR JSR JSR JSR JSR BCC AND CMP BCC AND EQU BCC AND EQU JSR BC0 BFL BC1 BFL BC2 JSR SSR JSR JSR JSR BMI JMP	NDESC SEQUENCE: OLLOWING DNI EAR FT GHT WN TO UPPERCASI ESC-R OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE OLUMN MODE SCON ESCON	;=>ND, IT'S NORMAL ES:

0945:09 11	63		CMP	#\$11	IS IT ESC-CTLQ?
C945:C9 11 C947:D0 OB C954	64		BNE		; =>ND
C949:20 AA CD	65		JSR	QUIT	DO THE QUITTING STUFF
C94C: A9 98 C94E: 8D 7B 06	66 67		LDA	#\$98 CHAR	; THE CHARACTER
C94E: 8D 7B 08	68		JMP	BIORET	;=>QUIT THE CARD FOREVER
C954:	69	*	011	DIGNET	Jesdorf the only fold fer
C954: C954		ESCSPEC2	EQU	*	
C954:C9 52	71		CMP	# 'R '	IS IT ESC-R?
C956: DO OB C963	72		BNE.	ESCSPEC3	; =>NO
C958: AD FB 04	73		LDA	MODE	;YES, SET IT
C95B:09 80 C95D:8D FB 04	74 75		ORA STA	#M. ESCR MODE	
C960: 4C 05 C9		ESCNONE	JMP	B. INPUT	QUIT ESCAPE MODE
C963:	77				
C963: C963		ESCSPEC3	EQU	*	
C963: C9 54	79		CMP	#'T'	IS IT ESC-T?
C965: D0 F9 C960	80		BNE	ESCNONE MODE	;=>NOTHING
C967: AD FB 04 C96A: 29 7F	81 82		LDA AND	#255-M. ESCR	
C96C 8D FB 04	83		STA	MODE	
C96C: 8D FB 04 C96F: 4C 05 C9	84		JMP	B. INPUT	; QUIT ESCAPE MODE
C972: C972		ESCTAB	EQU	*	
C972: 40	87		ASC	·e ·	
C973: 41	88		ASC	'A' 'B'	HANDLE OLD ESCAPES
C974: 42 C975: 43	89 90		ASC	.с.	
C975: 43 C976: 44	91		ASC	'a'	
C977: 45	92		ASC	'E'	
C978: 46	93		ASC	'F'	
C979: 49	94		ASC	'I'	
C97A: 4A	95		ASC	، ر.	
C97B: 4B	96 97		ASC	'K' 'M'	
C97C: 4D C97D: 34	98		ASC	·4·	
C97E: 38	99		ASC	'8'	
C97F: 08	100		DFB	\$08	LEFT ARROW
C980: 0A	101		DFB	\$0A	DOWN ARROW
C981: 0B C982: 15	102		DFB	\$OB \$15	;UP ARROW ;RITE ARROW
C982:15 C983: 0011		ESCNUM	EQU	⇒15 *-ESCTAB	RITE ARROW
C983: 0011	105	LOCINOI	MSB	ON	
C983: C983	106	ESCCHAR	EQU	*	
C983: 0C	107		DFB	\$00+\$00	e: FORMFEED
C984: 1C	108		DFB	\$10	;A: FS ;B: BS
C785: 08 C986: 0A	109 110		DFB DFB	\$08 \$0A) B: BS) C: LF
C988: 0A	111		DFB	\$1F	; D: US
C788: 1D	112		DFB	\$1D+\$00	E: GS
C989: 0B	113		DFB	\$0B+\$00 \$1F+\$80	F: VT
C98A: 9F	114		DFB	\$1F+\$80	; I: US (STAY ESC) ; J: BS (STAY ESC)
C78B: 88 C98C: 9C	115		DFB DFB	\$08+\$80 \$1C+\$80	; J: BS (STAY ESC) ; K: FS (STAY ESC)
C980: 90			DFB	\$IC+\$60	M: LF (STAY ESC)
	117		DEB		
C98E: 11	117 118		DFB	\$0A+\$80 \$11+\$00	; 4 : DC1
C98E: 11 C98F: 12	118 119		DFB	\$11+\$00 \$12+\$00	;4 :DC1 ;8 :DC2
C98E: 11 C98F: 12 C990: 88	118 119 120		DFB DFB DFB	\$11+\$00 \$12+\$00 \$08+\$80	;4 :DC1 ;8 :DC2 ;<:BS (STAY ESC)
C78E: 11 C78F: 12 C790: 88 C791: 8A	118 119 120 121		DFB DFB DFB DFB	\$11+\$00 \$12+\$00 \$08+\$80 \$0A+\$80	;4 :DC1 ;8 :DC2 ;<-:BS (STAY ESC) ;DN:LF (STAY ESC)
C98E: 11 C98F: 12 C990: 88 C991: 8A C992: 9F	118 117 120 121 122		DFB DFB DFB DFB DFB	\$11+\$00 \$12+\$00 \$08+\$80 \$0A+\$80 \$1F+\$80	;4 :DC1 ;8 :DC2 ;<-:BS (STAY ESC) ;DN:LF (STAY ESC) ;UP:US (STAY ESC)
C98E: 11 C98F: 12 C990: 88 C991: 8A C992: 9F C993: 9C	118 119 120 121 122 123		DFB DFB DFB DFB	\$11+\$00 \$12+\$00 \$08+\$80 \$0A+\$80	;4 :DC1 ;8 :DC2 ;<-:BS (STAY ESC) ;DN:LF (STAY ESC)
C98E: 11 C98F: 12 C990: 88 C991: 8A C992: 9F	118 117 120 121 122		DFB DFB DFB DFB DFB	\$11+\$00 \$12+\$00 \$08+\$80 \$0A+\$80 \$1F+\$80	;4 :DC1 ;8 :DC2 ;<-:BS (STAY ESC) ;DN:LF (STAY ESC) ;UP:US (STAY ESC)
C98E: 11 C99F: 12 C99F: 88 C991: 8A C992: 9F C992: 9F C994: C994: C994: C994:	118 119 120 121 122 123 124 126 127	* PASCAL S	DFB DFB DFB DFB DFB DFB DFB	\$11+\$00 \$12+\$00 \$08+\$80 \$0A+\$80 \$1F+\$80 \$1C+\$80	;4 :DC1 :8 :DC2 : <bs (stay="" esc)<br="">;DN:LF (STAY ESC) ;DN:LF (STAY ESC) ;UP:US (STAY ESC) ;->:FS (STAY ESC)</bs>
C98E: 11 C99F: 12 C99F: 12 C990: 88 C991: 8A C992: 9F C993: 9C C994: C994: C994: C994:	118 119 120 121 122 123 124 126 127 128	* PASCAL S	DFB DFB DFB DFB DFB DFB STATUS:	\$11+\$00 \$12+\$00 \$08+\$80 \$0A+\$80 \$1F+\$80 \$1C+\$80	;4 :DC1 ;8 :DC2 ;<-:BS (STAY ESC) ;DN:LF (STAY ESC) ;UP:US (STAY ESC)
C 798E: 11 C 797: 12 C 797: 12 C 797: 8A C 7973: 7C C 7974: C	118 119 120 121 122 123 124 126 127 128 129	* PASCAL S	DFB DFB DFB DFB DFB DFB DFB STATUS: EQU	\$11+\$00 \$12+\$00 \$08+\$80 \$0A+\$80 \$1F+\$80 \$1C+\$80	;4 :DC1 ;8 :DC2 ;<->BS (STAY ESC) ;DN:LF (STAY ESC) ;DN:LF (STAY ESC) ;UP:US (STAY ESC) ;->:FS (STAY ESC)
C98E:11 C98F:12 C990:88 C991:8A C992:9F C993:9C C994: C995: C996: C994: C995:	118 119 120 121 122 123 124 126 127 128 129 130	* PASCAL S	DFB DFB DFB DFB DFB DFB DFB DFB ETATUS: EQU TAX	\$11+\$00 \$12+\$00 \$08+\$80 \$08+\$80 \$0A+\$80 \$16+\$80 \$16+\$80 *	; 4 : DC1 ; 8 : DC2 ;: BS (STAY ESC) ; DN: LF (STAY ESC) ; JN: LF (STAY ESC) ; JP: US (STAY ESC) ; ->: FS (STAY ESC)
C98E:11 C98F:12 C990:88 C992:9F C991:8A C992:9F C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C992:20 C8 CF	118 119 120 121 122 123 124 126 127 128 129	* PASCAL S	DFB DFB DFB DFB DFB DFB DFB STATUS: EQU	\$11+\$00 \$12+\$00 \$08+\$80 \$08+\$80 \$0A+\$80 \$1F+\$80 \$1C+\$80 * PSETUP	;4:DC1 ;8:DC2 ;<-:BS (STAY ESC) ;DN:LF (STAY ESC) ;UP:US (STAY ESC) ;->:FS (STAY ESC)
C98E:11 C98F:12 C990:88 C992:9F C991:8A C992:9F C993:9C C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C995:20 C8 CF C995:8A C999:80 0 3 C99E	118 119 120 121 122 123 124 126 127 128 129 130 131 132 133	* PASCAL S	DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	\$11+\$00 \$12+\$00 \$08+\$80 \$08+\$80 \$0A+\$80 \$16+\$80 \$16+\$80 *	;4 :DC1 ;8 :DC2 ; <bs (stay="" esc)<br="">;DN:LF (STAY ESC) ;DV:US (STAY ESC) ;->:FS (STAY ESC) ;->:FS (STAY ESC) </bs>
C98E:11 C98F:12 C990:88 C991:8A C992:9F C993:9C C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C999:20 C8 CF C998:38 C999:00 O3 C998:20 C994: C998: C994: C998: C998: C998: C998: C998: C998: C998: C998: C998: C998: C	118 119 120 121 122 123 124 126 127 128 129 130 131 132 133 134	* PASCAL S	DFB DFB DFB DFB DFB DFB DFB DFB STATUS: EQU TAX JSR TXA BNE SEC	\$11+\$00 \$12+\$00 \$08+\$80 \$0A+\$80 \$1F+\$80 \$1C+\$80 * PSETUP PSTATUS2	;4:DC1 ;8:DC2 ;<-:BS (STAY ESC) ;DN:LF (STAY ESC) ;UP:US (STAY ESC) ;->:FS (STAY ESC)
C98E:11 C98F:12 C990:88 C992:9F C991:8A C992:9F C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C995:20 C8 CF C998:8A C999:00 03 C99E C998:3B	118 119 120 121 122 123 124 126 127 128 129 130 131 132 133 134 135	* PASCAL S	DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	\$11+\$00 \$12+\$00 \$08+\$80 \$08+\$80 \$0A+\$80 \$1F+\$80 \$1C+\$80 * PSETUP	;4 :DC1 ;8 :DC2 ; <bs (stay="" esc)<br="">;DN:LF (STAY ESC) ;DV:US (STAY ESC) ;->:FS (STAY ESC) ;->:FS (STAY ESC) </bs>
C98E:11 C98F:12 C990:88 C991:8A C992:9F C993:9C C994:C994: C994:C994: C994:C994:C994	118 119 120 121 122 123 124 126 127 128 127 128 129 130 131 132 133 134 135 136	* PASCAL S	DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	\$11+\$00 \$12+\$00 \$08+\$80 \$0A+\$80 \$1F+\$80 \$1C+\$80 * PSETUP PSTATUS2	;4 :DC1 ;8 :DC2 ; <bs (stay="" esc)<br="">;DN:LF (STAY ESC) ;DV:US (STAY ESC) ;->:FS (STAY ESC) ;->:FS (STAY ESC) </bs>
C98E:11 C98F:12 C990:18 C990:18 C992:9F C994:0 C994:10 C993:20 C993:20 C993:20 C999:10 C999:10 C999:10 C999:10 C999:10 C999:10 C991:20 C991:30 C992:10 C992:10 C994:10 C994:10 C995:10 C995:10 C995:10 C995:10 C995:10 C995:10 C995:10 C995:10	118 119 120 121 122 123 124 126 127 128 127 128 129 130 131 132 133 134 135 136 137	* PASCAL S	DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	\$11+\$00 \$12+\$00 \$08+\$80 \$0A+\$80 \$1F+\$80 \$1C+\$80 * PSETUP PSETUP PSTATUS2	;4 :DC1 ;8 :DC2 ;<-:BS (STAY ESC) ;DN:LF (STAY ESC) ;UP:US (STAY ESC) ;UP:US (STAY ESC) ;-:FS (STAY ESC) :
C98E:11 C98F:12 C990:18 C992:9F C993:9C C994:12 C994:4 C994:2 C994:4 C994:2 C994:2 C994:4 C994:2 C998:20 C8 C998:20 C9 C998:20 C8 C998:20 C8 C998:20 C8 C998:20 C8 C998:20 C9 C998:20 C8 C998:20 C8 C998:20 C8 C998:20 C8 C998:20 C8 C998:20 C8	$\begin{array}{c} 118\\ 119\\ 120\\ 121\\ 122\\ 123\\ 124\\ 126\\ 127\\ 128\\ 130\\ 131\\ 132\\ 133\\ 134\\ 135\\ 136\\ 137\\ 138\\ 139 \end{array}$	* PASCAL S	DFB DFB DFB DFB DFB SFATUS: STATUS: EQU TAX JSR TXA BNE SEC BCS EQU CMP BEQ	\$11+\$00 \$12+\$00 \$08+\$80 \$0A+\$80 \$1C+\$80 \$1C+\$80 * PSETUP PSTATUS2 PSTATUS2 * # 1 PSTATUS3	;4 :DC1 ;8 :DC2 ;<-:BS (STAY ESC) ;DN:LF (STAY ESC) ;UP:US (STAY ESC) ;UP:US (STAY ESC) ;->:FS (STAY ESC) :
C98E:12 C98F:12 C970:88 C970:88 C992:9F C973:9C C974: C978:30 C979:30 C979:30 C979:30 C979:30 C979:30 C979:5 C975:5 C975:2 C975:30 C975:30 C975:5 C975:5 C975:30 C975:30 C975:30 C975:30 50 C975:5	$\begin{array}{c} 118\\ 119\\ 120\\ 121\\ 122\\ 123\\ 124\\ 126\\ 127\\ 128\\ 130\\ 131\\ 132\\ 133\\ 134\\ 135\\ 136\\ 137\\ 138\\ 139\\ 140 \end{array}$	* PASCAL S	DFB DFB DFB DFB DFB DFB DFB DFB STATUS: EQU TAX JSR EQU TAX BNE SECS EQU CMP BEQ EQU CMP BEQ LDX	\$11+\$00 \$12+\$00 \$08+\$80 \$0A+\$80 \$1F+\$80 \$1C+\$80 \$1C+\$80 PSETUP PSTATUS2 PSTATUS2 * 1	;4 :DC1 ;8 :DC2 ;<-:BS (STAY ESC) ;DN:LF (STAY ESC) ;UP:US (STAY ESC) ;UP:US (STAY ESC) ;-:FS (STAY ESC) :
C98E:11 C98F:12 C990:88 C990:88 C992:9F C974:0 C974:40 C979:30 C979:50	$\begin{array}{c} 118\\ 119\\ 120\\ 122\\ 123\\ 124\\ 126\\ 127\\ 128\\ 127\\ 131\\ 132\\ 133\\ 134\\ 135\\ 136\\ 137\\ 138\\ 138\\$	* PASCAL S	DFB DFB DFB DFB DFB DFB STATUS: STATUS	\$11+\$00 \$12+\$00 \$08+\$80 \$0A+\$80 \$1C+\$80 \$1C+\$80 * PSETUP PSTATUS2 PSTATUS2 * # 1 PSTATUS3	;4 :DC1 ;8 :DC2 ;<-:BS (STAY ESC) ;DN:LF (STAY ESC) ;UP:US (STAY ESC) ;UP:US (STAY ESC) ;->:FS (STAY ESC) :
C98E:11 C98F:12 C990:88 C992:9F C991:8A C992:9F C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C998:8A C999:00 C998:8B C998:8B C998:8B C998:3C C998:3C C998 C998:3C C998 C9	118 119 120 121 122 123 124 126 127 130 131 132 134 135 136 137 138 137 140 141	* PASCAL S PSTATUS * PSTATUS2	DFB DFB DFB DFB DFB DFB DFB DFB STATUS: EQU TAX JSR EQU TAX BNE SECS EQU CMP BEQ EQU CMP BEQ LDX	\$11+\$00 \$12+\$00 \$08+\$80 \$0A+\$80 \$1C+\$80 \$1C+\$80 * PSETUP PSTATUS2 PSTATUS2 * # 1 PSTATUS3	;4 :DC1 ;8 :DC2 ;<-:BS (STAY ESC) ;DN:LF (STAY ESC) ;UP:US (STAY ESC) ;UP:US (STAY ESC) ;->:FS (STAY ESC) :
C98E:11 C98F:12 C990:88 C990:88 C992:9F C973:9C C974:0 C974:4 C974:4 C974:4 C974:4 C974:4 C974:4 C974:4 C974:4 C974:4 C974:5 C974:5 C974:8 C979:8 C979:8 C979:8 C979:8 C979:9 C979:8 C979:9	118 119 120 121 122 123 124 126 127 128 127 128 130 131 132 133 134 135 136 137 138 139 140 141 142	* PASCAL S PSTATUS * PSTATUS2	DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	*11+*00 \$12+*00 \$08+*80 \$0A+*80 \$1F+*80 \$1C+*80 * PSETUP PSTATUS2 PSTATUS2 PSTATUS3 #3	;4 :DC1 ;8 :DC2 ;<-:BS (STAY ESC) ;DN:LF (STAY ESC) ;UP:US (STAY ESC) ;UP:US (STAY ESC) ;->:FS (STAY ESC) :
C98E:11 C98F:12 C990:88 C992:9F C991:8A C992:9F C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C998:8A C999:00 C998:8B C998:8B C998:8B C998:3C C998:3C C998 C998:3C C998 C9	118 119 120 121 122 123 124 126 127 128 127 128 130 131 132 133 134 135 136 137 138 139 140 141 142	* PASCAL S PSTATUS * PSTATUS2	DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	*11+*00 \$12+*00 \$08+*80 \$0A+*80 \$1F+*80 \$1C+*80 * PSETUP PSTATUS2 PSTATUS2 PSTATUS3 #3	<pre>;4 :DC1 ;8 :DC2 ;<-:BS (STAY ESC) ;DN:LF (STAY ESC) ;UP:US (STAY ESC) ;UP:US (STAY ESC) ;-:FS (STAY ESC) ;-:FS (STAY ESC) ;:STAY ;SAVE REQUEST CODE ;SETUP ZP STUFF ;IS IT 'READY FOR OUTPUT?' ;=:NO ;YES, READY FOR OUTPUT ;IS IT 'ANY INPUT?' ;=:YES ;IORESULT='ILGL OPERATION'</pre>
C98E:11 C98F:12 C990:18 C990:88 C992:9F C991:8A C992:9F C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C994: C998:8A C998:8A C998:38 C998:38 C998:38 C998:38 C998:38 C998:38 C998:38 C998:38 C998:38 C998:38 C998:38 C998:38 C998:38 C998:38 C998:40 C998 C	118 119 120 121 122 123 124 124 127 128 127 130 131 134 135 134 135 136 137 138 134 137 138 134 144 145	* PASCAL S PSTATUS * PSTATUS2	DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	*11+*00 \$12+*00 \$08+*80 \$0A+*80 \$1F+*80 \$1C+*80 * PSETUP PSTATUS2 PSTATUS2 PSTATUS3 #3	;4 :DC1 ;8 :DC2 ;<-:BS (STAY ESC) ;DN:LF (STAY ESC) ;UP:US (STAY ESC) ;UP:US (STAY ESC) ;->:FS (STAY ESC) :
C98E:11 C98E:12 C990:88 C990:88 C992:9F C973:9C C974:0 C974:4 C974:30 C974:4 C974:4 C974:4 C974:4 C974:4 C974:4 C974:4 C974:4 C974:30 C978:30 C978:38 C979:30 C979:30 C979:30 C978:4 C978:5 C978:5 C978:7 C978:7 C978:8 C978:9 C978:9 C978:10 C978:2 C978:2 C978:4 C978:5 C978:5 C978:5	118 119 120 121 122 123 124 124 124 125 127 130 132 133 133 134 135 136 137 138 137 144 144 144 144 144 144 147	* PASCAL S PSTATUS * PSTATUS2	DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	*11+*00 \$12+*00 \$08+*80 \$0A+*80 \$1F+*80 \$1C+*80 * PSETUP PSTATUS2 PSTATUS2 PSTATUS3 #3	<pre>;4 :DC1 ;8 :DC2 ;<-:BS (STAY ESC) ;DN:LF (STAY ESC) ;UP:US (STAY ESC) ;UP:US (STAY ESC) ;-:FS (STAY ESC) ;-:FS (STAY ESC) ;:STAY ;SAVE REQUEST CODE ;SETUP ZP STUFF ;IS IT 'READY FOR OUTPUT?' ;=:NO ;YES, READY FOR OUTPUT ;IS IT 'ANY INPUT?' ;=:YES ;IORESULT='ILGL OPERATION'</pre>
C98E:11 C98F:12 C990:88 C992:9F C991:8A C992:9F C993:9C C994: C994: C994: C994: C994: C994: C994: C994: C994: C998:0 C998:8A C998:8A C998:38 C998:38 C998:38 C998:38 C998:38 C998:38 C998:38 C998:38 C998:38 C998:38 C998:38 C998:4 C998:	118 119 120 121 122 123 124 126 127 130 131 132 133 134 137 136 137 138 137 140 141 142 144 145 144	* PASCAL S PSTATUS * PSTATUS2	DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	*11+*00 \$12+*00 \$08+*80 \$0A+*80 \$1F+*80 \$1C+*80 * PSETUP PSTATUS2 PSTATUS2 PSTATUS3 #3	<pre>;4 :DC1 ;8 :DC2 ;<-:BS (STAY ESC) ;DN:LF (STAY ESC) ;UP:US (STAY ESC) ;UP:US (STAY ESC) ;-:FS (STAY ESC) ;-:FS (STAY ESC) ;:STAY ;SAVE REQUEST CODE ;SETUP ZP STUFF ;IS IT 'READY FOR OUTPUT?' ;=:NO ;YES, READY FOR OUTPUT ;IS IT 'ANY INPUT?' ;=:YES ;IORESULT='ILGL OPERATION'</pre>
C98E:11 C98E:12 C990:88 C990:88 C992:9F C974:0 C974:4 C973:20 C8 C978:38 C978:38 C978:39 C978:39 C978:4 C978:50 C978:50 C978:50 C978:60 C978:700 C978:700 C978:00	118 119 120 121 122 123 124 126 127 128 127 130 132 133 132 133 134 135 134 137 138 137 138 140 141 144 144 144 144 144 144 144	* PASCAL S PSTATUS * PSTATUS2	DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	\$11+\$00 \$12+\$00 \$08+\$80 \$0A+\$80 \$1F+\$80 \$1C+\$80 PSETUP PSTATUS2 PSTATUS2 PSTATUS2 * 1 PSTATUS3 #3	<pre>;4 :DC1 ;8 :DC2 ;<-:BS (STAY ESC) ;DN:LF (STAY ESC) ;UP:US (STAY ESC) ;UP:US (STAY ESC) ;-:FS (STAY ESC) ;-:FS (STAY ESC) ;:STAY ;SAVE REQUEST CODE ;SETUP ZP STUFF ;IS IT 'READY FOR OUTPUT?' ;=:NO ;YES, READY FOR OUTPUT ;IS IT 'ANY INPUT?' ;=:YES ;IORESULT='ILGL OPERATION'</pre>
C98E:11 C98F:12 C990:88 C992:9F C991:8A C992:9F C993:9C C994: C994: C994: C994: C994: C994: C994: C994: C994: C998:0 C998:8A C998:8A C998:38 C998:38 C998:38 C998:38 C998:38 C998:38 C998:38 C998:38 C998:38 C998:38 C998:38 C998:4 C998:	118 119 120 121 122 123 124 126 127 130 131 132 133 134 137 136 137 138 137 140 141 142 144 145 144	* PASCAL S PSTATUS * PSTATUS2	DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	*11+*00 \$12+*00 \$08+*80 \$0A+*80 \$1F+*80 \$1C+*80 * PSETUP PSTATUS2 PSTATUS2 PSTATUS3 #3	<pre>;4 :DC1 ;8 :DC2 ;<-:BS (STAY ESC) ;DN:LF (STAY ESC) ;UP:US (STAY ESC) ;UP:US (STAY ESC) ;-:FS (STAY ESC) ;-:FS (STAY ESC) ;:STAY ;SAVE REQUEST CODE ;SETUP ZP STUFF ;IS IT 'READY FOR OUTPUT?' ;=:NO ;YES, READY FOR OUTPUT ;IS IT 'ANY INPUT?' ;=:YES ;IORESULT='ILGL OPERATION'</pre>
C98E:11 C98F:12 C990:88 C992:9F C991:8A C992:9F C994:C C994:C C994:C C994:C C994:C C994:C C994:C C994:C C994:C C995:20 C995:20 C995:20 C995:20 C995:20 C995:20 C995:20 C995:20 C995:20 C995:20 C995:20 C995:20 C995:20 C995:20 C995:20 C995:20 C996:C C C996:C C C C996:C C C C996:C C C C996:C C C C C C C C C C C C C C C C C C C	$\begin{array}{c} 118\\ 119\\ 120\\ 121\\ 122\\ 122\\ 124\\ 126\\ 127\\ 128\\ 127\\ 128\\ 127\\ 128\\ 127\\ 131\\ 131\\ 132\\ 133\\ 134\\ 137\\ 138\\ 137\\ 141\\ 142\\ 143\\ 144\\ 145\\ 144\\ 145\\ 144\\ 145\\ 151\\ 152 \end{array}$	* PASCAL S PSTATUS * PSTATUS2	DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	*11+400 \$12+500 \$08+580 \$0A+580 \$1C+580 *1C+580 * PSETUP PSTATUS2 PSTATUS2 PSTATUS2 #3 UT HODK: *-\$C9AA 2, 'C9AA	<pre>;4 :DC1 ;8 :DC2 ;<- BS (STAY ESC) ;DN:LF (STAY ESC) ;DN:LF (STAY ESC) ;->:FS (STAY ESC) ;->:FS (STAY ESC) ;->:FS (STAY ESC) ;->:FS (STAY ESC) ;:SAVE REQUEST CODE ;SAVE REQUEST COD</pre>
C98E:11 C98E:12 C970:88 C970:88 C970:88 C971:8A C973:7C C794: C798:30 C979:30 C979:30 C979:30 C979:30 C979:30 C979:30 C979:50 C979:50 C974:30 C974:0 C974:10 C974:10 C974:00 C974:00 C974:00 C974:00 C974:00 C974:00 C974:00 C974:00 C974:00	$\begin{array}{c} 118\\ 119\\ 120\\ 121\\ 122\\ 124\\ 127\\ 128\\ 124\\ 127\\ 128\\ 130\\ 131\\ 134\\ 135\\ 134\\ 135\\ 136\\ 137\\ 138\\ 140\\ 141\\ 144\\ 144\\ 144\\ 144\\ 144\\ 144$	* PASCAL S PSTATUS * PSTATUS2	DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	\$11+\$00 \$12+\$00 \$08+\$80 \$0A+\$80 \$1F+\$80 \$1F+\$80 \$1C+\$80 PSETUP PSTATUS2 PSTATUS2 PSTATUS2 #1 PSTATUS3 #3 UT HODK: 	<pre>;4 :DC1 ;8 :DC2 ;<-:BS (STAY ESC) ;DN:LF (STAY ESC) ;DV:US (STAY ESC) ;->:FS (STAY ESC) ;->:FS (STAY ESC) ;->:FS (STAY ESC) ;->:FS (STAY ESC) ;:SAVE REQUEST CODE ;SAVE REQUEST CODE ;SETUP ZP STUFF ;IS IT 'READY FOR OUTPUT?' ;=>NO ;IS IT 'ANY INPUT?' ;=>VES ;IORESULT='ILGL OPERATION' ;PADDING HOOK ALIGNMENT' ;QET OUTPUT CHARACTER</pre>
C98E:11 C98F:12 C990:88 C992:9F C991:8A C992:9F C994:C C994:C C994:C C994:C C994:C C994:C C994:C C994:C C994:C C995:20 C995:20 C995:20 C995:20 C995:20 C995:20 C995:20 C995:20 C995:20 C995:20 C995:20 C995:20 C995:20 C995:20 C995:20 C995:20 C996:C C C996:C C C C996:C C C C996:C C C C996:C C C C C C C C C C C C C C C C C C C	$\begin{array}{c} 118\\ 119\\ 120\\ 121\\ 122\\ 122\\ 124\\ 126\\ 127\\ 128\\ 127\\ 128\\ 127\\ 128\\ 127\\ 131\\ 131\\ 132\\ 133\\ 134\\ 137\\ 138\\ 137\\ 141\\ 142\\ 143\\ 144\\ 145\\ 144\\ 145\\ 144\\ 145\\ 151\\ 152 \end{array}$	* PASCAL S PSTATUS * PSTATUS2 * PASCAL	DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	*11+400 \$12+500 \$08+580 \$0A+580 \$1C+580 *1C+580 * PSETUP PSTATUS2 PSTATUS2 PSTATUS2 #3 UT HODK: *-\$C9AA 2, 'C9AA	<pre>;4 :DC1 ;8 :DC2 ;<- BS (STAY ESC) ;DN:LF (STAY ESC) ;DN:LF (STAY ESC) ;->:FS (STAY ESC) ;->:FS (STAY ESC) ;->:FS (STAY ESC) ;->:FS (STAY ESC) ;:SAVE REQUEST CODE ;SAVE REQUEST COD</pre>

C9BO: C9BO	156 157		EQU	*	TO THERE & KEVERECCO
C9BO: AD 00 C0 C9B3: 0A	158		ASL	A	; IS THERE A KEYPRESS? ; STROBE>CARRY
C9B4: A2 00 C9B6: 60	159	PSTATUS4	LDX	#0	; IORESULT='GOOD'
C9B7:	162				
C9B7: C9B7:	164	* BASIC * NOT A	N ESCA	PE SEQUENCE-	
C9B7: C9B7: C9B7	165		EQU		NOT ESCAPE KEY
C9B7:	167	*		î	
C9B7: C9 95	168		CMP BNE	#\$95 8 NORICK) IS IT PICK? ;=>NOPE ;YOU CAN PICK YER FRIENDS ;YES, PICK THE CHAR ;ALWAYS PICK AS NORMAL ;SAVE AS KEYSTROKE
C9B9: DO OB C9C6 C9BB: AC 7B 05	170		LDY	OURCH	YOU CAN PICK YER FRIENDS
C9BE: 20 01 CF C9C1: 09 80	171		JSR	PICK	YES, PICK THE CHAR
C9C3:8D 7B 06	172		STA	CHAR	SAVE AS KEYSTROKE
C9C6:	174	*			
C9C6: C9C6:	175	* TRACK QUD * RESTRICT	-UPPER	CASE FEATURE	HE.
C9C6:	177	*			
C9C6: C9C6 C9C6: AD FB 04	178	B. NOPICK	EQU	* MODE	ARE WE DOING LITERAL INPUT?
C9C9:29 10	180		AND	#M. LIT	
C9CB: DO 12 C9DF C9CD:	181 182		BNE	B. CHKCAN	;=>YES
C9CD:	183	* LITERAL I	NPUT'S	INACTIVE. S	EE IF
C9CD: C9CD:	184		TART L	ITERAL INPUT	
C9CD: AD 7B 06	186		LDA	CHAR	GET THE CHAR
C9D0: C9 A2	187		CMP BEQ	#\$A2	;GET THE CHAR ;IS IT A DOUBLE QUOTE? ;=>YES, FLIP LITERAL MODE ;IS HE MOVING LEFT?
C9D2: FU 23 C9F7 C9D4: C9 88	188			#\$88	; IS HE MOVING LEFT? ;=>NOPE, JUST REG CHAR
CTUB. DO SE CHUM			BNE	B. FIXCHR	;=>NOPE, JUST REG CHAR
C9D8: 20 27 CA C9DB: DO 2D CAOA	191 192		BNE	GETPRIOR B. FIXCHR	; GRAB PRIOR CHAR ; =>NOT DELETING A QUDTE
C9DD: FO 18 C9F7	193		BEQ	B. FLIP	;IS HE MOVING LEFT? ;=>NDFE, JUST REG CHAR ;GRAB PRIOR CHAR ;=>NOT DELETING A QUOTE ;(ALWAYS) HIE'S DELETED THE QUOTE
C9DF: C9DF:	194	* LITERAL I	NPUT 'S	ACTIVE SEE	IF
C9DF:	196	* IT SHOUL	DBEC	ANCELLED YET	
C9DF: C9DF: C9DF	197		EQU	*	
C9DF: AD 7B 06	199	D. CHINCHIN	LDA	CHAR	; GET CURRENT CHAR
C9E2: C9 A2	200		CMP	#\$A2	; IS CURRENT CHAR THE CLOSING QUOTE? ;=>YES
C9E4: F0 1C CA02 C9E6: C9 98	202		CMP	CHAR #\$A2 B.CANLIT #\$98	CANCEL LITERAL INPUT
C9E8: FO 18 CA02			BEG	B. CANLIT	;CANCEL LITERAL INPUT ; IF CTLX OR RETURN ; OR BACK OVER "
C9EA: C9 8D C9EC: F0 14 CA02	204		CMP	#\$BD B. CANLIT	; OR BACK OVER
C9EE: C9 88	206		CMP	#\$88	BACKSPACE?
C9F0: D0 18 CA0A C9F2: 20 27 CA	207		BNE.	B. FIXCHR GETPRIOR	;=>ND, NOT DELETING QUOTE ;GET CHAR HE'S DELETING
C9F5: DO 13 CAOA	209		BNE	B. FIXCHR	;=>NOT DELETING A QUOTE
C9F7: C9F7: C9F7	210	* B.FLIP	EQU	*	
C9F7: AD FB 04	212		LDA	MODE	FLIP THE MODE
C9FA: 49 10	213		EOR	#M.LIT MGDE	
C9FF: 4C OA CA	215		JMP	B. FIXCHR	
C9FC: 8D FB 04 C9FF: 4C 0A CA CA02: CA02 CA02: AD FB 04 CA02: AD FB 04	216	B. CANLIT	EQU	* MODE	
	218		AND		CANCEL LITERAL INPUT
CA05: 29 EF CA07: 8D FB 04	219		STA	MODE	
CADA: CADA	771		EQU	*	
			LDA	MODE	; ESC-R FACILITY ACTIVE?
CAOF: E0 13 CA24			AND	#M. ESCR B. INRET	; =>NOPE
CA11: AD FB 04	225		LDA	MODE	LITERAL INPUT ACTIVE?
CA14: 29 10 CA16: DO OC CA24	226		AND BNE	#M.LIT B.INRET	;=>YES, NO UPSHIFT
CA18: AD /B 06	228		LDA	B. INRET CHAR #\$E0	;GET THE CHAR
CA1B: C9 E0 CA1D: 90 05 CA24 CA1E: 29 DE	229		CMP BCC	#SEO	;IS CHAR LOWERCASE? ;=>NO, NO NEED TO SHIFT IT
CA1F: 29 DF	231		AND	B. INRET #SDF	RESTRICT TO U/C
CA21:8D 7B 06	231		0.114	CHAR	
CA24: CA24: CA24		B. INRET			
CH24.40 E2 00	200		JMP	* BIORET	=>RETURN TO CALLER
CA27: CA27:	237	* NAME :	GETPR	IOR	
CA27:	770	* FUNCTION	OFT C	WAR REFORE C	URSOR
CA27: CA27:	240	* INPUT : * OUTPUT	'BFQ'	IF CHAR=DBI	QUDTE
CA27:	242	:* :	'BNE '	IF NOT	
CA27: CA27:	243	* FONCTION * INPUT : * OUTPUT : * : * VOLATILE: * CALLS :	AC, T	EMP1'	
CA27:	245				
CA27: CA27: CA27	246	*	EQU	*	
CA27: AD FB 05	247	GETPRIOR	LDA	DURCY	; DON'T TRY TO LOOK
second the second of The Total					

CA2A: 0D 7B CA2D: F0 1A CA2F: 98	CA49 2	249 250 251		ORA BEG TYA	OURCH GPX	; BACK IF @ UPPER-LEFT ; CORNER OF WINDOW!!! ;SAVE Y
CA30: 48 CA31: 20 DB		252		PHA	X BS	BACK UP 1 CHAR
CA31: 20 DB CA34: AC 7B		253		LDY	OURCH	GET CH AND
CA37: 20 01	CF 2	255		JSR	PICK	; PICK PRIOR CHAR
CA3A: 09 80 CA3C: 8D 78		256		ORA STA		PICK AS NORMAL VIDED
CA3F: 20 26	cc a	258		JSR	X. FS	
CA42: 68		259		PLA		RESTORE
CA43: A8 CA44: AD 78		260		LDA	TEMP 1	1 Y
CA47: C9 A2	2	262		CMP		IS IT DBL QUOTE?
CA49: CA49:60		263	GPX	EQU	*	RETURN WITH BEG/BNE
CA49: 80	4	-04			DE PINIT	RETORN WITH BEGUBNE
CA4A:		2				
CA4A:		3	* PASCAL IN	TIALI	ATION:	
CA4A:	CA4A		PINIT1. 0	EQU	*	
CA4A: A9 22		6		LDA	#M. PASCAL+M.	PAS1. 0
CA4C: 4C 51	CA CA4F	7	PINIT	JMP	PINIT2	
CA4F: A9 20	CHAI	9	1 1011	LDA	#M. PASCAL	SAY WE'RE
CA51:		10	*			
CA51: CA51:8D FB	CA51	11 12	PINIT2	EQU	* MODE	; RUNNING PASCAL
CA54: 20 9B	CD	10		1CD	C101.00	CET ELLI DAVED LITNDOLL
CA57:20 C8	CF	14		JSR	PSETUP	SETUP ZP STUFF
CA5A:		15	* BASE ADDR	IS WRO	ING, BUT X. FF	SET FOLL 24160 WINDOW SETUP ZP STUFF FIXES IT BELDW: ASCALC
CA5A: CA5A:		16	* JSR BASUA	ALC IFU	DRUE A GUUD B	ASCALC
CA5A:			* SEE IF THE	CARD	S PLUGGED IN	E
CA5A:		19	*			
CA5A: 20 24 CA5D: F0 03		20 21		JSR	TESTCARD PIGOOD	; IS IT THERE? ; =>YES
CA5F: A2 09	CHOR	22		LDX		; IORESULT= 'NO DEVICE'
CA61:60		23		RTS		
CA62: CA62		24	* PIGOOD	EQU		
CA62:8D 01		26	F 1600D	STA	SETBOCOL	ENABLE BO STORE
CA65: BD OD	CO	27		STA	SETBOVID	, AND 80 VIDEO
CA68: 80 OF	co	28		STA		NORM+INV LCASE
CA6B: 20 42 CA6E: 20 DD		29 30		JSR	X.FF INVERT	;HOME & CLEAR IT ;PUT CURSOR THERE
	01					
CA71: A2 00		31		LDX	#Q	; IORESULT='GOOD'
CA71:A2 00 CA73:60		32		RTS		; IORESULT='GOOD'
CA71:A2 00 CA73:60 CA74:		32 10		RTS	#O DE PREAD	; IDRESULT='GOOD'
CA71:A2 00 CA73:60 CA74: CA74: CA74: CA74:		32 10 2 3	* PASCAL INF	RTS INCLUI		; IORESULT='GOOD'
CA71: A2 00 CA73: 60 CA74: CA74: CA74: CA74: CA74:		32 10 2 3 4		RTS INCLUI PUT:		; IORESULT= 'QOOD'
CA71:A2 00 CA73:60 CA74: CA74: CA74: CA74: CA74: CA74: CA74:	CA74	32 10 2 3 4		RTS INCLUI	DE PREAD	; IORESULT='GODD'
CA71:A2 00 CA73:60 CA74: CA74: CA74: CA74: CA74: CA74: CA74:20 C8 CA77:	CA74 CF	32 10 2 3 4 5 6 7		RTS INCLUI PUT: EQU JSR	DE PREAD * PSETUP	;SETUP ZP STUFF
CA71:A2 00 CA73:60 CA74: CA74: CA74: CA74: CA74: CA74: CA74: CA74:20 CB CA77:20 CA77:20 CA75:20 CA75:20 CA75:20 CA75:20 CA75:20 CA75:20 CA75:20 CA75:20 CB CA75:20 CA75:CA75:CA75:CA75:20 CA75:CA75:CA75:CA75:CA75:20 CA75:CA75:CA75:CA75:CA75:CA75:CA75:CA75:	CA74 CF	32 10 2 3 4 5 6 7 8	PREAD	RTS INCLUI PUT: EQU JSR JSR	DE PREAD * PSETUP GETKEY	SETUP ZP STUFF
CA71:A2 00 CA73:60 CA74: CA74: CA74: CA74: CA74: CA74: CA74:20 C8 CA77:	CA74 CF CB	32 10 2 3 4 5 6 7	PREAD	RTS INCLUI PUT: EQU JSR	DE PREAD * PSETUP GETKEY #\$7F	;SETUP ZP STUFF
CA71: A2 00 CA73: 60 CA74: CA74: CA74: CA74: CA74: CA74: 20 CA77: 20 CA77: 20 CA77: 20 CA77: 20 CA77: 20 CA77: 20 CA77: A2 CA72: A2 CA74: CA7	CA74 CF CB 06	32 10 2 3 4 5 6 7 8 9 10 11	PREAD	RTS INCLUI PUT: EQU JSR JSR AND STA LDX	PE PREAD * PSETUP GETKEY #\$7F CHAR #0	;SETUP ZP STUFF ;GET A KEYSTROKE ;DROP HI BIT ;SAVE THE CHAR ;IDRESULT= 'GOOD'
CA71:A2 00 CA73:60 CA74: CA74: CA74: CA74: CA74: CA74:20 CB CA77:20 15 CA77:20 15 CA76:27 7F CA7C:8D 7B CA7F:A2 00 CA81:AD FB	CA74 CF CB 06	32 10 2 3 4 5 6 7 8 9 10 11 12	PREAD	RTS INCLUI PUT: EQU JSR AND STA LDX LDA	PE PREAD * PSETUP GETKEY #57F CHAR #0 MODE	;SETUP ZP STUFF ;GET A KEYSTROKE ;DROP HI BIT ;SAVE THE CHAR
CA71: A2 00 CA72: 60 CA74: CA74: CA74: CA74: CA74: CA74: 20 CA74: 20 CA77: 20 CA77: 20 CA77: 20 CA77: 20 CA77: A2 CA77: A2 CA77: A2 CA77: A2 CA76: A2 CA77: A2 CA76: A2 CA77:	CA74 CF CB 06	32 10 2 3 4 5 6 7 8 9 10 11	PREAD	RTS INCLUI PUT: EQU JSR JSR AND STA LDX	* PSETUP GETKEY #\$7F CHAR #0 MODE #M.PAS1.0 PREADRET2	;SETUP ZP STUFF ;CET A KEYSTROKE ;DROP HI BIT ;SAVE THE CHAR ;IORESULT='GODD' ;ARE WE IN 1.0-MODE? ;=>NOPE
CA71: A2 00 CA73: 60 CA73: 60 CA74: CA74: CA74: CA74: CA74: CA74: CA74: CA74: CA74: CA74: CA77: CA77: CA77: CA77: CA77: A2 00 CA77: A2 00 CA77: A2 00 CA81: AD FB CA84: 27 02 CA88: A2 C3	CA74 CF CB 06 04	32 10 23 45 67 89 10 11 12 13 45 10	PREAD	RTS INCLUI PUT: EQU JSR AND STA LDX LDA AND	* PSETUP GETKEY #\$7F CHAR #0 MODE #M.PAS1.0 PREADRET2	SETUP ZP STUFF GET A KEYSTROKE DROP HI BIT SAVE THE CHAR IDREBULT= GODD ' ARE WE IN 1.0-MODE?
CA71: A2 00 CA73: 60 CA74: C CA74: C CA74: C CA74: C CA74: C CA77: C CA77: C CA77: C CA77: C CA77: C CA77: C CA77: C CA76: A2 7 C CA76: C C CA88: C C CA88: C C	CA74 CF 06 04 CABA	32 10 23 45 67 89 10 11 12 13 45 67 89 10 11 12 13 45 67	*	RTS INCLUI PUT: EQU JSR AND STA LDX LDA AND BEO LDX	* PSETUP GETKEY #\$7F CHAR #0 MODE #M.PAS1.0 PREADRET2	;SETUP ZP STUFF ;CET A KEYSTROKE ;DROP HI BIT ;SAVE THE CHAR ;IORESULT='GODD' ;ARE WE IN 1.0-MODE? ;=>NOPE
CA71: A2 00 CA73: 60 CA74: C CA74: C CA74: C CA74: C CA74: C CA74: 20 CA77: 20 CA77: 20 CA77: 20 CA77: 20 CA77: 20 CA77: 42 CA77: 42 CA77: 42 CA77: 42 CA77: 42 CA76:	СА74 СF СВ 06 04 САВА САВА	32 10 23 45 67 89 10 11 12 13 45 67 89 10 11 12 13 45 67	PREAD	RTS INCLUI PUT: EQU JSR AND STA LDX LDA AND BEQ	* PSETUP GETKEY #\$7F GHAR #0 MODE #M.PAS1 0 PREADRET2 #CON00 *	;SETUP ZP STUFF ;GET A KEYSTROKE ;DROP HI BIT ;SAVE THE CHAR ;IORESULT= 'GOOD ' ;ARE WE IN 1. 0-MODE? ;=>NOPE ;YES, RETURN CN IN X
CA71: A2 00 CA73: A0 CA74: CA74: CA74: CA74: CA74: CA74: CA74: CA74: CA74: CA74: CA74: CA74: CA74: CA77: CA77: CA77: A2 01 CA77: A2 01 CA77: A2 01 CA77: A2 02 CA81: A0 FB CA88: CA88: CA88: CA88: A2 78 CA88: A0 78 CA88: A0 78	СА74 СF СВ 06 04 САВА САВА	32 10 2 3 4 5 6 7 8 9 10 11 2 3 4 5 11 12 3 14 5 16 7 18 9 10 11 12 13 14 5 16 17 18 9	*	RTS INCLUI PUT: EQU JSR AND STA LDX LDX LDX LDX LDX LDX LDX LDX LDX LDX	* PSETUP GETKEY #\$7F CHAR #0 MODE #M PASL 0 PREADRET2 #CNOO * CHAR	;SETUP ZP STUFF ;CET A KEYSTROKE ;DROP HI BIT ;SAVE THE CHAR ;IORESULT='GODD' ;ARE WE IN 1.0-MODE? ;=>NOPE
CA71: A2 00 CA73: 60 CA74: C CA74: C CA74: C CA74: C CA74: C CA77: C C CA77: C C C CA77: C C CA77: C C C CA77: C C C C C CA77: C C C C C C C C C C C C C C C C C C C	СА74 СF СВ 06 04 САВА САВА	32 10 2 3 4 5 6 7 8 9 10 11 2 3 4 5 11 12 3 14 15 16 7 18 9 10 11 2 13 14 15 16 17 18 19 11	*	RTS INCLUI PUT: EQU JSR AND STA LDX LDX LDX LDX LDX LDX LDX LDX LDX LDX	* PSETUP GETKEY #\$7F GHAR #0 MODE #M.PAS1 0 PREADRET2 #CON00 *	;SETUP ZP STUFF ;GET A KEYSTROKE ;DROP HI BIT ;SAVE THE CHAR ;IORESULT= 'GOOD ' ;ARE WE IN 1. 0-MODE? ;=>NOPE ;YES, RETURN CN IN X
CA71: A2 00 CA73: A2 CA74: CA74: CA74: CA74: CA74: CA74: CA74: CA74: CA74: CA74: CA74: CA74: CA77: CA77: CA77: A2 01 CA77: A2 01 CA77: A2 07 CA77: A2 07 CA77: A2 07 CA77: A2 07 CA77: A2 07 CA84: A2 07 CA84: CA84: CA84: CA84: CA84: A2 07 CA84: A2 07 CA84: A2 CA84: CA88: CA8	СА74 СF СВ 06 04 САВА САВА	32 10 2 3 4 5 6 7 8 9 10 11 2 13 14 15 16 7 11 2	PREAD * * PREADRET2	RTS INCLUI PUT: EQU JSR AND STA LDX LDA LDA LDA LDX EQU LDA RTS INCLUI	* PSETUP GETKEY #\$7F CHAR #0 MODE #M PASL 0 PREADRET2 #CNOO * CHAR	;SETUP ZP STUFF ;GET A KEYSTROKE ;DROP HI BIT ;SAVE THE CHAR ;IORESULT= 'GOOD ' ;ARE WE IN 1. 0-MODE? ;=>NOPE ;YES, RETURN CN IN X
CA71: A2 00 CA73: 60 CA74: C CA74: C CA74: C CA74: C CA74: C CA77: C C CA77: C C C CA77: C C CA77: C C C CA77: C C C C C CA77: C C C C C C C C C C C C C C C C C C C	СА74 СF СВ 06 04 САВА САВА	320234567890112314567891011234	PREAD * * PREADRET2 * PASCAL QU	RTS INCLUI PUT: EQU JSR AND STA AND STA LDX LDX LDX LDX LDX LDX RTS INCLUI	PE PREAD * PSETUP GETKEY #\$7F CHAR #0 MDDE #M PASI 0 PREADRET2 # <cno0 * CHAR CHAR CHAR CHAR CHAR</cno0 	;SETUP ZP STUFF ;GET A KEYSTROKE ;DROP HI BIT ;SAVE THE CHAR ;IORESULT= 'GOOD ' ;ARE WE IN 1. 0-MODE? ;=>NOPE ;YES, RETURN CN IN X
CA71: A2 00 CA73: 60 CA74: C CA74: C CA74: C CA74: C CA74: 20 CA77: 20 CA77	СА74 СF 06 04 САВА 06 САВА 06	320234567890112345678912345	PREAD * * PREADRET2 * PASCAL QU	RTS INCLUI PUT: EQU JSR STA LDX LDX LDX LDX LDX EQU LDX RTS INCLUI FPUT: EQU	<pre>> PREAD * PSETUP GETKEY #\$7F CHAR #0 #MDPASI 0 PREADRET2 #<cnoo * CHAR CHAR >> PWRITE *</cnoo </pre>	; SETUP ZP STUFF ; SETUP ZP STUFF ; SETU A KEYSTROKE ; DREP HI BIT ; SAVE THE CHAR ; IORESULT= 'GODD' ; ARE WE IN 1. 0-MODE? ; =>NOPE ; YES; RETURN CN IN X ; RESTORE CHAR
CA71: A2 00 CA73: A2 CA74: C CA74: C CA74: C CA74: C CA74: C CA74: C CA74: C CA74: C CA77: C CA77: C CA77: C CA77: A2 00 CA81: AD FB CA81: AD FB CA84: C CA84: C C CA84: C C C CA84: C C C C C C C C C C C C C C C C C C C	CA74 CF 06 04 CABA 06 CABE 06	320234567890112345678910123456	PREAD * * PREADRET2 * PASCAL QU	RTS INCLUI PUT: EQU JSR AND STA LDX LDX LDX LDX LDX LDX LDX LDX LDX LDX	PE PREAD PSETUP GETKEY #\$7F CHAR #0 MDDE #MDPASI 0 PREADRET2 # <cno0 * CHAR CHAR</cno0 	;SETUP ZP STUFF ;GET A KEYSTROKE ;DROP HI BIT ;SAVE THE CHAR ;IORESULT= 'GOOD ' ;ARE WE IN 1. 0-MODE? ;=>NOPE ;YES, RETURN CN IN X
CA71: A2 00 CA73: 60 CA74: C CA74: C CA74: C CA74: C CA74: 20 CA77: 20 CA77	CA74 CF 06 04 CABA 06 CABE 06	320234567890112345678912345678	PREAD * * PREADRET2 * PASCAL QU	RTS INCLUI 20T: EQU JSR AND STA LDX LDA LDA LDA LDA LDA LDA LDA LDA LDA LDA	* PSETUP GETKEY #\$7F CHAR MODE #MDPAS1.0 PREADRET2 # <cno0 *="" char="" de="" psetup<="" pwrite="" td=""><td>;SETUP ZP STUFF ;GET A KEYSTROKE ;DROP HI BIT ;DAVE THE CHAR ;IORESULT='GODD' ;ARE WE IN 1.0-MODE? ;YES, RETURN CN IN X ;RESTORE CHAR ;SAVE CHARACTER ;SETUP ZP STUFF</td></cno0>	;SETUP ZP STUFF ;GET A KEYSTROKE ;DROP HI BIT ;DAVE THE CHAR ;IORESULT='GODD' ;ARE WE IN 1.0-MODE? ;YES, RETURN CN IN X ;RESTORE CHAR ;SAVE CHARACTER ;SETUP ZP STUFF
CA71: A2 00 CA73: A2 CA74: C CA74: C CA74: C CA74: C CA74: C CA77: C CA77: C CA77: C CA77: C CA77: C CA77: C CA77: C CA77: A7 CA77: A7 CA77: A7 CA77: A7 CA77: A7 CA77: A7 CA77: A7 CA76: A7 CA84: C CA84: C C CA84: C C C CA84: C C C CA84: C C C C C C C C C C C C C C C C C C C	CA74 CF 06 04 CABA 06 CABA 06 CABE 06 CF CF	3202345678901123456789	PREAD * * * PREADRET2 * PASCAL OU PWRITE	RTS INCLUI PUT: EQU JSR STA LDX LDX LDX LDX LDX LDX LDX LDX EQU LDX EQU LDX STA STA STA STA STA STA STA STA STA STA	* PSETUP GETKEY #\$76 MODE #M PASI 0 PREADRET2 #CHAR CHAR PE PWRITE * CHAR PSETUP INVERT	; SETUP ZP STUFF ; GET A KEYSTROKE ; DROP HI BIT ; DAVE THE CHAR ; IORESULT= 'GODD' ; ARE WE IN 1. 0-MODE? ; =>NOPE ; YES; RETURN CN IN X ; RESTORE CHAR ; SAVE CHARACTER ; SAVE CHARACTER ; SETUP ZP STUFF ; TURN CURSOR DFF
CA71: A2 00 CA73: A2 CA74: C CA74: C CA74: C CA74: C CA74: C CA74: C CA74: C CA74: C CA77: C CA77: C CA77: C CA77: C CA77: C CA77: A2 00 CA81: AD FB CA77: A2 00 CA84: C CA84: C C CA84: C C C CA84: C C C C C C C C C C C C C C C C C C C	CA74 CF 06 04 CABA 06 CABA 06 CABE 06 CF CF	32023456789011234567890	PREAD * * * PREADRET2 * PASCAL OU PWRITE	RTS INCLUI PUT: EQU JSR AND STA LDX LDA LDA LDA EQU LDA RTS INCLUI STA JSR LDA	* PSETUP GETKEY #\$7F CHAR #0 MODE ##7F CHAR CHAR CHAR CHAR CHAR PSETUP INVERT MODE	;SETUP ZP STUFF ;GET A KEYSTROKE ;DROP HI BIT ;DAVE THE CHAR ;IORESULT='GODD' ;ARE WE IN 1.0-MODE? ;YES, RETURN CN IN X ;RESTORE CHAR ;SAVE CHARACTER ;SETUP ZP STUFF
CA71: A2 00 CA73: 60 CA74: C CA74: C CA74: C CA74: C CA74: C CA74: C CA77: C CA77: C CA77: C CA77: C CA77: C CA77: C CA77: 20 15 CA77: 27 F CA70: 8D 78 CA77: A2 77 CA70: C CA84: 4D 78 CA84: C CA84: C CA84: C CA84: C CA84: C CA84: C CA84: C CA84: C CA88: C C CA88: C CA88: C C CA88: C CA88: C CA88: C CA88: C CA88: C CA88: C C CA88: C C C C CA88: C C C C C C C C C C C C C C C C C C C	CA74 CF 06 04 CABA 06 CABA 06 CABE 06 CF CF	31023456789011234567890112	PREAD * * * PREADRET2 * PASCAL OU PWRITE	RTS INCLUI PUT: EQU JSR STA LDX LDX LDX LDX LDX LDX LDX LDX EQU LDX EQU LDX STA STA STA STA STA STA STA STA STA STA	* PSETUP GETKEY ##7F CHAR #0 MDDE ##ASI 0 #M PASI 0 PREADRET2 #CNO0 * CHAR DE PWRITE CHAR PSETUP INVERT MDDE INVERT MDDE	; SETUP ZP STUFF ; GET A KEYSTROKE ; DROP HI BIT ; DAVE THE CHAR ; IORESULT= 'GODD' ; ARE WE IN 1. 0-MODE? ; =>NOPE ; YES; RETURN CN IN X ; RESTORE CHAR ; SAVE CHARACTER ; SAVE CHARACTER ; SETUP ZP STUFF ; TURN CURSOR DFF
CA71: A2 00 CA73: A2 CA74: C CA74: C CA74: C CA74: C CA77: A2 CA78: A2 CA88: C CA88: C C CA88: C CA88: C C CA88: C C CA88: C C CA88: C C CA88: C C C C C C C C C C C C C C C C C C C	CA74 CF 06 04 CABA 06 CABA 06 CABE 06 CF CF 04	31023456789011234567890112345678901123	PREAD * * PREADRET2 * PASCAL OU PWRITE * *	RTS INCLUI UT: EQU JSR JSR JSR LDA LDA LDA LDA LDA LDA LDA LDA RTS LDA LDA LDA STA JSR JSR JSR JSR JSR LDA AND BEG	PE PREAD * PSETUP GETKEY ##77 CHAR #0 PREADRET2 #CONO * CHAR DE PWRITE CHAR PSETUP INVERT MODE INVERT MODE WM GDXY PWRITE3	;SETUP ZP STUFF ;GET A KEYSTROKE ;DROP HI BIT ;SAVE THE CHAR ;IORESULT= 'GODD' ;ARE WE IN 1. 0-MODE? ;YES, RETURN CN IN X ;RESTORE CHAR ;SAVE CHARACTER ;SAVE CHARACTER ;SETUP ZP STUFF ;TURN CURSOR OFF ;ARE WE DOING GOTOXY?
CA71: A2 00 CA73: 60 CA74: C CA74: C CA74: C CA74: C CA74: C CA74: C CA77: C CA78: C CA84: C CA88: C C CA88: C C C CA88: C C C C C C C C C C C C C C C C C C C	CA74 CF 06 04 CABA 06 CABE 06 CF CF CF 04	$\begin{array}{c} 32\\ 0\\ 2\\ 3\\ 4\\ 5\\ 6\\ 7\\ 8\\ 9\\ 0\\ 1\\ 1\\ 1\\ 2\\ 3\\ 4\\ 5\\ 6\\ 7\\ 8\\ 9\\ 0\\ 1\\ 1\\ 1\\ 1\\ 1\\ 1\\ 1\\ 1\\ 1\\ 1\\ 1\\ 1\\ 1\\$	PREAD * * PREADRET2 * PASCAL OU PWRITE * * * HANDLE G	RTS INCLUI UT: EQU JSR JSR JSR LDA LDA LDA LDA LDA LDA LDA LDA RTS LDA LDA LDA STA JSR JSR JSR JSR JSR LDA AND BEG	PE PREAD * PSETUP GETKEY ##77 CHAR #0 PREADRET2 #CONO * CHAR DE PWRITE CHAR PSETUP INVERT MODE INVERT MODE WM GDXY PWRITE3	;SETUP ZP STUFF ;GET A KEYSTROKE ;DROP HI BIT ;SAVE THE CHAR ;IORESULT= 'GODD' ;ARE WE IN 1. 0-MODE? ;YES, RETURN CN IN X ;RESTORE CHAR ;SAVE CHARACTER ;SAVE CHARACTER ;SETUP ZP STUFF ;TURN CURSOR OFF ;ARE WE DOING GOTOXY?
CA71: A2 00 CA73: A2 CA74: C CA74: C CA74: C CA74: C CA77: A2 CA78: A2 CA88: C CA88: C C CA88: C CA88: C C CA88: C C CA88: C C CA88: C C CA88: C C C C C C C C C C C C C C C C C C C	CA74 CF 06 04 CABA 06 CABE 06 CF CF CF 04	3202345678901123456789011234567890112345	PREAD * * PREADRET2 * PASCAL OU PWRITE * *	RTS INCLUI UT: EQU JSR JSR JSR LDA LDA LDA LDA LDA LDA LDA LDA RTS LDA LDA LDA STA JSR JSR JSR JSR JSR LDA AND BEG	PE PREAD PSETUP GETKEY #\$7F CHAR #00 #M.PASI.0 PREADRET2 # <cnoo * CHAR PSETUP INVERT MODE INVERT MODE #M.DESY PKEIDES STUFF: *</cnoo 	;SETUP ZP STUFF ;GET A KEYSTROKE ;DROP HI BIT ;SAVE THE CHAR ;IDRESULT='GODD' ;ARE WE IN 1.0-MODE? ;=>NOPE ;YES, RETURN CN IN X ;RESTORE CHAR ;SAVE CHARACTER ;SAVE CHARACTER ;SAVE CHARACTER ;SAVE CHARACTER ;SAVE CHARACTER ;SAVE CHARACTER ;STURN CURSOR OFF ;TURN CURSOR OFF ;TURN CURSOR OFF ;ARE WE DDING GOTOXY? ;=>NO, PRINT IT
CA71: A2 00 CA73: A2 CA74: C CA74: C CA74: C CA74: C CA74: C CA77: C CA77: C CA77: C CA77: C CA77: C CA77: A2 CA77: A2 CA77: A2 CA77: A2 CA77: A2 CA77: A2 CA81: A2 CA84: C CA84: C CA84: C CA84: C CA88: C C CA88: C CA88: C CA88: C CA88: C CA88: C CA88: C CA88: C	CA74 CF 06 04 CA8A 06 CA8A 06 CA8E 06 CF CE 04 CACB	310234567890112345678901123456789011234567	PREAD * * PREADRET2 * PASCAL OU PWRITE * * HANDLE G(*	RTS DUT: EQU JSR JSR JSR AND BEG LDX EQU LDA EQU LDA RTS U LDA STA JSR TOXY EQU EQU EQU EQU LDA EQU LDA	PSETUP PSETUP GETKEY #\$7F CHAR #0 MODE #M PASI 0 PREADRET2 # <cno0 * CHAR PSETUP INVERT MODE #M GOXY PWRITE3 STUFF: * XCCORD</cno0 	;SETUP ZP STUFF ;GET A KEYSTROKE ;DROP HI BIT ;SAVE THE CHAR ;IGRESULT='GODD' ;ARE WE IN 1.0-MODE? ;YES, RETURN CN IN X ;RESTORE CHAR ;SAVE CHARACTER ;SETUP ZP STUFF ;TURN CURSOR OFF ;ARE WE DOING GOTOXY? ;=>NO, PRINT IT
CA71: A2 00 CA73: 60 CA74: 0 CA74: 20 CA74: 20 CA74: 20 CA77: 20 CA78: 20 CA88: 20 C	СА74 СF 06 04 САВА 06 06 САВА 06 САВЕ 06 СF CE 04 САСВ	3102345678901123456789011234567890112345678	PREAD * * PREADRET2 * PASCAL OU PWRITE * * HANDLE G(*	RTS INCLUD JUT: EQU JSR AND STA AND STA AND BE0 LDX EQU LDA AND EQU INCLUD FUT: EQU STA JSR LDX STA AND EQU LDA STA AND STA STA STA STA STA STA STA STA STA STA	PE FREAD PSETUP GETKEY ##77 CHAR #00 #MD2 #MD2 #CHAR PREADRET2 # <cnoo * CHAR PSETUP INVERT MODE INVERT MODE PWRITE3 STUFF: * XCOORD GETY</cnoo 	;SETUP ZP STUFF ;GET A KEYSTROKE ;DROP HI BIT ;SAVE THE CHAR ;IDRESULT='GODD' ;ARE WE IN 1.0-MODE? ;=>NOPE ;YES, RETURN CN IN X ;RESTORE CHAR ;SAVE CHARACTER ;SAVE CHARACTER ;SAVE CHARACTER ;SAVE CHARACTER ;SAVE CHARACTER ;SAVE CHARACTER ;STURN CURSOR OFF ;TURN CURSOR OFF ;TURN CURSOR OFF ;ARE WE DDING GOTOXY? ;=>NO, PRINT IT
CA71: A2 00 CA73: A2 CA74: C CA74: C CA74: C CA74: C CA74: C CA74: 20 CA74: 20 CA77: 20 CA77: 20 CA77: 20 CA77: 20 CA77: 20 CA77: 20 CA77: 20 CA78: 20 CA81: AD CA81: AD CA88: C CA88: AD CA88: AD CA88: C CA88: C C CA88: C C C	СА74 СF 06 04 САВА 06 06 САВА 06 САВЕ 06 СF CE 04 САСВ	310234567890112345678901123456789011234567	PREAD * * PREADRET2 * PASCAL OU PWRITE * * HANDLE G(*	RTS DUT: EQU JSR JSR JSR AND BEG LDX EQU LDA EQU LDA RTS U LDA STA JSR TOXY EQU EQU EQU EQU LDA EQU LDA	PSETUP PSETUP GETKEY #\$7F CHAR #0 MODE #M PASI 0 PREADRET2 # <cno0 * CHAR PSETUP INVERT MODE #M GOXY PWRITE3 STUFF: * XCCORD</cno0 	;SETUP ZP STUFF ;GET A KEYSTROKE ;DROP HI BIT ;SAVE THE CHAR ;IGRESULT='GODD' ;ARE WE IN 1.0-MODE? ;YES, RETURN CN IN X ;RESTORE CHAR ;SAVE CHARACTER ;SETUP ZP STUFF ;TURN CURSOR OFF ;ARE WE DOING GOTOXY? ;=>NO, PRINT IT
CA71: A2 00 CA73: A2 CA74: C CA74: C CA74: C CA74: C CA74: C CA74: 20 CA74: 20 CA77: 20 CA77: 20 CA77: 20 CA77: 20 CA77: 42 CA77: 42 CA77: 42 CA77: 42 CA77: 42 CA77: 42 CA78: 42 CA88: A2 CA88: A2 CA88: A2 CA88: A2 CA88: A2 CA88: A2 CA88: C CA88: C C CA88: C CA88: C C CA88: C C C CA88: C C C C C C C C C C C C C C C C C C C	CA74 CF 06 04 CA8A 05 CA8A 05 CA8E 05 CA8E 04 CA6E 04 CA6E 05 CA7E 05 CA4F	3123456789011234567891123456789012345678901	PREAD * * PREADRET2 * PASCAL OU PWRITE * * HANDLE G(*	RTS INCLUD 2011: 2012: 2013 2014 2015 2015 2015 2015 2015 2015 2015 2015	* PSETUP GETKEY #\$7F CHAR #0 MODE #M.PASI.0 PREADRET2 # <char PSETUP INVERT MODE #MDE ** CHAR PSETUP INVERT MODE ** ** XCOORD GETY CHAR ** ** ** ** ** ** ** ** ** *</char 	;SETUP ZP STUFF ;GET A KEYSTROKE ;DROP HI BIT ;SAVE THE CHAR ;IGRESULT='GODD' ;ARE WE IN 1.0-MODE? ;YES, RETURN CN IN X ;RESTORE CHAR ;SAVE CHARACTER ;SETUP ZP STUFF ;TURN CURSOR OFF ;ARE WE DOING GOTOXY? ;=>NO, PRINT IT
CA71: A2 00 CA73: A2 CA74: C CA74: C CA74: C CA74: C CA74: C CA77: C CA78: C CA88: C C	CA74 CF 06 04 CA8A 06 CA8A 06 CA8A 06 CA8E 06 CF CF CF CF CA2E 04 CA2E 06 CA2E	3102345678901123456789112345678901234567890112345678901222	PREAD * * PREADRET2 * PASCAL OU PWRITE * * HANDLE G(*	RTS INCLUD PUT: EQU JSR AND STA LDX AND LDX AND LDX LDX LDX LDX EQU LDA RTS EQU LDA RTS STA LDX RTS STA LDX EQU LDA STA AND EQU EQU LDA STA LDX EQU LDA STA LDX EQU LDA STA STA STA STA EQU LDA STA STA STA STA STA STA STA STA STA ST	PE FREAD PSETUP GETKEY #\$7F CHAR #0 PREADRET2 # <cnoo * CHAR PSETUP INVERT MODE WM ITE STUFF: * XCOORD GETY CHAR #32 XCOORD</cnoo 	<pre>;SETUP ZP STUFF ;GET A KEYSTROKE ;DROP HI BIT ;SAVE THE CHAR ;IGRESULT= 'GODO' ;ARE WE N 1.0-MODE? ;FOPE ;YES, RETURN CN IN X ;RESTORE CHAR ;SAVE CHARACTER ;SAVE CHARACTER ;SETUP ZP STUFF ;SETUP ZP STUFF ;ARE WE DOING GOTOXY? ;=>NO, PRINT IT ;ARE WE WAITING FOR X? ;=>NO, THIS IS Y ;MAKE BINARY</pre>
CA71: A2 00 CA73: A2 CA74: C CA74: C CA74: C CA74: C CA74: C CA74: 20 CA74: 20 CA77: 20 CA77: 20 CA77: 20 CA77: 20 CA77: 42 CA77: 42 CA77: 42 CA77: 42 CA77: 42 CA77: 42 CA78: 42 CA88: A2 CA88: A2 CA88: A2 CA88: A2 CA88: A2 CA88: A2 CA88: C CA88: C C CA88: C CA88: C C CA88: C C C CA88: C C C C C C C C C C C C C C C C C C C	CA74 CF 06 04 CA8A 06 CA8A 06 CA8A 06 CA8E 06 CF CF CF CF CA2E 04 CA2E 06 CA2E	3123456789011234567891123456789012345678901	<pre>PREAD * * * PREADRET2 * PASCAL OU PNRITE * * * * HANDLE GI * PWRITE2</pre>	RTS INCLUD 2011: 2012: 2013 2014 2015 2015 2015 2015 2015 2015 2015 2015	PE FREAD PSETUP GETKEY ##77 CHAR #0 MDDE #MDPASI.0 PREADRET2 # <cnoo * CHAR PSETUP INVERT MODE CHAR PSETUP INVERT MODE STUFF: * XCOORD GETY CHAR #32 XCOORD</cnoo 	<pre>;SETUP ZP STUFF ;GET A KEYSTROKE ;DROP HI BIT ;SAVE THE CHAR ;IGRESULT= 'GODO' ;ARE WE IN 1.0-MODE? ;ARE WE IN 1.0-MODE? ;SAVE CHARACTER ;SAVE CHARACTER ;SAVE CHARACTER ;SAVE CHARACTER ;SETUP ZP STUFF ;ARE WE DOING GOTOXY? ;=>NO, PRINT IT ;ARE WE WAITING FOR X? ;=>NO, THIS IS Y</pre>

CAAF:	25 * NOW DO T	HE GOTO	¥¥-	
CAAF: CAAF	26 * 27 GETY	EQU		
CAAF: AD 7B 06	27 GETY 28	LDA	* CHAR	CONVERT YCOORD
CAB2: 38 CAB3: E9 20	29 30	SEC	#32	
CAB5: 8D FB 05	31	STA	DURCV	
CAB8: 20 51 CB CAB8: AD FB 06	32 33	JSR	BASCALC	COMPUTE BASE ADDRESS
CABE: AD FB 06 CABE: BD 7B 05	34	STA	OURCH	
CAC1: AD FB 04 CAC4: 29 F7	35 36	LDA	MODE #255-M. GOXY	; TURN OFF GOTOXY
CAC6: 8D FB 04 CAC9: D0 44 CBOF	37 38	STA	MODE	
CACE	39 *		FWRITERET	;=>DONE (ALWAYS TAKEN)
CACB: CACB CACB: AD 7B 06	40 PWRITE3 41	EQU	* CHAR	GET CHAR TO PRINT
CACE: CY 1E	42	CMP	#\$1E	IS IT COTOXY?
CADO: FO OA CADC CAD2: C9 20	43 44		51AR1XY	;=>YES ;IS IT DTHER CTL?
CAD4: B0 15 CAEB CAD6: 20 99 CB	45 46	BCS	PWRITE4	;=>ND; PRINT IT
CAD9: 4C OF CB	47	JMP	PWRITERET	;EXECUTE IT IF POSSIBLE ;=>EXECUTED OR IGNORED
CADC: CADC:	48 * 49 * START TH	F GOTOX	Y SEQUENCE	
CADC	50 *			
CADC AD FB 04	52	EQU	* MODE	TURN ON FLAG
CADF:09 OB CAE1:8D FB 04	53 54	ORA	MODE	
CAE4: A9 FF	55	LDA	#-1	SET X NEGATIVE TO
CAE6 8D FB 06 CAE9 30 24 CBOF	56 57	STA	#-1 XCOORD PWRITERET	; SHOW WE NEED IT ;=>EXIT TILL COORDS COME BY (ALWAYS)
CAEB	58 *			
CAEB	59 * JUST A PI 60 *	RINTABLE	E CHARACTER:	
CAEB: CAEB	61 PWRITE4		* #\$80	FORCE TO NORMAL
CAEB: 09 80 CAED: AC 78 05	63	LDY	#\$80 OURCH STORCHAR	GET CH
CAFO: 20 F2 CE CAF3:	64 65 *	JSR	STORCHAR	STUFF IT!
CAF3:	66 * BUMP CUR	SOR HOR	IZONTAL	
CAF3: CAF3:EE 7B 05	67 * 68	INC	DURCH	;BUMP IT ;ARE WE PAST THE ; END OF THE LINE? ;=>NO, NO PROBLEM
CAF6: AD 7B 05 CAF9: C5 21	69 70	LDA	OURCH	ARE WE PAST THE
CAFB: 90 12 CBOF	71	BCC	PWRITERET	; =>NO, NO PROBLEM
CAFD: CAFD:	72 * 73 * IF IN TR/	ANSPAREN	T MODE, DON	Ϋ́Τ
CAFD: CAFD:	74 * WRAPARO	JND THE	RIGHT EDGE.	
CAFD: AD FB 04	76	LDA	MODE	GET MODE
CB00:29 01 CB02:F0 05 CB09	77 78	AND	#M. TRANS	;WELL??? ;≥NOT TRANSPARENT ;PIN AT RIGHT EDGE
CB04: CE 7B 05 CB07: D0 06 CB0F	79	DEC	OURCH	PIN AT RIGHT EDGE ; (ALWAYS TAKEN)
CB09:	81 *		PWRITERET	(ALWAYS TAKEN)
CB09: CB09 CB09:20 EC CB	82 PWWRAP 83	JSR	* X.CR	YES, DO C/R
CBOC: 20 91 CC	84 85 *		X.LF	; AND L/F
CBOF: CBOF: CBOF			*	
CBOF: 20 DD CE CB12: A2 00	87 88	JSR	INVERT #0	; TURN CURSOR ON ; IORESULT='GOOD'
CB14:60	89	RTS		10KE30C1- 000D
CB15: CB15:	12 2		DE SUBS1	
CB15: CB15:	3 * NAME 4 * FUNCTION	GETKEY	KEVETBOKE	
CB15:	5 * INPUT 6 * OUTPUT	NONE	RETSTRUKE	
CB15: CB15:	6 * OUTPUT 7 * VOLATILE:	AC=KEY	CODE	
CB15:	8			
CB15: CB15: CB15	9 * 10 GETKEY	EQU	*	
CB15:E6 4E CB17:D0 02 CB1B	11	INC	RNDL GE1K2	BUMP RANDOM SEED
CB19: E6 4F	13	INC	RNDH	
CB1B: AD OO CO	14 GETK2 15	EQU	* KBD	KEYPRESS?
CB1E:10 F5 CB15 CB20:8D 10 CO	16 17	BPL	* KBD GETKEY KBDSTRB	;=>NOPE ;CLEAR STROBE
CB23: 60	18	N LO		
CB24 CB24		TESTCA		
CB24: CB24:	20 * NAME 21 * FUNCTION	SEE IF	BOCOL CARD	PLUGGED IN
CB24	22 * INPUT 23 * OUTPUT 24 *	'BEG'	IF CARD AVAL	LABLE
CB24: CB24:	24 * : 25 * VOLATILE:	AC. Y	IF NO1	
CB24 CB24	26			
GBE4	c/ #			

CB24: CB24	28 TESTCARD	EQU *	
CB24: AD 1C CO	29 IESICARD	LDA RDPAGE2	REMEMBER CURRENT VIDEO DISPLAY
CB27: 0A	30	ASI. A	IN THE CARRY
CB28: A9 88	31	LDA #\$88	USEFUL CHAR FOR TESTING
CB2A:2C 18 CO CB2D:8D 01 CO	32 33	BIT RDBOCOL STA SETBOCOL	;REMEMBER VIDEO MODE IN 'N' ;ENABLE BOCOL STORE
CB30: 08	34	PHP	LOCK INTERRUPTS WHILE
CB31:78	35	SEI	SCREENHOLES ARE WRONG
CE32: 08	36	PHP	;SAVE 'N' AND 'C' FLAGS
CB33:8D 55 CO	37	STA TXTPAGE2	SET PAGE2
CB36: AC 00 04 CB39: BD 00 04	38 37	LDY \$0400 STA \$0400	GET FIRST CHAR
CE3C: AD 00 04	40	LDA \$0400	GET IT BACK FROM RAM
CB3F: 8C 00 04	41	STY \$0400	RESTORE ORIG CHAR
CB42: 28	42	PLP	RESTORE IN' AND IC' FLAGS
CB43: BO 03 CB48 CB45: 8D 54 CO	43 44	BCS STAY2 STA TXTPAGE1	;STAY IN PAGE2 ;RESTORE PAGE1
CB48: CB48	45 STAY2	EQU *	RESIDE PAGEI
CB48: 30 03 CB4D	46	BMI STAYBO	;=>STAY IN BOCOL MODE
CB4A: 8D 00 CO	47	STA CLRBOCOL	; TURN OFF BOCOL STORE
CB4D: CB4D CB4D: 28	48 STAY80 49	EQU * PLP	ALLOW IRQ AGAIN
CB4E: CB4E	50 TESTFAIL	EQU *	ALLOW ING AGAIN
CB4E: C9 88	51	CMP #\$88	; WAS CHAR VALID?
CB50: 60	52	RTS	RETURN RESULT AS BEQ/BNE
CB51: CB51:	53 54 * NAME :	BASCALC, BASCALCZ	
CB51:	55 * FUNCTION.	CALC BASE ADDR FO	R SCREEN LINE
CB51:	56 * INPUT :	DURCY (BASCALC)	
CB51:	57 * :	AC=CV (BASCALCZ)	
CB51:	58 * OUTPUT :	BASL/BASH	
CB51: CB51:	59 * VOLATILE: 60 * CALLS :	NOTHING	
CB51	61	5N1FF1R0	
CB51:	62 *		
CB51: FC75	63 SNIFFIRG	EQU \$FC75	
CB51: CB51: CB51	64 * 65 BASCALC	EQU *	RIPPED OFF FROM 58 ROM
CB51: 18	66	CLC *	SHOW ENTRY POINT
CB52: 90 01 CB55	67	BCC BSCLC1	
CB54: CB54	68 BASCALCZ	EQU *	
CB54:38 CB55: CB55	69 70 BSCLC1	SEC EQU *	SHOW ENTRY POINT
CB55: 48	71	PHA *	SAVE AC
CB56: BO 03 CB5B	72	BCS BSCLCIA	;=>CV ALREADY IN AC
CB58: AD FB 05	73	LDA OURCV	
CB5B: CB5B	74 BSCLC1A	EQU *	
CB5B: 48 CB5C: 4A	75 76	PHA LSR A	
CB5D: 29 03	77	AND #\$03	
CB5F: 09 04	78	ORA #\$04	
CB61:85 29	79	STA BASH	
CB43:80 FB 07 CB44:68	80 81	STA OLDBASH PLA	SAVE FOR F/W PROTOCOL
CB67:29 18	82	AND #\$18	
CB69:90 02 CB6D	83	BCC BSCLC2	
CB6B: 69 7F	84	ADC #\$7F	
CB6D: 85 28 CB6F: 0A	85 BSCLC2 86	STA BASL ASL A	
CB70: OA	87	ASL A	
CB71:05 28	88	ORA BASL	
CB73:85 28	87	STA BASL	
CB75: CB75:	90 * 91 * HANDLE TH	E SCROLLING WINDOW	le l
CB75:	92 *	C SCROLLING WINDOW	•
CB75: A5 20	93	LDA WNDLFT	
CB77:08	94	PHP	PRESERVE CARRY
CB78:2C 1F CO CB7B:10 01 CB7E	95 96	BIT RDBOVID BPL BASCLC3	;WHICH MODE? ;=>40. NO DIVIDE.
CB7D: 4A	97	LSR A	DIVIDE BY 2 FOR SOCOL WINDOW
CB7E: CB7E	98 BASCLC3	EQU *	
CB7E: 28	99	PLP	RESTORE CARRY
CB7F: 65 28 CB81: 85 28	100 101	ADC BASL STA BASL	ADJUST BASE FOR WNDLFT
CB83: 8D 7B 07	102	STA OLDBASL	SAVE FOR F/W PROTOCOL
CB86:	103 *		
CB86:	104 * SNIFF FOR	IRG IF NECESSARY:	
CB86:	105 * 106	LDA MODE	
CB86: AD FB 04 CB89: 29 01	107	AND #M. IRQ	
CB8B: FO OA CB97	108	BEQ BASCLCX	;=>IRQ DISABLED, RETURN
CBBD: AD FB 04	109	LDA MODE	; IS BASIC RUNNING?
CB90:29 20 CB92:D0 03 CB97	110 111	AND #M. PASCAL BNE BASCLCX	=>DON'T SNIFF UNDER PASCAL
CB92: D0 03 CB97 CB94: 20 75 FC	112	JSR SNIFFIRG	GO DO IT
CB97: CB97	113 BASCLCX	EGU *	
CB97: 68	114	PLA	RESTORE AC
CB98: 60 CB99:	115	RTS	
CB99:	117 * NAME :	CTLCHAR	
CB99:	118 * FUNCTION:	EXECUTE CTL CHAR	
CB99:	119 * INPUT :	AC=CHAR	

CB99:		120 * (DUTPUT :	'BCS'	IF NOT CTL	
CB99: CB99:		121 *	: VOLATILE:	'BCC' NOTHIN	IF CTL EXECU	JTED
CB99:		123 * (CALLS :	MANY	THINGS	
CB99: CB99:		124				
CB99:	CB99	126 CTI	LCHAR	EQU	*	
CB99:80 78 CB9C:48	04	127		STA PHA	TEMP1	; TEMP SAVE OF CHAR ; SAVE AC
CB9C: 48 CB9D: 98		128 129		TYA		SAVE Y
CB9E: 48		130		PHA		
CB9F: CB9F:AC 78	04	131 *		LDY	TEMP 1	GET CHAR IN QUESTION
CBA2: CO 07	04	133		CPY	#\$07	IS.IT NUL. ACK? =>YES, NOT USED IS IT CTL?
CBA4: 90 05		134		BCC	CTLCHARX	=>YES, NOT USED
CBA6: B9 71 CBA9: D0 03	CBAE	135 136		LDA BNE	CTLADH-7,Y CTLGO	;=>YES
CBAB:	CBAB	137 CT	LCHARX	EQU	×	
CBAB: 38 CBAC: BO 04	CBB2	138		SEC BCS	CTLRET	SAY 'NOT CTL'
CBAE:		140 *			STERE!	
CBAE:	CBAE	141 CT	LGO	EQU	* CTLXFER	EXECUTE SUBROUTINE
CBAE: 20 B6 CBB1:	CB	142 *		Jak	UTEAPER	
CBB1:18		144		CLC		SAY 'CTL CHAR EXECUTED'
CBB2: CBB2:68	CBB2	145 CTI 146	LRET	EQU PLA	*	RESTORE
CBB3: A8		147		TAY		; Y
CBB4: 68 CBB5: 60		148 149		PLA RTS		; AND AC
CBB5: 80		150 *				
CBB6:	CBB6	151 CT	LXFER	EQU	*	DUCH ONTO STACK FOR
CBB6:48 CBB7:89 58	cc	152			CTLADL-7, Y	; PUSH ONTO STACK FOR ; TRANSFER TRICK
CBBA: 48	00	154		PHA		
CBBB: 60 CBBC:		155 156 *		RTS		XFER TO ROUTINE
CBBC:			EXECUTE E	BELL		
CBBC:		158 *				
CBBC: A9 40	CBRC	159 X.1 160	BELL	EQU LDA	* #\$40	RIPPED OFF FROM MONITOR
CBBC: A9 40 CBBE: 20 CF	CB	161		JSR	WAIT	
CBC1: AO CO		162 163 BE		LDY	#\$CO #\$OC	
CBC3: A9 0C CBC5: 20 CF	CB	163 50	L.L.2	JSR	WAIT	
CBCB: AD 30	co	165		LDA	SPKR	
CBCB:88 CBCC:D0 F5	свсз	166 167		DEY BNE	BELL2	
CBCE: 60	0000	168		RTS		
CBCF: CBCF:	CBCF	169 * 170 WA	· -	EQU		RIPPED OFF FROM MONITOR ROM
CBCF: 38	CBCF	170 WA	11	SEC		SKIPPED OF TROPTION ROLL
CBDO: 48		172 WA	112	PHA	#1	
CBD1:E9 01 CBD3:D0 FC	CBD1	173 WA 174	113	SBC	#1 WAIT3	
CBD5: 68		175		PLA		
CBD6: E9 01 CBD8: D0 F6	CBDO	176 177		SBC BNE	#1 WAIT2	
CBDA: 60	CBDO	178		RTS	WHITE	
CBDB:		179 *	EXECUTE E			
CBDB: CBDB:		180 * 1	EXECUTE	BACKSPA	JE:	
CBDB:	CBDB	182 X.	BS	EQU	*	
CBDB:CE 7B CBDE:10 OB	O5 CBEB	183 184		DEC BPL	OURCH BSDONE	;BACK UP CH ;=>DONE
CBEO: A5 21		185		LDA	WNDWDTH	BACK UP TO PRIOR LINE
CBE2: CBE2:8D 7B	CBE2	186 BS 187	40	EQU	* OURCH	SET CH
CBE5: CE 7B CBE5: CE 7B CBE8: 20 34	05	18/		DEC	OURCH	
CBE8: 20 34	CC	189		JSR	X. US	NOW DO REV LINEFEED
CBEB: CBEB: 60	CBEB	190 BS 191	DUNE	EQU RTS	*	
CBEC:		192 *		10.0 0.000		
CBEC: CBEC:		193 * 194 *	EXECUTE (CARRIAG	E RETURN:	
CBEC:	CBEC	195 X.	CR	EQU	*	
CBEC: AD FB	04	196		LDA	MODE	WHICH LANGUAGE?
CBEC: AD FB CBEF: 29 20 CBF1: DO OA	CBFD	197 198		AND	#M. PASCAL X. CRPAS	=>PASCAL, NO CLR EOL
CBF3: AD FB	04	199		LDA	MODE	INPUT OR DUTPUT?
CBF6: 29 40 CBF8: F0 03	CBFD	200 201		AND	#M. BINPUT X. CRPAS	;=>DUTPUT. NO CLEARING
CBFA: 20 48	co	202		JSR	X. GS	CLEAR TO EOL
CBFD: CBFD:		203 *	CRRAC	EQU	*	
CBFD: CBFD: A9 00		204 X. 205	URPAS	LDA	* #0	; BACK UP CH TO ; BEGINNING OF LINE
CBFF: 8D 7B	05	206		STA	DURCH	BEGINNING OF LINE
CC02: AD FB CC05: 29 20		207		LDA	MODE #M. PASCAL	; ARE WE IN BASIC?
CC05: 29 20 CC07: D0 03	ccoc	209		BNE	X. CRRET	=>PASCAL, AVOID AUTO L/F
CC09:20 91 CC0C:	ÇC	210 211 X.	CRRET	JSR	X.LF *	EXECUTE AUTO LF FOR BASIC
0000.	0000	A.	SINCE		535	

CCOC: 60	212	RTS		
CCOD: 0000	213	DO	0	IND MORE ROM SPACE!
S	214 * 215 *	EVECU	TE SYNC:	
S	216 *	EXECU	IE STNC.	
S	217 X. SYN	EQU	*	
S	218 219	LDA BPL	RDVBLBAR X. SYN	;WAIT FOR VBL ;=>WAIT FOR VIDEO SCAN
S	220 X. SYN2	LDA	RDVBLBAR	NOW WAIT FOR
S	221	BMI	X. SYN2	; BLANKING TO BEGIN
S	222	RTS		
CCOD:	224 *	L 114		
CCOD:	225 * EXECUTE	HOME:		
CCOD: CCOD: CCOD	226 * 227 X.EM	EQU		
CCOD: A5 22	228	LDA	WNDTOP	
CCOF: BD FB 05	229	STA	OURCV	STUFF CV
CC12:A9 00 CC14:8D 7B 05	230 231	LDA STA	#O OURCH	STUFF CH
CC17:4C 51 CB	232	JMP	BASCALC	RETURN VIA BASCALC (UGH!)
CC1A:	233 *			
CC1A: CC1A:	234 * EXECUTE 235 *	CLEAR L	INE:	
CC1A: CC1A	236 X. SUB	EQU	*	
CC1A: A4 21	237	LDY	WNDWDTH	
CC1C:88 CC1D: CC1D	238 237 X.SUB80	DEY EQU	*	
CC1D: A9 A0	240	LDA	* #′	' ; BLANKIE BLANK
CC1F: CC1F	241 X.SUBLP 242	EQU	*	
CC1F:20 F2 CE CC22:88	242	JSR DEY	STORCHAR	STUFF THE BLANK
CC23: 10 FA CC1F	244	BPL.	X. SUBLP	;=>CLEAR THE LINE
CC25: 60 CC26:	245 246 *	RTS		
CC26:	246 * 247 * EXECUTE	EDRWARD	SPACE	
CC26:	248 *			
CC26: CC26	249 X.FS	EQU	*	DUMP CU
CC26:EE 7B 05 CC29:AD 7B 05	250 251	LDA	BURCH BURCH	;BUMP CH ;GET THE POSITION
CC2C: C5 21	252	CMP	WNDWDTH	OFF THE RIGHT SIDE?
CC2E: 90 03 CC33 CC30: 20 EC CB	253 254	BCC	X.FSRET X.CR	;≃>NO, GODD ;=>YES, WRAP AROUND
CC33:	255 *	Jan	X. CR	-STES, WRAF AROUND
CC33: CC33	256 X.FSRET	EQU	*	
CC33: 60 CC34:	257 258 *	RTS		
CC34:	259 * EXECUTE	REVERSE	LINEFEED:	
CC34:	260 *			
CC34: CC34 CC34:CE FB 05	261 X.US 262	EQU DEC	* OURCV	BACK UP CV
CC37: 30 07 CC40	263	BMI	X. US1	;=>OFF TOP OF SCREEN
CC39:AD FB 05 CC3C:C5 22	264 265	CMP	OURCV WNDTOP	OFF TOP OF WINDOW?
CC3E: BO 05 CC45	266	BCS	X. US2	;=>NO, STILL IN WINDOW
CC40:	267 *			
CC40: CC40:	268 * PIN CV T 269 *	U WINDO	W TOP:	
CC40: CC40	270 X. US1	EQU	*	
CC40:EE FB 05 CC43:F0 03 CC48	271 272	INC	OURCV X. USRET	PUT BACK WHERE IT WAS
CC45: CC45	273 X.US2	EQU	X. USREI *	
CC45:20 51 CB	274	JSR	BASCALC	RECOMPUTE BASE ADDR
CC48: CC48 CC48: 60	275 X.USRET 276	EQU	*	
CC49:	277 *			
CC49:	278 * EXECUTE 279 *	"NORMAL	VIDEO"	
CC49: CC49: CC49	279 * 280 X.SO	EQU	*	
CC49: AD FB 04	281	LDA	MODE	SET MODE BIT
CC4C: 29 FB	282	AND	#255-M. VMODE	E SET 'NORMAL'
CC4E: A0 FF CC50: D0 07 CC59	283 284	LDY BNE	#255 STUFFINV	; (ALWAYS)
CC52:	285 *			
CC52: CC52:	286 * EXECUTE 287 *	"INVERS	E VIDEO"	
CC52: CC52	288 X.SI	EQU	*	
CC52: AD FB 04	289	LDA	MODE	SET MODE BIT
CC55:09 04 CC57:A0 7F	290 291	ORA	#M. VMODE #127	;SET 'INVERSE'
CC59: CC59	292 STUFFINV	EQU	#12/ *	
CC59:8D FB 04	293	STA	MODE	SET MODE
CC5C:84 32 CC5E:40	294 295	STY	INVELG	STUFF FLAG TOD
CC5F: CC5F	297 CTLADL	EQU	*	
CC5F: BB	298 299	DFB	>X. BELL-1) BEL
CC60: DA CC61: 00	299	DFB	>X.BS-1 O	; BS ; HT
CC62: 90	301	DFB	>X. LF-1	: LF
CC63: 22 CC64: 41	302 303	DFB	>X. VT-1 >X. FF-1	; VT ; FF
CC65: EB	303	DFB	>X. FF=1 >X. CR=1	; CR

CC46: 48 CC67: 51 CC68: 00 CC69: 58 CC69: 58 CC66: 00 CC60: 00 CC60: 00 CC60: 49 CC66: A9 CC66: A9 CC70: 00 CC71: 00 CC72: 19 CC73: 00 CC74: 25 CC75: 47 CC76: 00 CC77: 33	305 306 307 310 311 312 313 314 313 314 315 315 315 317 317 319 320 321 322 323 *	DFB DFB DFB DFB DFB DFB DFB DFB DFB DFB	0 >X. DC1-1 >X. DC2-1 0 0 >X. NAK-1 >SCROLLDN-1 >SCROLLUP-1 0 >X. EM-1	: SO : SI : DLE : DC2 : DC3 : DC4 : DC4 : DC4 : CAN : ETB : CAN : ETS : ETC : ETS : CAN : ETS : ETC : ETS : CAN : ETS : ETS : CAN : ETS : ETS : CAN : ETS : CAN : ETS : ETS : CAN :
CC78: CC78 CC78: CC78 CC78:CB	323 * 324 CTLADH 325	EQU	* <x. bell-1<="" td=""><td>BEL</td></x.>	BEL
CC79: CB CC7A: 00	326 327	DFB	<x. bs-1<="" td=""><td>i BS i HT</td></x.>	i BS i HT
CC7B: CC	328	DFB	<x. lf-1<="" td=""><td>i LF</td></x.>	i LF
CC7C:CD CC7D:CD	329 330	DFB	<x. vt-1<br=""><x. ff-1<="" td=""><td>i VT i FF</td></x.></x.>	i VT i FF
CC7E: CB	331	DFB	<x. cr-1<="" td=""><td>; CR</td></x.>	; CR
CC7F: CC CC80: CC	332 333	DFB DFB		; SD ; SI
CC81:00 CC82:CD	334 335	DFB	0	; DLE ; DC1
CC83: CD	335	DFB		; DC2
CC84:00 CC85:00	337 338	DFB DFB	0	; DC3 ; DC4
CC86: CD	337	DFB	<x. nak-1<="" td=""><td>NAK</td></x.>	NAK
CC87: CC CC88: CC	340 341	DFB	<scrolldn-1 <scrollup-1< td=""><td>; ETB</td></scrollup-1<></scrolldn-1 	; ETB
CC89: 00 CC8A: CC	342 343	DFB DFB		; CAN ; EM
CCBB: CC	344	DFB	<x. sub-1<="" td=""><td>; SUB</td></x.>	; SUB
CCBC: 00 CCBD: CC	345 346	DFB	0 <x. fs-1<="" td=""><td>; ESC ; FS</td></x.>	; ESC ; FS
CCBE: CD CCBF: 00	347 348	DFB		; GS ; RS
CC90: CC	349	DFB	<x. td="" us-1<=""><td>; US</td></x.>	; US
CC91: CC91:	13 2 *	INCLU	DE SUBS2	
CC91: CC91:	3 * EXECUTE 4 *	LINEFEE	D:	
CC91: CC91	5 X.LF	EQU	*	
CC91:EE FB 05 CC94:AD FB 05	6 7	INC LDA	DURCV DURCV	;BUMP CV ;SEE IF OFF BOTTOM ;OFF THE END?
CC97:C5 23 CC99:B0 03 CC9E	8	CMP BCS	WNDBTM X. LF2	; OFF THE END? ; =>YES
CC9B: 4C 20 CD	10	JMP	X. LFRET	;=>NO, DONE
CC9E: CC9E CC9E: A4 23	11 X.LF2 12	EQU	* WNDBTM	SET TO
CCA0: 88	13	DEY	DURCV	; THE BOTTOM
CCA1: BC FB 05 CCA4:	15 *	STY		, THE BUTTON
CCA4: CCA4:	16 * SCROLL 17 *		EN:	
CCA4: CCA4 CCA4: BA	18 SCROLLUP 19	EQU	*	SAVE X
CCA5: 48	20	PHA		
CCA6: A2 01 CCA8: D0 04 CCAE	21 22	LDX BNE	#1 SCROLL1	; DIRECTION=UP
CCAA: CCAA CCAA: 8A	23 SCROLLDN 24	EQU	*	SAVE X
CCAB: 48	25	PHA		
CCAC: A2 00 CCAE:	26 27 *	LDX	#O	; DIRECTION=DOWN
CCAE: CCAE CCAE: 2C 1F CO	28 SCROLL1 29	EQU BIT	* RD80VID	WHICH MODE?
CCB1: 10 05 CCB8	30	BPL	SCROLL2	;=>40. DO WITH EXISTING WIDTH
CCB3: A5 21 CCB5: 48	31	LDA PHA	WNDWDTH	; TEMPORARILY SAVE ; THE WIDTH AND
CCB6: 46 21 CCB8:	33 34 *	LSR	WNDWDTH	, DIVIDE IT BY 2
CCB8: CCB8	35 SCROLL2	EQU	*	
CCB8: 20 D1 CC CCB8: 2C 1F CO	36 37	JSR BIT	SCRLSUB RD80VID	SCROLL 40 COLS ARE WE IN 80-MODE?
CCBE: 10 51 CD11	38	BPL.	X. SCRLRET	;=>NO, DONE
CCCO: CCCO:		DO THE	OTHER PAGE	
CCCO: CCCO: OB	41 * 42	PHP		ENSURE IRQ INHIBITED
CCC1:78 CCC2:AD 55 CO	43 44	SEI	TXTPAGE2	; WHILE TXTPAGE2 MAPPED IN ;SET PAGE2
CCC5: 20 D1 CC	45	JSR	SCRLSUB	SCROLL PAGE 2
CCCB: AD 54 CO CCCB: 28	46 47	LDA PLP	TXTPAGE1	RESTORE PAGE1

CCCC: 68 CCCD: 85 21	48 49	PLA	WNDWDTH	
CCCF: DO 40 CD11	50	BNE	X. SCRURET	:=>DONE SCROLL80 (ALWAYS TAKEN)
CCD1:	51 *			
CCD1:	52 * 40-COLUM	WIND	OWED SCROLL:	
CCD1: CCD1: CCD1	53 *	=		
CCD1: BC F9 CF	54 SCRLSUB 55	EQU	WNDTAB, X	GET WINDOW TOP/BOT
CCD4: B9 00 00	56	LDA	0, Y	GET WINDOW TOPYBOT
CCD7: EO 01	57	CPX	#1	SCROLLING UP?
CCD9: BO 02 CCDD	58	BCS	MSCRLO	;=>YES, NO PROBLEM
CCDB: E9 00	59	SBC	#O	;-1 IF DOWN (SRC=BTM-1)
CCDD: CCDD	60 MSCRLO	EQU	*	
CCDD: 48	61	PHA		
CCDE: 20 54 CB CCE1: A5 28	62 63 MSCRL1	JSR LDA	BASCALCZ BASL	
CCE3: 85 2A	64	STA	BAS2L	
CCE5: A5 29	65	LDA	BASH	
CCE7: 85 28	66	STA	BAS2H	
CCE9: A4 21	67	LDY	WNDWDTH	
CCEB: BB	68	DEY		
CCEC: 68	69	PLA		
CCED: 18 CCEE: 7D FO CF	70 71	ADC	PLUSMINUS1,	Y . LIG (DOLIN
CCF1: D5 22	72	CMP	WNDTOP, X	AT THE END?
CCF3: FO OD CD02	73	BEG	MSCRLRET	
CCF5: 48	74	PHA		
CCF6:20 54 CB	75	JSR	BASCALCZ	
CCF9: B1 28	76 MSCRL2	LDA	(BASL), Y	
CCFB: 91 2A CCFD: 88	77 78	STA	(BAS2L), Y	
CCFE: 10 F9 CCF9	7B 79	BPL	MSCRL2	
CD00: 30 DF CCE1	80	BMI	MSCRL1	
CDO2:	81 *			
CDO2: CDO2	B2 MSCRLRET	EQU	*	
CD02: E0 00	83	CPX	# 0	SCROLLING DOWN?
CD04: DO OA CD10	84	BNE	MSCRLRTS	; =>NO
CD06:20 54 CB CD09: CD09	85 86 DNEMORE	JSR	BASCALCZ	
CD09: B1 28	87	LDA	(BASL), Y	
CDOB: 91 2A	88	STA	(BAS2L), Y	
CDOD: 88	87	DEY		
CDOE: 10 F9 CD09	90	BPL	ONEMORE	
CD10: CD10 CD10: 60	91 MSCRLRTS 92	EQU	*	
CD10:60		RTS		
CD11.	93 *			
CD11: CD11:	93 * 94 * DONE WITH	THE	SCROLLING JAZ	Ζ:
CD11: CD11:	94 * DONE WITH 95 *		SCROLLING JAZ	2:
CD11: CD11: CD11: CD11	94 * DONE WITH 95 * 96 X.SCRLRET	EQU	*	
CD11: CD11: CD11: CD11 CD11: B4 22	94 * DONE WITH 95 * 96 X.SCRLRET 97	EQU	SCROLLING JAZ * WNDTOP,X	CLEAR TOP OR BOTTOM LINE
CD11: CD11: CD11: CD11: CD11 CD11: B4 22 CD13: BA	94 * DONE WITH 95 * 96 X.SCRLRET 97 98	EQU LDY TXA	* WNDTOP, X	;CLEAR TOP OR BOTTOM LINE ;IF GETTING TOP,
CD11: CD11: CD11: CD11: CD11 CD11: B4 22 CD13: BA CD14: FO 01 CD17	94 * DONE WITH 95 * 96 X.SCRLRET 97 98 99	EQU LDY TXA BEQ	*	CLEAR TOP OR BOTTOM LINE
CD11: CD11: CD11: CD11: CD11 CD11: B4 22 CD13: BA	94 * DONE WITH 95 * 96 X.SCRLRET 97 98	EQU LDY TXA	* WNDTOP, X	;CLEAR TOP OR BOTTOM LINE ;IF GETTING TOP,
CD11: CD11: CD11: CD11: CD11 CD11:B4 22 CD13:B4 CD14:F0 01 CD17 CD16:B8 CD17: CD17 CD17:78	94 * DONE WITH 95 * 96 X. SCRLRET 97 98 99 100 101 X. SCRLRET2 102	EQU LDY TXA BEQ DEY EQU TYA	* WNDTOP,X X.SCRLRET2 *	CLEAR TOP OR BOTTOM LINE FF GETTING TOP, DON'T DECREMENT!
CD11: CD11: CD11: CD11: CD11: CD13: CD13: SA CD14: FO 01 CD17: CD17: CD17: CD17: CD17: CD17: CD17: CD17: CD17: CD17: CD17: CD17: CD17: CD17: CD11: CD12: CD12: CD12: CD12: CD12: CD13: CD14: CD14: CD14: CD14: CD14: CD14: CD14: CD14: CD14: CD17: CD1	94 * DONE WITH 95 * 96 X.SCRLRET 97 98 99 100 101 X.SCRLRET2 102 103	EQU LDY TXA BEQ DEY EQU TYA JSR	* WNDTOP, X X. SCRLRET2	CLEAR TOP OR BOTTOM LINE IF GETTING TOP, DON'T DECREMENT! TEMP CV SETUP COMPUTE BASE OF LINE TO CLEAR
CD11: CD11: CD11: CD12: CD13: B4 CD13: B4 CD13: B4 CD14: F0 O1 CD17 CD14: B8 CD17: CD17: CD17 CD17: 78 CD17: 754 CD18: 20 54 CB CD18: 68	94 * DONE WITH 95 * 96 X.SCRLRET 97 98 99 100 101 X.SCRLRET2 102 103 104	EQU LDY TXA BEQ DEY EQU TYA JSR PLA	* WNDTOP,X X.SCRLRET2 *	CLEAR TOP OR BOTTOM LINE IF GETTING TOP, DON'T DECREMENT! TEMP CV SETUP COMPUTE BASE OF LINE TO CLEAR RESTORE
CD11: CD11: CD11: CD11: CD11: CD11: CD13: SA CD14:F0 01 CD17 CD17:	94 * DONE WITH 95 * 96 X.SCRLRET 97 98 99 99 100 101 X.SCRLRET2 102 103 104 105	EQU LDY TXA BEQ DEY EQU TYA JSR PLA TAX	* WNDTOP,X X.SCRLRET2 * BASCALCZ	CLEAR TOP OR BOTTOM LINE ; IF GETTING TOP, ; DON'T DECREMENT! ; TEMP CV SETUP ; COMPUTE BASE OF LINE TO CLEAR ; RESTORE ; X
CD11: CD11: CD11: CD12: CD13: B4 CD13: B4 CD13: B4 CD14: F0 O1 CD17 CD14: B8 CD17: CD17: CD17 CD17: 78 CD17: 754 CD18: 20 54 CB CD18: 68	94 * DONE WITH 95 * 96 X.SCRLRET 97 98 99 100 101 X.SCRLRET2 102 103 104	EQU LDY TXA BEQ DEY EQU TYA JSR PLA	* WNDTOP,X X.SCRLRET2 *	CLEAR TOP OR BOTTOM LINE IF GETTING TOP, DON'T DECREMENT! TEMP CV SETUP COMPUTE BASE OF LINE TO CLEAR RESTORE
CD11: CD11: CD11: CD11: CD11: CD13: SA CD14: SA CD14: SA CD17: CD12: CD20: CD2	94 * DONE WITH 95 * 94 96 X.SCRLRET 97 98 99 100 101 X.SCRLRET2 102 103 104 105 105 105 107 *	EQU LDY TXA BEQ DEY EQU TYA JSR PLA TAX JSR	* WNDTOP,X X.SCRLRET2 * BASCALCZ	CLEAR TOP OR BOTTOM LINE ; IF GETTING TOP, ; DON'T DECREMENT! ; TEMP CV SETUP ; COMPUTE BASE OF LINE TO CLEAR ; RESTORE ; X
CD11: CD11: CD11: CD11: CD11: CD11: CD12: CD13: BA CD14:F0 01 CD17 CD14: BA CD17: CD20: CD	94 * DONE WITH 95 * SCRLRET 97 X. SCRLRET 97 100 101 X. SCRLRET2 102 103 104 105 106 107 * 107 * 109	EQU LDY TXA BEQ DEY EQU TYA JSR PLA TAX	* WNDTOP,X X.SCRLRET2 * BASCALCZ	CLEAR TOP OR BOTTOM LINE ; IF GETTING TOP, ; DON'T DECREMENT! ; TEMP CV SETUP ; COMPUTE BASE OF LINE TO CLEAR ; RESTORE ; X
CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD17: CD20: CD	94 * DONE WITH 95 * 2000 WITH 97 97 99 99 100 101 X.SCRLRET2 102 103 104 104 105 104 105 106 107 * 108 X.LFRET 109 *	EQU LDY TXA BEG DEY EQU TYA JSR PLA TAX JSR EQU JMP	* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC	;CLEAR TOP OR BOTTOM LINE ;IF GETTING TOP, ; DON'T DECREMENT! ;TEMP CV SETUP ;COMPUTE BASE OF LINE TO CLEAR ;RESTORE ;X ;CLEAR BOTTOM LINE
CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD13: CD13: EA CD13: EA CD14: FO 01 CD17 CD17: CD18: CD18: CD18: CD18: CD18: CD12: CD20: C	94 * DONE WITH 95 * 96 X.SCRLRET 97 X.SCRLRET 97 100 100 X.SCRLRET2 102 103 104 105 106 107 * 108 X.LFRET 109 *	EQU LDY TXA BEG DEY EQU TYA JSR PLA TAX JSR EQU JMP	* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC	;CLEAR TOP OR BOTTOM LINE ;IF GETTING TOP, ; DON'T DECREMENT! ;TEMP CV SETUP ;COMPUTE BASE OF LINE TO CLEAR ;RESTORE ;X ;CLEAR BOTTOM LINE
CD11: CD11: CD11 CD11: CD11 CD11: B4 22 CD13: B4 CD14: F0 01 CD17 CD14: B8 CD17: CD17 CD17: CD17 CD17: R5 CD17: 78 CD18: 20 54 CB CD11: 48 CD11: 48 CD12: 44 CD12: 20 1A CC CD20: CD20: CD20 CD20: CD20 CD23: CD23: CD23:	94 * DONE WITH 95 * 200 CREET 97 99 100 101 X.SCRLRET2 102 103 104 105 106 107 * 108 X.LFRET 109 * 110 * 112 *	EQU LDY TXA BEQ DEY EQU TYA JSR PLA TAX JSR EQU JMP	* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC	;CLEAR TOP OR BOTTOM LINE ;IF GETTING TOP, ; DON'T DECREMENT! ;TEMP CV SETUP ;COMPUTE BASE OF LINE TO CLEAR ;RESTORE ;X ;CLEAR BOTTOM LINE
CD11: CD11: CD11: CD11 CD11: CD11 CD11: B4 22 CD13: BA CD14: F0 01 CD17 CD16: BB CD17: CD17 CD17: CD17 CD17: 78 CD18: 20 54 CB CD16: AA CD10: AA CD10: AA CD10: AA CD10: CD20: CD20: CD20 CD20: AC 51 CB CD23: CD2	94 * DONE WITH 95 * 96 X.SCRLRET 97 X.SCRLRET 97 100 100 X.SCRLRET2 102 103 104 105 106 107 * 108 X.LFRET 109 *	EGU LDY TXA BEG DEY EGU JSR PLA TAX JSR EGU JMP CLR TO	* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC	CLEAR TOP OR BOTTOM LINE IF GETTING TOP. DON'T DECREMENT! TEMP CV SETUP COMPUTE BASE OF LINE TO CLEAR RESTORE X CLEAR BOTTOM LINE RETURN VIA BASCALC (UGH!)
CD11: CD11: CD11: CD11: CD11: CD11: CD13: CD13: CD13: CD13: CD13: CD13: CD17: CD17: CD17: CD17: CD18: CD18: CD16: CD18: CD16: CD12: CD20: CD20: CD20: CD20: CD20: CD20: CD23: CD24: CD24: CD25: CD25: CD25: CD25: CD25: </td <td>94 * DONE WITH 95 * 96 X.SCRLRET 97 X.SCRLRET 97 100 101 X.SCRLRET2 102 103 104 105 106 107 * 108 X.LFRET 109 * 111 * EXECUTE 0 112 * 113 X.VT</td> <td>EQU LDY TXA BEG DEY EQU TYA JSR PLA JSR PLA JSR EQU JSR EQU JSR EQU JSR EQU JSR</td> <td>* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC EOS: *</td> <td>;CLEAR TOP OR BOTTOM LINE ;IF GETTING TOP, ; DON'T DECREMENT! ;TEMP CV SETUP ;COMPUTE BASE OF LINE TO CLEAR ;RESTORE ;X ;CLEAR BOTTOM LINE</td>	94 * DONE WITH 95 * 96 X.SCRLRET 97 X.SCRLRET 97 100 101 X.SCRLRET2 102 103 104 105 106 107 * 108 X.LFRET 109 * 111 * EXECUTE 0 112 * 113 X.VT	EQU LDY TXA BEG DEY EQU TYA JSR PLA JSR PLA JSR EQU JSR EQU JSR EQU JSR EQU JSR	* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC EOS: *	;CLEAR TOP OR BOTTOM LINE ;IF GETTING TOP, ; DON'T DECREMENT! ;TEMP CV SETUP ;COMPUTE BASE OF LINE TO CLEAR ;RESTORE ;X ;CLEAR BOTTOM LINE
CD11: CD11: CD11 CD11: CD11 CD11: CD11 CD11: BA CD13: BA CD14: FO<01	94 * DONE WITH 95 * 200 CREET 97 98 99 100 101 X.SCRLRET2 102 103 104 105 104 105 106 107 * 108 X.LFRET 108 X.LFRET 108 X.LFRET 110 * 112 * 113 X.VT 114 115 116	EQU LDY TXA BEG DEY EQU TYA JSR EQU JSR EQU JSR LDR TO EQU JSR LDR	* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC EOS: * X. es gurcv	CLEAR TOP OR BOTTOM LINE ; IF GETTING TOP, ; DON'T DECREMENT! ; TEMP CV SETUP ; COMPUTE BASE OF LINE TO CLEAR ; RESTORE ; X ; CLEAR BOTTOM LINE ; RETURN VIA BASCALC (UGH!) ; CLEAR TO EOL ; SAVE CV
CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD13: CD13: CD13: CD13: CD13: CD17: CD17: CD17: CD18: CD18: CD18: CD16: CD10: CD12: CD12: CD20: CD20: CD20: CD20: CD20: CD23: CD23: CD23: CD23: CD23: CD23: CD23: CD23: CD24: CD24: </td <td>94 * DONE WITH 95 * SCRLRET 97 * SCRLRET 97 00 100 100 101 X SCRLRET2 102 103 104 105 106 107 * 108 108 X LFRET 109 * 111 * EXECUTE C 112 * 113 113 X VT 114 115 116 117</td> <td>EQU LDY TXA BEG DEY EQU TYA JSR PLA TAX JSR EQU JMP CLR TO EQU JSR LDA PHA BPL</td> <td>* WNDTOP, X X. SCRLRETZ * BASCALCZ X. SUB * BASCALC EOS: * X. CS</td> <td>CLEAR TOP OR BOTTOM LINE ; IF GETTING TOP, ; DON'T DECREMENT! ; TEMP CV SETUP ; COMPUTE BASE OF LINE TO CLEAR ; RESTORE ; X ; CLEAR BOTTOM LINE ; CLEAR TO EOL</td>	94 * DONE WITH 95 * SCRLRET 97 * SCRLRET 97 00 100 100 101 X SCRLRET2 102 103 104 105 106 107 * 108 108 X LFRET 109 * 111 * EXECUTE C 112 * 113 113 X VT 114 115 116 117	EQU LDY TXA BEG DEY EQU TYA JSR PLA TAX JSR EQU JMP CLR TO EQU JSR LDA PHA BPL	* WNDTOP, X X. SCRLRETZ * BASCALCZ X. SUB * BASCALC EOS: * X. CS	CLEAR TOP OR BOTTOM LINE ; IF GETTING TOP, ; DON'T DECREMENT! ; TEMP CV SETUP ; COMPUTE BASE OF LINE TO CLEAR ; RESTORE ; X ; CLEAR BOTTOM LINE ; CLEAR TO EOL
CD11: CD11: CD11: CD11 CD11: CD11 CD11: B4 CD13:BA CD17 CD14:F0 01 CD17 CD17:78 CD17 CD18:20 54 CB CD19:20 1A CC CD20: CD20 CD23: CD23 CD23: CD23 CD23: CD23: CD24:AD FB 05 CD29:AB CD24:10 06 CD32 CD25: CD26	94 * DONE WITH 95 * 200 CREET 97 98 99 100 101 X.SCRLRET2 102 103 104 105 104 105 106 107 * 108 X.LFRET 108 X.LFRET 108 X.LFRET 110 * 112 * 113 X.VT 114 115 116 117 118 X.VTLOOP	EQU LDY TXA BEQ DEY EQU TYA JSR PLA TAX JSR CU JSR TO LR TO LR EQU JSR EQU JSR EQU JSR EQU EQU EQU EQU	* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC EOS: * X. CS OURCO X. VTNEXT *	CLEAR TOP OR BOTTOM LINE F GETTING TOP, DON'T DECREMENT! TEMP CV SETUP COMPUTE BASE OF LINE TO CLEAR RESTORE X CLEAR BOTTOM LINE CLEAR TO EOL SAVE CV DO NEXT LINE (ALWAYS TAKEN)
CD11: CD11: CD11 CD11: CD11 CD11: CD11 CD11: CD11 CD11: CD11 CD13: BA CD14: F0 CD16: BB CD17: CD17 CD17: CD17 CD18: 20 CD18: 20 CD18: 20 CD18: 20 CD10: AB CD11: 20 CD12: CD20 CD20: CD20 CD20: CD20 CD23: CD23 CD23: CD23: CD23: CD23: CD24: AP FB 05 CD24: AP FB 05 CD24: AP FB 05 CD24: CD32 CD26: CD26: CD26: CD26:	94 * DONE WITH 95 * DONE WITH 95 * SCRLRET 97 98 99 99 100 101 X SCRLRET2 102 103 104 105 104 105 106 107 * 108 X LFRET 109 111 * EXECUTE C 112 * 111 * 113 X VT 115 115 116 117 118 X VTLOOP	EQU LDY TXA BEQ DEY EQU TYA JSR PLA TAX JSR EQU JSR EQU JSR EQU JSR EQU JSR EQU JSR	* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC EQS: * X. GS QURCY X. VTNEXT * BASCALC	;CLEAR TOP OR BOTTOM LINE ;IF GETTING TOP. ;DON'T DECREMENT! ;COMPUTE BASE OF LINE TO CLEAR ;RESTORE ;X ;CLEAR BOTTOM LINE ;RETURN VIA BASCALC (UGH!) ;CLEAR TO EOL ;SAVE CV ;DO NEXT LINE (ALWAYS TAKEN) ;BASCALC IT
CD11: CD11: CD11: CD11 CD11: CD11 CD11: B4 CD13:BA CD17 CD14:F0 01 CD17 CD17:78 CD17 CD18:20 54 CB CD19:20 1A CC CD20: CD20 CD23: CD23 CD23: CD23 CD23: CD23: CD24:AD FB 05 CD29:AB CD24:10 06 CD32 CD25: CD26	94 * DONE WITH 95 * 200 CREET 97 98 99 100 101 X.SCRLRET2 102 103 104 105 104 105 106 107 * 108 X.LFRET 108 X.LFRET 108 X.LFRET 110 * 112 * 113 X.VT 114 115 116 117 118 X.VTLOOP	EQU LDY TXA BEQ DEY EQU TYA JSR PLA TAX JSR CU JSR TO LR TO LR EQU JSR EQU JSR EQU JSR EQU EQU EQU EQU	* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC EOS: * X. CS OURCO X. VTNEXT *	CLEAR TOP OR BOTTOM LINE F GETTING TOP, DON'T DECREMENT! TEMP CV SETUP COMPUTE BASE OF LINE TO CLEAR RESTORE X CLEAR BOTTOM LINE CLEAR TO EOL SAVE CV DO NEXT LINE (ALWAYS TAKEN)
CD11: CD1: CD1: CD1: CD2: CD2: CD2: CD2: CD2: CD2: CD2: CD2: <td>94 * DONE WITH 95 * 96 X.SCRLRET 97 X.SCRLRET 97 00 100 102 103 104 104 105 106 106 107 * 108 X.LFRET 109 111 * EXECUTE 0 112 * 113 X.VT 114 115 116 117 118 X.VTLDOP 120 121 X.VTNEXT 122</td> <td>EQU LDY TXA BEQ DEY EQU TYA JSR TAX JSR EQU JSR LDA PHA EQU LDA PHA EQU SR USR LDA PHA EQU LDA ISR ISR ISR ISR ISR ISR ISR</td> <td>* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC EOS: * X. GS GURCV X. VTNEXT * BASCALC X. SUB * BASCALC X. SUB * BASCALC X. SUB</td> <td>;CLEAR TOP OR BOTTOM LINE ;IF GETTING TOP. ;DON'T DECREMENT! ;COMPUTE BASE OF LINE TO CLEAR ;RESTORE ;X ;CLEAR BOTTOM LINE ;RETURN VIA BASCALC (UGH!) ;CLEAR TO EOL ;SAVE CV ;DO NEXT LINE (ALWAYS TAKEN) ;BASCALC IT</td>	94 * DONE WITH 95 * 96 X.SCRLRET 97 X.SCRLRET 97 00 100 102 103 104 104 105 106 106 107 * 108 X.LFRET 109 111 * EXECUTE 0 112 * 113 X.VT 114 115 116 117 118 X.VTLDOP 120 121 X.VTNEXT 122	EQU LDY TXA BEQ DEY EQU TYA JSR TAX JSR EQU JSR LDA PHA EQU LDA PHA EQU SR USR LDA PHA EQU LDA ISR ISR ISR ISR ISR ISR ISR	* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC EOS: * X. GS GURCV X. VTNEXT * BASCALC X. SUB * BASCALC X. SUB * BASCALC X. SUB	;CLEAR TOP OR BOTTOM LINE ;IF GETTING TOP. ;DON'T DECREMENT! ;COMPUTE BASE OF LINE TO CLEAR ;RESTORE ;X ;CLEAR BOTTOM LINE ;RETURN VIA BASCALC (UGH!) ;CLEAR TO EOL ;SAVE CV ;DO NEXT LINE (ALWAYS TAKEN) ;BASCALC IT
CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD17: CD20: CD20: CD20: CD20: CD20: CD20: CD22: CD23: CD23: CD23: CD23: CD22: CD	94 * DONE WITH 95 * CRLRET 97 98 99 99 100 101 X.SCRLRET2 102 103 104 105 106 107 * 108 X.LFRET 108 X.LFRET 108 * 111 * 112 * 113 X.VT 114 115 116 117 118 X.VTLOOP 117 120 121 X.VTNEXT 122 123	EQU LDY TXA BEQY EQU TYA JSR PLA JSR EQU JSR EQU JSR EQU JSR EQU JSR EQU LDA	* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC EQS: * X. GS QURCY X. VTNEXT * BASCALC X. SUB * OURCY OURCY OURCY	;CLEAR TOP OR BOTTOM LINE ;IF GETTING TOP. ;DON'T DECREMENT! ;COMPUTE BASE OF LINE TO CLEAR ;RESTORE ;X ;CLEAR BOTTOM LINE ;RETURN VIA BASCALC (UGH!) ;CLEAR TO EOL ;SAVE CV ;DO NEXT LINE (ALWAYS TAKEN) ;BASCALC IT ;CLEAR LINE ;BUMP CV
CD11: CD11: CD11 CD11: CD11 CD11: CD11 CD11: BA CD14: FO 01 CD14: FO 01 CD17: 78 CD18: CD CD17: 78 CD18: CD CD19: CD CD19: CD CD19: CD CD19: CD CD20: CD20 CD20: CD20 CD20: CD20 CD20: CD20 CD20: CD20 CD20: CD20 CD23: CD23 CD23: CD23: CD24: AD CD24: AD CD24: AD CD22: CD CD22: CD CD24: CD CD22: CD CD22: CD CD22: CD CD22: CD	94 * DONE WITH 95 * 96 X.SCRLRET 97 98 99 97 100 102 103 102 104 105 100 104 105 106 107 * 108 107 * 108 X.UFRET 108 X.UFRET 109 111 * EXECUTE 0 112 * 113 X.VT 114 5 115 116 117 118 X.VTLDOP 119 221 X.VTNEXT 122 124	EGU LDY TXA BEG DEY EGU TYA JSR TAX JSR EGU JSR EGU JSR EGU LDA PHA SR USR USR EGU CMP	* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC EOS: * X. GS OURCV OURCV X. VINEXT * BASCALC X. SUB * BASCALC X. SUB * BASCALC URCV URCV URCV URCV	CLEAR TOP OR BOTTOM LINE F GETTING TOP. TOP. COMPUTE BASE OF LINE TO CLEAR COMPUTE BASE OF LINE CLEAR TO EOL SAVE CV DO NEXT LINE (ALWAYS TAKEN) PBASCALC IT CLEAR LINE BUMP CV OFF SCREEN?
CD11: CD11: CD11 CD11: CD11 CD11: CD11 CD11: CD11 CD11: CD11 CD11: BA CD11: BA CD11: CD17 CD16: BB CD17: CD17 CD18: CD 4 CD11: AC CD12: CD20 CD20: CD20 CD20: CD20 CD20: CD20 CD20: CD20 CD23: CD23 CD23: CD23 CD24: AD<	94 * DONE WITH 95 * DONE WITH 95 * SCRLRET 97 98 99 99 100 101 X. SCRLRET2 102 103 104 105 104 105 106 107 * 108 X. LFRET 108 X. LFRET 108 X. LFRET 110 * 111 * 112 * 113 X. VT 114 115 116 117 118 X. VTLOOP 119 120 121 121 X. VTNEXT 122 123 124	EQU LDY TXA BEO DEY EQU JSR FNA JSR EQU JSR EQU JSR EQU JSR EQU JSR EQU JSR EQU SSR EQU CDA BFL EQU CDA CHO C	* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC EQS: * X. GS QURCY X. VTNEXT * BASCALC X. SUB * OURCY OURCY OURCY	<pre>;clear top or bottom line ;if getting top, ; don't decrement: ;compute base of line to clear ;restore ; x ;clear bottom line ;return via bascalc (ugh:) ;clear to eol ;save cv ;do next line (always taken) ;bascalc it ;clear line ;bump cv ;off screen? ;=>NO, KEEP going</pre>
CD11: CD1: CD1: CD1: CD2: CD2: CD2: CD2: CD2: CD2: CD2: CD2: CD2: <td>94 * DONE WITH 95 * 97 96 X.SCRLRET 97 98 99 100 101 X.SCRLRET2 102 103 104 105 106 107 * 108 107 * 108 107 * 108 109 * 107 111 * EXECUTE 0 112 * 113 115 116 117 118 X.VTLDOP 119 121 X.VTLDOP 121 X.VTNEXT 122 123 124 125</td> <td>EQU LDY TXA BE0 DEY TYA JSR TAX JSR EQU JSR LDA EQU JSR LDA SR EQU JSR LDA EQU LDA CMP EQU PLA</td> <td>* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC EOS: * X. GS OURCV OURCV WNDBTM X. VTLOOP</td> <td><pre>;CLEAR TOP OR BOTTOM LINE ;IF GETTING TOP. ; DON'T DECREMENT! ;COMPUTE BASE OF LINE TO CLEAR ;RESTORE ; X ;CLEAR BOTTOM LINE ;RETURN VIA BASCALC (UGH!) ;CLEAR TO EOL ;SAVE CV ;DO NEXT LINE (ALWAYS TAKEN) ;BASCALC IT ;CLEAR LINE ;BUMP CV ;OFF SCREEN? ;=>NO, KEEP GOING ;RESTORE</pre></td>	94 * DONE WITH 95 * 97 96 X.SCRLRET 97 98 99 100 101 X.SCRLRET2 102 103 104 105 106 107 * 108 107 * 108 107 * 108 109 * 107 111 * EXECUTE 0 112 * 113 115 116 117 118 X.VTLDOP 119 121 X.VTLDOP 121 X.VTNEXT 122 123 124 125	EQU LDY TXA BE0 DEY TYA JSR TAX JSR EQU JSR LDA EQU JSR LDA SR EQU JSR LDA EQU LDA CMP EQU PLA	* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC EOS: * X. GS OURCV OURCV WNDBTM X. VTLOOP	<pre>;CLEAR TOP OR BOTTOM LINE ;IF GETTING TOP. ; DON'T DECREMENT! ;COMPUTE BASE OF LINE TO CLEAR ;RESTORE ; X ;CLEAR BOTTOM LINE ;RETURN VIA BASCALC (UGH!) ;CLEAR TO EOL ;SAVE CV ;DO NEXT LINE (ALWAYS TAKEN) ;BASCALC IT ;CLEAR LINE ;BUMP CV ;OFF SCREEN? ;=>NO, KEEP GOING ;RESTORE</pre>
CD11: CD11: CD11 CD11: CD11 CD11: CD11 CD11: CD11 CD11: CD11 CD11: BA CD11: BA CD11: CD17 CD16: BB CD17: CD17 CD18: CD 4 CD11: CD 17 CD12: CD 17 CD12: CD 17 CD12: CD 14 CD12: CD 14 CD20: CD 20 CD23: CD 23 CD23: CD 23 CD23: CD 24 CD24: AD CD22: CD 25 CD24: AD CD25: CD 14 CD24: CD 25 CD23: CD 23 CD24: CD 24	94 * DONE WITH 95 * DONE WITH 95 * SCRLRET 97 99 99 100 101 X.SCRLRET2 102 103 104 105 104 105 104 105 106 107 * 108 X.LFRET 108 X.LFRET 108 X.LFRET 118 X.VT 112 * 114 115 116 117 118 X.VTLOOP 119 120 121 121 123 124 125 125 126	EQU LDY TXA BEO DEY EQU JSR FNA JSR EQU JSR EQU JSR EQU JSR EQU JSR EQU JSR EQU SSR EQU CDA BFL EQU CDA CMP	* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC EOS: * X. GS OURCV OURCV X. VINEXT * BASCALC X. SUB * BASCALC X. SUB * BASCALC URCV URCV URCV URCV	<pre>;CLEAR TOP OR BOTTOM LINE ;IF GETTING TOP, ;DON'T DECREMENT! ;COMPUTE BASE OF LINE TO CLEAR ;RESTORE ;X ;CLEAR BOTTOM LINE ;RETURN VIA BASCALC (UGH!) ;CLEAR TO EOL ;SAVE CV ;DO NEXT LINE (ALWAYS TAKEN) ;BASCALC IT ;CLEAR LINE ;BUMP CV ;OFF SCREEN? ;=>NO, KEEP GOING ;RESTORE ;CV</pre>
CD11: CD11: CD11 CD11: CD11 CD11: CD11 CD11: CD11 CD11: CD11 CD11: BA CD11: BA CD11: CD17 CD16: BB CD17: CD17 CD18: CD 4 CD11: CD 17 CD12: CD 17 CD12: CD 17 CD12: CD 14 CD12: CD 14 CD20: CD 20 CD23: CD 23 CD23: CD 23 CD23: CD 24 CD24: AD CD22: CD 25 CD24: AD CD25: CD 14 CD24: CD 25 CD23: CD 23 CD24: CD 24	94 * DONE WITH 95 * 97 96 X.SCRLRET 97 98 99 100 101 X.SCRLRET2 102 103 104 105 106 107 * 108 107 * 108 107 * 108 109 * 107 111 * EXECUTE 0 112 * 113 115 116 117 118 X.VTLOOP 119 121 X.VTLOOP 121 X.VTNEXT 122 123 124 125	EQU LDY TXA BEG DEY JYA JSR EQU JSR EQU JSR EQU JSR EQU JSR EQU JSR EQU JSR EQU SSR E SS E SSR E SS E S E E S E S E S E S E S E S E S E S E S E S E S E S E S E S E S E S E E S E S E S E S E S E E S E S E S E S E S E S E S E S E S E E S E S E E E E S E S E E E S E E S E E E E S E E E S E	* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC EOS: * X. eS OURCV X. VTNEXT * BASCALC X. SUB * CURCV OURCV OURCV OURCV OURCV OURCV OURCV	<pre>;CLEAR TOP OR BOTTOM LINE ;IF GETTING TOP. ; DON'T DECREMENT! ;COMPUTE BASE OF LINE TO CLEAR ;RESTORE ; X ;CLEAR BOTTOM LINE ;RETURN VIA BASCALC (UGH!) ;CLEAR TO EOL ;SAVE CV ;DO NEXT LINE (ALWAYS TAKEN) ;BASCALC IT ;CLEAR LINE ;BUMP CV ;OFF SCREEN? ;=>NO, KEEP GOING ;RESTORE</pre>
CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD13: CD13: CD13: CD13: CD13: CD17: CD17: CD11: CD11: CD17: CD11: CD1: CD2: CD2: CD2: CD2: CD2: CD2: CD2: CD2: CD2: CD2: <td>94 * DONE WITH 95 * SCRLRET 97 97 98 99 100 101 X.SCRLRET2 102 103 104 105 104 105 104 105 104 107 108 X.LFRET 109 111 * EXECUTE C 112 * 113 X.VT 114 115 116 117 117 118 X.VTLOOP 119 120 121 124 125 124 125 126 127 128 129 129 129 129 129 129 129 129</td> <td>EQU LDY TXA BEG DEY JSR EQU JSR EQU JSR EQU JSR EQU JSR EQU JSR EQU JSR EQU SSR E SS E S E S E E S E E S E S E S E S E S E S E S E E S E E E S E S E S E S E S E S E S E S E S E S E S E</td> <td>* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC EOS: * X. eS OURCV X. VTNEXT * BASCALC X. SUB * CURCV OURCV OURCV OURCV OURCV OURCV OURCV</td> <td><pre>;CLEAR TOP OR BOTTOM LINE ;IF GETTING TOP, ;DON'T DECREMENT! ;COMPUTE BASE OF LINE TO CLEAR ;RESTORE ;X ;CLEAR BOTTOM LINE ;RETURN VIA BASCALC (UGH!) ;CLEAR TO EOL ;SAVE CV ;DO NEXT LINE (ALWAYS TAKEN) ;BASCALC IT ;CLEAR LINE ;BUMP CV ;OFF SCREEN? ;=>NO, KEEP GOING ;RESTORE ;CV</pre></td>	94 * DONE WITH 95 * SCRLRET 97 97 98 99 100 101 X.SCRLRET2 102 103 104 105 104 105 104 105 104 107 108 X.LFRET 109 111 * EXECUTE C 112 * 113 X.VT 114 115 116 117 117 118 X.VTLOOP 119 120 121 124 125 124 125 126 127 128 129 129 129 129 129 129 129 129	EQU LDY TXA BEG DEY JSR EQU JSR EQU JSR EQU JSR EQU JSR EQU JSR EQU JSR EQU SSR E SS E S E S E E S E E S E S E S E S E S E S E S E E S E E E S E S E S E S E S E S E S E S E S E S E S E	* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC EOS: * X. eS OURCV X. VTNEXT * BASCALC X. SUB * CURCV OURCV OURCV OURCV OURCV OURCV OURCV	<pre>;CLEAR TOP OR BOTTOM LINE ;IF GETTING TOP, ;DON'T DECREMENT! ;COMPUTE BASE OF LINE TO CLEAR ;RESTORE ;X ;CLEAR BOTTOM LINE ;RETURN VIA BASCALC (UGH!) ;CLEAR TO EOL ;SAVE CV ;DO NEXT LINE (ALWAYS TAKEN) ;BASCALC IT ;CLEAR LINE ;BUMP CV ;OFF SCREEN? ;=>NO, KEEP GOING ;RESTORE ;CV</pre>
CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD17: CD20: CD20: CD20: CD20: CD20: CD20: CD20: CD22: CD	94 * DONE WITH 95 * 200 WITH 97 * 96 99 99 100 X.SCRLRET2 102 103 104 105 104 105 106 107 * 108 X.LFRET 109 110 * 111 * EXECUTE C 113 X.VT 114 115 116 117 118 X.VTLOOP 119 120 121 123 124 123 124 125 125 126 127 128 128 128 128 128 128 128 128 128 128	EQU LDY TXA DEY EQU DEY SR EQU JSR EQU JSR EQU JSR EQU JSR EQU JSR EQU SSR E SSR E SSR E SSR E SSR E SSR E SSR E SSR SSR	* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC EOS: * X. eS OURCV X. VTNEXT * BASCALC X. SUB * CURCV OURCV OURCV OURCV OURCV OURCV OURCV	<pre>;CLEAR TOP OR BOTTOM LINE ;IF GETTING TOP, ;DON'T DECREMENT! ;COMPUTE BASE OF LINE TO CLEAR ;RESTORE ;X ;CLEAR BOTTOM LINE ;RETURN VIA BASCALC (UGH!) ;CLEAR TO EOL ;SAVE CV ;DO NEXT LINE (ALWAYS TAKEN) ;BASCALC IT ;CLEAR LINE ;BUMP CV ;OFF SCREEN? ;=>NO, KEEP GOING ;RESTORE ;CV</pre>
CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD13: CD13: CD13: CD13: CD13: CD17: CD17: CD11: CD11: CD117: CD11: CD1: CD1: CD2: CD2: CD2: CD2: CD2: CD2: CD2: CD2: CD2: CD2: <td>94 * DONE WITH 95 * SCRLRET 97 97 98 99 100 101 X.SCRLRET2 102 103 104 105 104 105 104 105 106 107 * 108 X.LFRET 109 111 * EXECUTE C 112 * 113 X.VT 114 115 116 117 117 118 X.VTLOOP 119 120 121 124 125 126 127 128 129 * 130 * EXECUTE C 121 124 125 126 127 128 129 * 130 * 131 * 128 129 * 131 * 129 * 130 * 131 * 128 128 129 * 130 * 131 * 128 128 129 * 130 * 131 * 128 128 128 128 129 * 130 * 131 * 128 128 128 129 * 130 * 131 * 128 128 129 * 130 * 131 * 132 * 130 * 132 *</td> <td>EQU LDY TXA BEG DEY TXA SEG JSR EGU JSR EGU JSR EGU JSR EGU JSR EGU JSR EGU JSR EGU JSR EGU LDA SC EGU LDA SC EGU SC SC EGU SC E EGU SC E EGU SC E E EGU SC EC E EC EC EC EC EC EC EC EC EC EC EC</td> <td>* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC EOS: * X. CS GURCV UNCLY X. SUB * UNCDBTM X. VTLOOP OURCV X. LFRET *</td> <td><pre>;CLEAR TOP OR BOTTOM LINE ;IF GETTING TOP. ; DON'T DECREMENT! ;COMPUTE BASE OF LINE TO CLEAR ;RESTORE ;X ;CLEAR BOTTOM LINE ;RETURN VIA BASCALC (UGH!) ;CLEAR TO EOL ;SAVE CV ;DO NEXT LINE (ALWAYS TAKEN) ;BASCALC IT ;CLEAR LINE ;BUMP CV ;OFF SCREEN? ;=>NO, KEEP GOING ;RESTORE ;CV ;RETURN VIA SIMILAR CODE</pre></td>	94 * DONE WITH 95 * SCRLRET 97 97 98 99 100 101 X.SCRLRET2 102 103 104 105 104 105 104 105 106 107 * 108 X.LFRET 109 111 * EXECUTE C 112 * 113 X.VT 114 115 116 117 117 118 X.VTLOOP 119 120 121 124 125 126 127 128 129 * 130 * EXECUTE C 121 124 125 126 127 128 129 * 130 * 131 * 128 129 * 131 * 129 * 130 * 131 * 128 128 129 * 130 * 131 * 128 128 129 * 130 * 131 * 128 128 128 128 129 * 130 * 131 * 128 128 128 129 * 130 * 131 * 128 128 129 * 130 * 131 * 132 * 130 * 132 *	EQU LDY TXA BEG DEY TXA SEG JSR EGU JSR EGU JSR EGU JSR EGU JSR EGU JSR EGU JSR EGU JSR EGU LDA SC EGU LDA SC EGU SC SC EGU SC E EGU SC E EGU SC E E EGU SC EC E EC EC EC EC EC EC EC EC EC EC EC	* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC EOS: * X. CS GURCV UNCLY X. SUB * UNCDBTM X. VTLOOP OURCV X. LFRET *	<pre>;CLEAR TOP OR BOTTOM LINE ;IF GETTING TOP. ; DON'T DECREMENT! ;COMPUTE BASE OF LINE TO CLEAR ;RESTORE ;X ;CLEAR BOTTOM LINE ;RETURN VIA BASCALC (UGH!) ;CLEAR TO EOL ;SAVE CV ;DO NEXT LINE (ALWAYS TAKEN) ;BASCALC IT ;CLEAR LINE ;BUMP CV ;OFF SCREEN? ;=>NO, KEEP GOING ;RESTORE ;CV ;RETURN VIA SIMILAR CODE</pre>
CD11: CD11: CD11 CD11: CD11 CD11: CD11 CD11: CD11 CD11: CD11 CD11: BA CD11: BA CD11: CD17 CD12: CD17 CD17: CD17 CD11: CD17 CD12: CD2 CD20: CD20 CD20: CD20 CD22: CD2 CD23: CD22 CD24: CD24 CD22: CD1 CD22: CD1 CD22: CD1 CD22: CD1 CD24: CD22 CD25: CD1 CD26: <td>94 * DONE WITH 95 * DONE WITH 97 * 96 97 99 100 X.SCRLRET2 102 103 104 105 104 105 107 * 108 108 X.LFRET 109 110 * 111 112 * EXECUTE OF 113 X.VT 114 115 116 117 118 X.VTLOOP 119 118 120 121 121 121 122 123 124 125 124 125 126 127 128 128 128 128 128 128 128 128 128 128</td> <td>EQU LDY TXA BEO DEY EQU TXA JSR FLA JSR COMP EQU LDA SR EQU LDA SR EQU LDA SR EQU LDA SR EQU LDA SR EQU SR E SR E SR E SR E SR E SR E SR E SR</td> <td>* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC EOS: * X. eS GURCV X. VTNEXT * BASCALC X. SUB * URCV UNEXT * URCV UNEXT * C. SUB * X. CS GURCV UNEXT * C. SUB * X. SCRLRET2 * A SCALCZ * A * A SCALCZ * SCALCZ * A SCALCZ * S * A SCALCZ * S * * S * S *</td> <td><pre>;clear top or bottom line ;if getting top, ; don't decrement: ;compute base of line to clear ;restore ; x ;clear bottom line ;return via bascalc (ugh!) ;clear to eol ;save cv ;do next line (always taken) ;bascalc it ;clear line ;bump cv ;off screen? ;=>NO, keep going ;return via similar code ;eturn via similar code</pre></td>	94 * DONE WITH 95 * DONE WITH 97 * 96 97 99 100 X.SCRLRET2 102 103 104 105 104 105 107 * 108 108 X.LFRET 109 110 * 111 112 * EXECUTE OF 113 X.VT 114 115 116 117 118 X.VTLOOP 119 118 120 121 121 121 122 123 124 125 124 125 126 127 128 128 128 128 128 128 128 128 128 128	EQU LDY TXA BEO DEY EQU TXA JSR FLA JSR COMP EQU LDA SR EQU LDA SR EQU LDA SR EQU LDA SR EQU LDA SR EQU SR E SR E SR E SR E SR E SR E SR E SR	* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC EOS: * X. eS GURCV X. VTNEXT * BASCALC X. SUB * URCV UNEXT * URCV UNEXT * C. SUB * X. CS GURCV UNEXT * C. SUB * X. SCRLRET2 * A SCALCZ * A * A SCALCZ * SCALCZ * A SCALCZ * S * A SCALCZ * S * * S * S *	<pre>;clear top or bottom line ;if getting top, ; don't decrement: ;compute base of line to clear ;restore ; x ;clear bottom line ;return via bascalc (ugh!) ;clear to eol ;save cv ;do next line (always taken) ;bascalc it ;clear line ;bump cv ;off screen? ;=>NO, keep going ;return via similar code ;eturn via similar code</pre>
CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD11: CD13: CD13: CD13: CD13: CD13: CD17: CD17: CD11: CD11: CD117: CD11: CD1: CD1: CD2: CD2: CD2: CD2: CD2: CD2: CD2: CD2: CD2: CD2: <td>94 * DONE WITH 95 * SCRLRET 97 97 98 99 100 101 X.SCRLRET2 102 103 104 105 104 105 104 105 106 107 * 108 X.LFRET 109 111 * EXECUTE C 112 * 113 X.VT 114 115 116 117 117 118 X.VTLOOP 119 120 121 124 125 126 127 128 129 * 130 * EXECUTE C 121 124 125 126 127 128 129 * 130 * 131 * 128 129 * 131 * 129 * 130 * 131 * 128 128 129 * 130 * 131 * 128 128 129 * 130 * 131 * 128 128 128 128 129 * 130 * 131 * 128 128 128 129 * 130 * 131 * 128 128 129 * 130 * 131 * 132 * 130 * 132 *</td> <td>EQU LDY TXA BEG DEY TXA SEG JSR EGU JSR EGU JSR EGU JSR EGU JSR EGU JSR EGU JSR EGU JSR EGU LDA SC EGU LDA SC EGU SC SC EGU SC E EGU SC E EGU SC E E EGU SC EC E EC EC EC EC EC EC EC EC EC EC EC</td> <td>* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC EOS: * X. CS GURCV UNCLY X. SUB * UNCDBTM X. VTLOOP OURCV X. LFRET *</td> <td><pre>;CLEAR TOP OR BOTTOM LINE ;IF GETTING TOP. ; DON'T DECREMENT! ;COMPUTE BASE OF LINE TO CLEAR ;RESTORE ;X ;CLEAR BOTTOM LINE ;RETURN VIA BASCALC (UGH!) ;CLEAR TO EOL ;SAVE CV ;DO NEXT LINE (ALWAYS TAKEN) ;BASCALC IT ;CLEAR LINE ;BUMP CV ;OFF SCREEN? ;=>NO, KEEP GOING ;RESTORE ;CV ;RETURN VIA SIMILAR CODE</pre></td>	94 * DONE WITH 95 * SCRLRET 97 97 98 99 100 101 X.SCRLRET2 102 103 104 105 104 105 104 105 106 107 * 108 X.LFRET 109 111 * EXECUTE C 112 * 113 X.VT 114 115 116 117 117 118 X.VTLOOP 119 120 121 124 125 126 127 128 129 * 130 * EXECUTE C 121 124 125 126 127 128 129 * 130 * 131 * 128 129 * 131 * 129 * 130 * 131 * 128 128 129 * 130 * 131 * 128 128 129 * 130 * 131 * 128 128 128 128 129 * 130 * 131 * 128 128 128 129 * 130 * 131 * 128 128 129 * 130 * 131 * 132 * 130 * 132 *	EQU LDY TXA BEG DEY TXA SEG JSR EGU JSR EGU JSR EGU JSR EGU JSR EGU JSR EGU JSR EGU JSR EGU LDA SC EGU LDA SC EGU SC SC EGU SC E EGU SC E EGU SC E E EGU SC EC E EC EC EC EC EC EC EC EC EC EC EC	* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC EOS: * X. CS GURCV UNCLY X. SUB * UNCDBTM X. VTLOOP OURCV X. LFRET *	<pre>;CLEAR TOP OR BOTTOM LINE ;IF GETTING TOP. ; DON'T DECREMENT! ;COMPUTE BASE OF LINE TO CLEAR ;RESTORE ;X ;CLEAR BOTTOM LINE ;RETURN VIA BASCALC (UGH!) ;CLEAR TO EOL ;SAVE CV ;DO NEXT LINE (ALWAYS TAKEN) ;BASCALC IT ;CLEAR LINE ;BUMP CV ;OFF SCREEN? ;=>NO, KEEP GOING ;RESTORE ;CV ;RETURN VIA SIMILAR CODE</pre>
CD11: CD11: CD11 CD11: CD11 CD11: CD11 CD11: CD11 CD11: A CD13: BA CD17: CD17 CD17: CD17 CD11: A CD11: CD10: CD12: CD20 CD20: CD20 CD20: CD20 CD23: CD23 CD23: CD23 CD24: A CD22: CD32 CD22: CD32 CD22: CD32 CD22: CD32 CD22: CD32 CD22: CD32 CD32: CD32 CD32: CD32 CD34: CD42 CD34: CD	94 * DONE WIT 95 * DONE WIT 95 * SCRLRET 97 98 99 99 100 101 X. SCRLRET2 102 103 104 105 104 105 104 105 106 107 * 108 X. LFRET 109 111 * EXECUTE 0 112 * 115 116 117 118 X. VTLOP 117 120 121 X. VTLOP 119 120 121 124 125 126 127 123 124 127 123 124 127 128 129 * 130 * 131 * 132 * 133 * 134 135 * 136 136 137 138 138 139 139 139 139 139 139 139 139	EQU LDY TXA BEO DEY EQU TXA JSR SR JSR LDA EQU LDA SR SR LDA SR LDA SR EQU LDA SR EQU SR SR LDA SR EQU SR SR LDA SR EQU SR SR SR SR SR SR SR SR SR SR SR SR SR	* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC EOS: * X. SUB * CURCV X. VTNEXT * BASCALC CURCV VINEXT * BASCALC CURCV VINEXT * SASCALC EOS: * X. SUB * SASCALC EOS: * X. SUB * * SASCALC EOS: * X. SUB * * SASCALC EOS: * * X. SUB * * SASCALC * SASCALC * * SASCALC * * * SASCALC * * * SASCALC * * * SASCALC * * * * * * * * SASCALC * * * * * * * * * * * * * * * * * * *	<pre>;clear top or bottom line ;if getting top, ; don't decrement: ;compute base of line to clear ;restore ; x ;clear bottom line ;return via bascalc (ugh!) ;clear to eol ;save cv ;do next line (always taken) ;bascalc it ;clear line ;bump cv ;off screen? ;=>NO, keep going ;return via similar code ;eturn via similar code</pre>
CD11: CD11: CD11 CD11: CD11 CD11: CD11 CD11: CD11 CD11: BA CD11: BA CD11: CD17 CD117: CD17 CD118: CD CD117: CD17 CD118: CD CD112: CA CD12: CD20 CD20: CD20 CD22: CD20: CD22: CD20: CD24: CD24: CD24: CD22 CD22: CD32 CD22: CD32 CD22: CD32 CD24: CD24 CD24: CD24 CD42:	94 * DONE WITH 95 * DONE WITH 97 * 96 97 99 99 101 X.SCRLRET2 102 103 104 105 104 105 107 * 108 101 * X.URET 109 * 112 * EXECUTE 0 112 * 113 114 115 116 117 118 X.VTL10P 119 118 X.VTL00P 119 120 121 X.VTNEXT 122 124 125 126 127 128 129 * 133 134 139 * EXECUTE 0 137 *	EQU LDY TXA BEO DEY EQU TXA JSR FLA JSR CAL EQU JSR LDA EQU JSR EQU JSR EQU JSR EQU JSR EQU SSR SSR SSR SSR SSR SSR SSR SSR SSR SS	* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC EOS: * X. SUB * BASCALC CURCV X. VTNEXT * BASCALC X. SUB * BASCALC URCV URCV V. VINEXT * BASCALC URCV URCV V. SUB * CORCV V. SUB * BASCALC URCV URCV V. SUB * BASCALC URCV URCV X. SUB * BASCALC URCV URCV V. SUB * BASCALC URCV V. SUB * SASCALC EOS: * SASCALC EOS: * SASCALC URCV V. SUB * SASCALC EOS: * SASCALC URCV V. SUB * SASCALC EOS: * SASCALC URCV V. SUB * SASCALC URCV V. SUB * SASCALC URCV V. SUB * SASCALC EOS: * SASCALC URCV V. SUB * SASCALC URCV V. SUB * SASCALC URCV V. SUB * SASCALC URCV V. SUB * SASCALC URCV V. SUB * SASCALC * S SASCALC * S SASCALC * S SASCALC * S SASCALC * S SASCALC * S SASCALC * S SASCALC * S SASCALC * S S S SASCALC * S SASCALC * S S S S S S S S S S S S S S S S S S	<pre>;CLEAR TOP OR BOTTOM LINE ;IF GETTING TOP, ;DON'T DECREMENT: ;COMPUTE BASE OF LINE TO CLEAR ;RESTORE ;X ;CLEAR BOTTOM LINE ;RETURN VIA BASCALC (UGH!) ;CLEAR TO EOL ;SAVE CV ;DD NEXT LINE (ALWAYS TAKEN) ;BASCALC IT ;CLEAR LINE ;BUMP CV ;OFF SCREEN? ;=>NO, KEEP GOING ;RESTORE ;CV ;RETURN VIA SIMILAR CODE ;HOME THE CURSOR</pre>
CD11: CD11: CD11 CD11: CD11 CD11: CD11 CD11: CD11 CD11: A CD13: BA CD17: CD17 CD17: CD17 CD11: A CD11: CD10: CD12: CD20 CD20: CD20 CD20: CD20 CD23: CD23 CD23: CD23 CD24: A CD22: CD32 CD22: CD32 CD22: CD32 CD22: CD32 CD22: CD32 CD22: CD32 CD32: CD32 CD32: CD32 CD34: CD42 CD34: CD	94 * DONE WIT 95 * DONE WIT 95 * SCRLRET 97 98 99 99 100 101 X. SCRLRET2 102 103 104 105 104 105 104 105 106 107 * 108 X. LFRET 109 111 * EXECUTE 0 112 * 115 116 117 118 X. VTLOP 117 120 121 X. VTLOP 119 120 121 124 125 126 127 123 124 127 123 124 127 128 129 * 130 * 131 * 132 * 133 * 134 135 * 136 136 137 138 138 139 139 139 139 139 139 139 139	EQU LDY TXA BEO DEY EQU TXA JSR SR JSR LDA EQU LDA SR SR LDA SR LDA SR EQU LDA SR EQU SR SR LDA SR EQU SR SR LDA SR EQU SR SR SR SR SR SR SR SR SR SR SR SR SR	* WNDTOP, X X. SCRLRET2 * BASCALCZ X. SUB * BASCALC EOS: * X. SUB * CURCV X. VTNEXT * BASCALC CURCV VINEXT * BASCALC CURCV VINEXT * SASCALC EOS: * X. SUB * SASCALC EOS: * X. SUB * * SASCALC EOS: * X. SUB * * SASCALC EOS: * * X. SUB * * SASCALC * SASCALC * * SASCALC * * * SASCALC * * * SASCALC * * * SASCALC * * * * * * * * SASCALC * * * * * * * * * * * * * * * * * * *	<pre>;clear top or bottom line ;if getting top, ; don't decrement: ;compute base of line to clear ;restore ; x ;clear bottom line ;return via bascalc (ugh!) ;clear to eol ;save cv ;do next line (always taken) ;bascalc it ;clear line ;bump cv ;off screen? ;=>NO, keep going ;return via similar code ;eturn via similar code</pre>

CD4B: AC	7B	05	139		LDY	DURCH	JOET CH
CD4B: 4C	54	CD	140		JMP	X. G52	CHECK FOR END FIRST!
CD4E CD4E: A9		CD4E	141	X. GSEOLZ	EQU	* # '	FER U HACKERS
CD30 20	AU ED	CE	142 143		JSR	# ' STORCHAR	STUFF IT
CD53 CB	<i>г и</i> .	UE.	144		INY	SIUNCHAR	ISTOFF II
CD54		CD54		X. 652	EQU	*	
CD54 C4			146		CPY	WNDWDTH	STOP SOMETIME
CD56: 90		CD4E	147		BCC	X. GSEDLZ	YASL DO MORE
CD58: 60			148		RTS		
CD59: CD59			149	* * EXECUTE			
CD57			150		40CUL	MUDET	
CD59		CD59		X. DC 1	EQU	*	
CD59: A9	00		153		LDA	#0	ASSUME TEXTMODE
CD58:85	20		154		STA	WNDLFT	
CD5D: 2C	1A	CO	155		BIT	RDTEXT	FARE WE IN TEXT MODE?
CD60: 30 CD62: A9	02	CD64			BMI	X. DC1B	;=>YES
CD62: A9	14	CD64	157	X. DC1B	LDA	#20	; IF GR, SET SPLITSCREEN
CD64:85	22		159	X. DC15	STA	WNDTOP	
CD66: A9	18		160		LDA	#24	
CD68: 85 CD6A: A9	23		161		STA	WNDBTM	
CD6A: A9	28		162		LDA	#40	
CD4C: 85	21		163		STA	WNDWDTH	
CD6E: 2C CD71: 10	1F	CO	164		BIT	RDBOVID	;WERE WE IN BO-MODE?
CD73: 20	03	CD76	165 166		BPL JSR	X. DC1RTS	x ≠>ND, NO CVT NEEDED
CD76:	18	CD76		X. DCIRTS	EQU	SCRN84	; CVT 80>40
CD76: 60			168		RTS		
CD77:			169				
CD77:			170	* EXECUTE	'SOCOL	MODE 1:	
CD77:			171				
CD77: CD77: 20	24	CD77	172	X. DC2	EQU	* TESTCARD	LO CARR THERE
CD7A DO	1E	CD9A	174		BNE		; IS CARD THERE? ;=>NDPE; FORGET IT
CD7A: DO CD7A: DO CD7C: 20	9B	CD	175		JSR	FULLBO	SET CULL MINDON
			176		BIT	RDTEXT	SET FULL WINDOW ARE WE IN TEXT MODE?
CD82: 30	04	CD88	177		BMI	X. DC2B	;=>YES
CD84: A9			178		LDA	#20	IF GR. SET SPLITSCREEN
CD86: 85 CD88:	22		179		STA	WNDTOP	
CD88: 2C	18	CDBB	180	X. DC2B	EQU	RDBOCOL	SEMEMBER BRIDE MODE
CD8B: 30			182		BMI	X. DC2RET	REMEMBER PRIOR MODE
CD8D: 4C	32	CF.	183			SCRN4B	;=>ND CVT NEEDED IF WAS 80 ;RET VIA CONVERT 40>80
					JMP		
CD90:		CE	184	*	-	SCRN4B	RET VIA CUNVERT 40 280
CD90: CD90:		CE.	184 185	* EXECUTE	-	SCRN4B	RET VIA CUNVERT 40280
CD90: CD90: CD90:			184 185 186	* EXECUTE *	'QUIT':	SCRN4B	THE VIA CUNVERT 40380
CD90: CD90: CD90:	ED		184 185 186 187	* EXECUTE	'QUIT': EQU	#	
CD90: CD90: CD90: CD90: CD90: CD90: AD	FB 20	CD90 04	184 185 186 187 188	* EXECUTE *	'QUIT': EQU LDA	* MODE	DNLY VALID IN BASIC
CD90: CD90: CD90: CD90: CD90: AD CD93: 29	20	CD90 04	184 185 186 187 188 189	* EXECUTE *	'QUIT': EQU LDA AND	* MODE #M. PASCAL	
CD90: CD90: CD90: CD90: CD90: AD CD93: 29 CD95: D0 CD97: 20	03	CD90 04 CD9A CD	184 185 186 187 188 189 190 190	* EXECUTE * X. NAK	'QUIT': EQU LDA	* MODE	ONLY VALID IN BASIC
CD90: CD90: CD90: CD90: CD90: AD CD93: 29 CD95: D0 CD97: 20 CD94:	03	CD90 04 CD9A CD CD9A	184 185 186 187 188 189 190 191 192	* EXECUTE * X. NAK X. NAKRET	'QUIT': EQU LDA AND BNE JSR EQU	* MODE #M. PASCAL X. NAKRET GUIT *	DNLY VALID IN BASIC
CD90: CD90: CD90: CD90: AD CD93: 29 CD95: D0 CD97: 20 CD9A: CD9A:	03	CD90 04 CD9A CD	184 185 186 187 188 189 190 191 192 193	* EXECUTE * X. NAK	'QUIT': EQU LDA AND BNE JSR EQU EQU	* MODE #M. PASCAL X. NAKRET GUIT	; ONLY VALID IN BASIC ; IGNORE IF PASCAL ; GET SETUP TO QUIT
CD90: CD90: CD90: CD90: CD90: AD CD93: 29 CD93: 29 CD95: D0 CD97: 20 CD97: 20 CD9A: CD9A: CD9A:	03	CD90 04 CD9A CD CD9A	184 185 186 187 188 189 190 191 192 193 194	* EXECUTE * X. NAK X. NAKRET X. DC2RET	'QUIT': EQU LDA AND BNE JSR EQU	* MODE #M. PASCAL X. NAKRET GUIT *	DNLY VALID IN BASIC
CD90: CD90: CD90: CD90: AD CD93: 29 CD95: D0 CD97: 20 CD9A: CD9A:	03	CD90 04 CD9A CD CD9A	184 185 186 187 188 187 188 187 190 191 192 193 194 195	* EXECUTE * X. NAK X. NAKRET X. DC2RET	'QUIT' EQU LDA AND BNE JSR EQU EQU RTS	* MODE #M. PASCAL X. NAKRET GUIT *	; ONLY VALID IN BASIC ; IGNORE IF PASCAL ; GET SETUP TO QUIT
CD90: CD90: CD90: CD90: AD CD93: 29 CD95: D0 CD97: 20 CD97: 20 CD9A: CD9A: CD9A: CD9A: CD9B: CD9B:	03	CD90 04 CD9A CD CD9A	184 185 186 187 188 187 190 191 192 193 194 195 196 197	* EXECUTE * X. NAK X. NAKRET X. DC2RET * NAME * FUNCTION	'QUIT': EQU LDA AND BNE JSR EQU EQU RTS FULL8 SET F	* MDDE #M. PASCAL X. NAKRET GUIT * * *	; ONLY VALID IN BASIC ; IGNORE IF PASCAL ; GET SETUP TO QUIT ; DONE; CALLER WON'T RETURN
CD90: CD90: CD90: CD90: CD90: AD CD93: 29 CD93: 29 CD97: 20 CD97: 20 CD9A: CD9A: CD9A: CD9A: CD9B: CD9B: CD9B:	03	CD90 04 CD9A CD CD9A	184 185 186 187 188 187 190 191 192 193 194 195 196 197	* EXECUTE * X. NAK X. NAKRET X. DC2RET * NAME * FUNCTION	'QUIT': EQU LDA AND BNE JSR EQU EQU RTS FULL8 SET F	* MDDE #M. PASCAL X. NAKRET GUIT * * *	; ONLY VALID IN BASIC ; IGNORE IF PASCAL ; GET SETUP TO QUIT ; DONE; CALLER WON'T RETURN
CD90: CD90: CD90: CD90: CD90: AD CD93: 29 CD95: D0 CD97: 20 CD97: 20 CD97: CD9A: CD9A: 60 CD9B: CD9B: CD9B: CD9B:	03	CD90 04 CD9A CD CD9A	184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199	* EXECUTE * X. NAK X. NAKRET X. DC2RET * NAME * FUNCTION * INPUT * OUTPUT	'QUIT': EQU LDA AND BNE JSR EQU EQU RTS : FULL8 SET F NONE WINDO	* MDDE #M.PASCAL X.NAKRET GUIT * *	; ONLY VALID IN BASIC ; IGNORE IF PASCAL ; GET SETUP TO QUIT ; DONE; CALLER WON'T RETURN
CD90: CD90: CD90: CD90: CD90: AD CD93: 29 CD95: D0 CD97: D0 CD97: D0 CD97: CD9A: CD9A: CD9A: CD9A: CD9B: CD9B: CD9B: CD9B: CD9B:	03	CD90 04 CD9A CD CD9A	184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200	* EXECUTE * X. NAK X. NAKRET X. DC2RET * NAME * FUNCTION * INPUT * VOLATILE	'QUIT': EQU LDA AND BNE JSR EQU EQU RTS : FULL8 SET F NONE WINDO	* MDDE #M. PASCAL X. NAKRET GUIT * * *	; ONLY VALID IN BASIC ; IGNORE IF PASCAL ; GET SETUP TO QUIT ; DONE; CALLER WON'T RETURN
CD90: CD90: CD90: CD90: CD90: AD CD93: 29 CD95: D0 CD97: 20 CD97: 20 CD97: CD9A: CD9A: 60 CD9B: CD9B: CD9B: CD9B:	03	CD90 04 CD9A CD CD9A	184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201	* EXECUTE * EXECUTE X. NAK X. NAKRET X. DC2RET * NAME * FUNCTION * INPUT * OUTPUT * OUTPUT * VOLATILE	'QUIT': EQU LDA AND BNE JSR EQU EQU RTS : FULL8 SET F NONE WINDO	* MDDE #M. PASCAL X. NAKRET GUIT * * *	; ONLY VALID IN BASIC ; IGNORE IF PASCAL ; GET SETUP TO QUIT ; DONE; CALLER WON'T RETURN
CD90: CD90: CD90: CD90: CD90: AD CD93: 29 CD93: 29 CD93: 20 CD97: 20 CD9A: CD9A: CD9A: CD9B: CD9B: CD9B: CD9B: CD9B: CD9B: CD9B: CD9B: CD9B:	03	CD90 04 CD9A CD CD9A	184 185 186 187 190 191 192 194 195 196 197 198 197 200 201 202 203	* EXECUTE * EXECUTE X. NAK X. NAKRET X. DC2RET * NAME * FUNCTION * INPUT * OUTPUT * OUTPUT * VOLATILE	GUIT ': EQU LDA AND BNE JSR EQU RTS FULL8 SET F NONE WINDO AC EQU	* MDDE MM.PASCAL X.NAKRET QUIT * OUL BOCOL WIT W PARAMETERS	; ONLY VALID IN BASIC ; IGNORE IF PASCAL ; GET SETUP TO QUIT ; DONE; CALLER WON'T RETURN
CD90: CD90: CD90: CD90: CD90: CD93: 29 CD93: 29 CD93: 20 CD93: 20 CD93: 20 CD93: 20 CD93: 20 CD93: 40 CD98: CD98: CD98: CD98: CD98: CD98: CD98:	03	CD90 04 CD9A CD CD9A CD9A	184 185 186 187 188 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204	* EXECUTE * X. NAK X. NAKRET X. DC2RET * NAME * FUNCTION * INPUT * VOLATILE *	GUIT ': EQU LDA AND BNE SNR EQU RTS FULL8 SET F WINDC AC EQU LDA	* MODE #M.PASCAL X.NAKRET GUIT * ULL BOCOL WIT W PARAMETERS * *	; ONLY VALID IN BASIC ; IGNORE IF PASCAL ; GET SETUP TO QUIT ; DONE; CALLER WON'T RETURN
CD90: CD90: CD90: CD90: CD90: CD90: CD93: 29 CD93: 29 CD93: 20 CD93: CD93: CD98: CD9	03 AA 00 22	CD90 04 CD9A CD CD9A CD9A	184 185 186 187 188 189 190 191 192 193 194 195 197 198 197 201 202 203 203 203	* EXECUTE * X. NAK X. NAKRET X. DC2RET * NAME * FUNCTION * INPUT * VOLATILE *	GUIT ': EQU LDA AND BNE SNE EQU RTS EQU RTS : SET F NONE : NONE : WINDO : AC EQU LDA STA	* MODE *M. PASCAL *. NAKRET GUIT * * O ULL BOCOL WIN W PARAMETERS * * * * * *	; ONLY VALID IN BASIC ; IGNORE IF PASCAL ; GET SETUP TO QUIT ; DONE; CALLER WON'T RETURN
CD90: CD90: CD90: CD90: CD90: CD90: CD93: 20 CD93: 20 CD93: 20 CD93: CD93: CD94: CD94: CD98: CD9	20 03 AA 00 22 20	CD90 04 CD9A CD CD9A CD9A	184 185 186 187 190 191 192 193 194 195 196 197 198 199 200 203 204 203 204 205	* EXECUTE * X. NAK X. NAKRET X. DC2RET * NAME * FUNCTION * INPUT * VOLATILE *	GUIT ': EQU LDA AND BNE EQU EQU EQU RTS FULL8 SET F NONE WINDO EQU LDA STA	* MODE MODE MODE MODE MODE MODE MODE UIL V MOL SOCOL WIN PARAMETERS * MOTOP WNDLFT	; ONLY VALID IN BASIC ; IGNORE IF PASCAL ; GET SETUP TO QUIT ; DONE; CALLER WON'T RETURN
CD90: CD90: CD90: CD90: CD90: CD92: CD93: AD CD93: 29 CD93: CD94: CD94: CD94: CD94: CD94: CD98:	20 03 AA 022 50	CD90 04 CD9A CD CD9A CD9A	184 185 186 187 188 189 190 192 193 194 195 196 197 198 200 201 203 204 203 204 203 207	* EXECUTE * X. NAK X. NAKRET X. DC2RET * NAME * FUNCTION * INPUT * VOLATILE *	GUIT ': EQU LDA AND BNE JSR EQU EQU RTS SET F NONE SET F NONE WINDO AC LDA STA STA STA LDA	* MODE *M.PASCAL *.NAKRET GUIT * * ULL SOCOL WIT W PARAMETERS * * * WNDTOP WNDLFT #80	; ONLY VALID IN BASIC ; IGNORE IF PASCAL ; GET SETUP TO QUIT ; DONE; CALLER WON'T RETURN
CD90: CD90: CD90: CD90: CD90: CD90: CD93: 20 CD93: 20 CD93: 20 CD93: CD9	20 03 AA 00 22 20 50 21	CD90 04 CD9A CD CD9A CD9A	184 185 186 187 190 191 192 193 194 195 196 197 198 199 200 203 204 203 204 205	* EXECUTE * X. NAK X. NAKRET X. DC2RET * NAME * FUNCTION * INPUT * VOLATILE *	GUIT ': EQU LDA AND BNE EQU EQU EQU RTS FULL8 SET F NONE WINDO EQU LDA STA	* MODE MODE MODE MODE MODE MODE MODE UIL V MOL SOCOL WIN PARAMETERS * MOTOP WNDLFT	; ONLY VALID IN BASIC ; IGNORE IF PASCAL ; GET SETUP TO QUIT ; DONE; CALLER WON'T RETURN
CD90: CD90: CD90: CD90: AD CD93: AD CD93: AD CD93: AD CD93: CD94: CD94: CD94: CD94: CD94: CD94: CD94: CD94: CD98: CD98: CD98: CD98: CD98: CD98: CD98: A9 CD98: A9 CD97: 85 CD41: A9 CD94: A5 CD43: A9 CD43: A9 CD43: A9	20 03 AA 00 22 20 50 21 18	CD90 04 CD9A CD CD9A CD9A	184 185 186 187 188 189 191 192 193 194 195 194 195 196 200 202 203 206 208	* EXECUTE * X. NAK X. NAKRET X. DC2RET * NAME * FUNCTION * INPUT * VOLATILE *	GUIT ': EQU LDA AND BNE JSR EQU EQU EQU EQU EQU EQU EQU EQU EQU EQU	* MODE * MDDE * MP. PASCAL X. NAKRET GUIT * * 0 ULL BOCOL WIN W PARAMETERS * * * * * * * * * * * * *	; ONLY VALID IN BASIC ; IGNORE IF PASCAL ; GET SETUP TO QUIT ; DONE; CALLER WON'T RETURN
CD90: CD90: CD90: CD90: CD90: CD90: CD93: CD93: CD93: CD93: CD93: CD94: CD94: CD94: CD94: CD98:	20 03 AA 00 22 20 50 21 18	CD90 04 CD9A CD CD9A CD9A	184 185 187 188 187 188 190 191 193 194 197 194 197 200 201 202 203 204 203 204 203 204 203 204 205 206 207 208 209 200 201	* EXECUTE * X. NAK X. NAKRET X. DC2RET * NAME * FUNCTION * INPUT * OULTPUT * VOLATILE * FULLBO	GUIT ': EGU LDA AND BNE JSR EGU EGU EGU EGU EGU EGU LDA STA STA STA STA STA LDA STA LDA	* MODE *M. PASCAL * NAKRET GUIT * * ULL BOCOL WIT W PARAMETERS * * * * * * * * * * * * *	; ONLY VALID IN BASIC ; IGNORE IF PASCAL ; GET SETUP TO QUIT ; DONE; CALLER WON'T RETURN
CD90: CD90: CD90: CD90: AD CD93: 29 CD93: 29 CD93: 20 CD93: 20 CD93: CD CD93: CD CD93: CD CD93: CD CD93: CD93: CD93: CD93: CD93: CD93: CD93: CD93: CD93: A9 CD93: A9	20 03 AA 00 22 20 50 21 18	CD90 04 CD9A CD CD9A CD9A	184 185 186 187 188 187 170 171 172 174 175 174 175 200 201 203 204 203 204 207 203 204 207 203 204 207 200 201 207 200 207 200 201 202 203 204 207 212	* EXECUTE * X. NAK X. NAKRET X. DC2RET * FUNCTION * FUNCTION * INIPUT * VOLATILE * FULLBO	GUIT ': EGU LDA AND BNE SND EGU RTS EGU RTS SET F ULL8 SET F NONE WINDO AC CO EGU LDA STA STA LDA STA STA LDA STA RTS	* MODE *M. PASCAL * NAKRET GUIT * * ULL BOCOL WIT W PARAMETERS * * * * * * * * * * * * *	; ONLY VALID IN BASIC ; IGNORE IF PASCAL ; GET SETUP TO QUIT ; DONE; CALLER WON'T RETURN
CD90: CD90: CD90: CD90: CD90: CD90: CD93: CD93: CD93: CD93: CD93: CD94: CD94: CD94: CD94: CD98:	20 03 AA 00 22 20 50 21 18	CD90 04 CD9A CD CD9A CD9A	184 185 187 188 187 188 187 190 191 193 194 197 197 197 197 197 200 203 204 204 203 204 203 204 203 204 203 204 204 204 204 204 204 204 204 204 204	* EXECUTE * X. NAK X. NAKRET X. DC2RET * NAME * FUNCTION * TUPUT * VOLATILE * FULLBO	GUIT ': EGU LDA AND BNE EGU RTS EGU RTS SET F NONE WINDO AC EGU LDA STA STA STA STA STA RTS	* MODE #M.PASCAL X.NAKRET GUIT * * ULL BOCOL WIT W PARAMETERS * WNDUTP WNDLFT #80 WNDUTH #24 WNDBTM	; ONLY VALID IN BASIC ; IGNORE IF PASCAL ; GET SETUP TO QUIT ; DONE, CALLER WON'T RETURN
CD90: CD90: CD90: CD90: AD CD93: 29 CD93: 29 CD93: 20 CD93: 20 CD93: CD CD93: CD CD93: CD CD93: CD CD93: CD CD93: CD93: CD93: CD93: CD93: CD93: CD93: CD93: CD93: A9 CD93: A9	20 03 AA 00 22 20 50 21 18	CD90 04 CD9A CD CD9A CD9A	184 185 187 188 187 197 197 197 197 197 197 197 197 197 201 202 203 204 203 204 203 204 203 204 203 204 203 204 207 202 203 204 201 202 209 211 212 213	* EXECUTE * X. NAK X. NAKRET X. DC2RET * NAME * FUNCTION * VOLATILE * VOLATILE * VOLATILE * VOLATILE * NAME * ENKCTION	GUIT ': EGU LDA AND BNE EGU EGU EGU EGU EGU EGU LDA STA STA STA STA STA RTS GUIT SCTUB	* MODE MM.PASCAL X.NAKRT GUIT 4 * * ULL BOCOL WIN PARAMETERS * 0 ULL BOCOL WIN W PARAMETERS * 0 MNDTOP MNDLFT * 80 MNDLFT #24 WNDBTM TO AUIT THE	; ONLY VALID IN BASIC ; IGNORE IF PASCAL ; GET SETUP TO QUIT ; DONE, CALLER WON'T RETURN
CD90: CD90: CD90: CD90: CD90: CD90: CD93: CD93: CD93: CD93: CD93: CD94: CD94: CD94: CD94: CD98:	20 03 AA 00 22 20 50 21 18	CD90 04 CD9A CD CD9A CD9A	184 185 187 188 187 197 197 197 197 197 197 197 197 197 201 202 203 204 203 204 203 204 203 204 203 204 203 204 207 202 203 204 201 202 209 211 212 213	* EXECUTE * X. NAK X. NAKRET X. DC2RET * NAME * FUNCTION * VOLATILE * VOLATILE * VOLATILE * VOLATILE * NAME * ENKCTION	GUIT ': EGU LDA AND BNE EGU EGU EGU EGU EGU EGU LDA STA STA STA STA STA RTS GUIT SCTUB	* MODE MM.PASCAL X.NAKRT GUIT 4 * * ULL BOCOL WIN PARAMETERS * 0 ULL BOCOL WIN W PARAMETERS * 0 MNDTOP MNDLFT * 80 MNDLFT #24 WNDBTM TO AUIT THE	; ONLY VALID IN BASIC ; IGNORE IF PASCAL ; GET SETUP TO QUIT ; DONE, CALLER WON'T RETURN
CD90: CD90: CD90: CD90: CD90: CD90: CD92: CD93:	20 03 AA 00 22 20 50 21 18	CD90 04 CD9A CD CD9A CD9A	184 185 186 187 188 187 190 191 192 193 194 197 198 200 201 202 203 204 205 203 204 205 207 208 207 208 207 208 207 208 207 211 212 213 214 215 214 215 214 215 214 215 214 215 214 215 214 215 216 216 217 218 218 219 200 201 202 203 204 205 207 200 201 207 200 207 200 207 200 207 207 200 207 207	* EXECUTE * X. NAK X. NAKRET X. DC2RET * NAME * FUNCTION * UNUT * VOLATILE * FUNCTION * INPUT * UNCTION * UNCTION * UNCTION	GUIT ': EGU LDA AND BNE BNE EGU EGU EGU RTS FULL8 SET F NONE EGU LDA STA STA STA LDA STA STA STA STA STA STA STA STA STA ST	* MODE MM.PASCAL MX.NAKRET GUIT * * ULL BOCOL WIT W PARAMETERS * WNDUCH WNDTOP WNDLFT * 80 WNDDTH WD4 MNDBTM TO GUIT THE NG N0 F09	: ONLY VALID IN BASIC ; IGNORE IF PASCAL : GET SETUP TO GUIT ; DONE, CALLER WON'T RETURN
CD90: CD90: CD90: CD90: AD CD93: 20 CD93: 29 CD95: D0 CD97: 20 CD97: 20 CD97: 20 CD98: CD9	20 03 AA 00 22 20 50 21 18	CD90 04 CD9A CD CD9A CD9A	184 185 186 187 187 187 197 197 198 197 197 200 201 202 203 204 204 203 204 203 204 204 203 204 204 203 204 204 203 204 204 204 204 204 204 204 204 204 204	* EXECUTE * X. NAK X. NAKRET X. DC2RET * NAME * FUNCTION * UNUT * VOLATILE * FUNCTION * INPUT * UNCTION * UNCTION * UNCTION	GUIT ': EGU LDA AND BNE BNE EGU EGU EGU RTS FULL8 SET F NONE EGU LDA STA STA STA LDA STA STA STA STA STA STA STA STA STA ST	* MODE MM.PASCAL MX.NAKRET GUIT * * ULL BOCOL WIT W PARAMETERS * WNDUCH WNDTOP WNDLFT * 80 WNDDTH WD4 MNDBTM TO GUIT THE NG N0 F09	: ONLY VALID IN BASIC ; IGNORE IF PASCAL : GET SETUP TO GUIT ; DONE, CALLER WON'T RETURN
CD90: CD90: CD90: CD90: CD90: CD90: AD CD93: 29 CD93: 20 CD93: 29 CD93: 20 CD93: 29 CD93: 20 CD93: CD93: CD98: CD98: CD98: CD98: CD98: CD98: CD98: CD98: CD98: 49 CD98: 40 CD98: 40 CD9	20 03 AA 00 22 20 50 21 18	CD90 04 CD9A CD CD9A CD9A	184 185 186 187 187 187 190 191 192 193 194 197 198 197 198 200 203 203 203 203 203 203 203 203 203	* EXECUTE * X. NAK X. NAK X. NAK X. NAK X. DC2RET * NAME * FUNCTION * INPUT * UTPUT * UTPUT * FULLSO * NAME * FUNCTION * INPUT * UNPUT * UN	GUIT ': EGU LDA AND BNE BNE EGU EGU EGU RTS FULL8 SET F NONE EGU LDA STA STA STA LDA STA STA STA STA STA STA STA STA STA ST	* MODE MM.PASCAL X.NAKRT GUIT 4 * * ULL BOCOL WIN PARAMETERS * 0 ULL BOCOL WIN W PARAMETERS * 0 MNDTOP MNDLFT * 80 MNDLFT #24 WNDBTM TO AUIT THE	: ONLY VALID IN BASIC ; IGNORE IF PASCAL : GET SETUP TO GUIT ; DONE, CALLER WON'T RETURN
CD90: CD90: CD90: CD90: CD90: CD92: CD93:	20 03 AA 00 22 20 50 21 18	CD90 04 CD9A CD CD9A CD9A	184 185 186 187 187 187 197 197 197 197 197 197 197 197 197 19	* EXECUTE * X. NAK X. NAKRET X. DC2RET * NAME * FUNCTION * UNUT * VOLATILE * FUNCTION * INPUT * UNCTION * UNCTION * UNCTION	GUIT ': EGU LDA AND BNE BNE EGU EGU EGU RTS FULL8 SET F NONE EGU LDA STA STA STA LDA STA STA STA STA STA STA STA STA STA ST	* MODE MM.PASCAL MX.NAKRET GUIT * * ULL BOCOL WIT W PARAMETERS * WNDUCH WNDTOP WNDLFT * 80 WNDDTH WD4 MNDBTM TO GUIT THE NG N0 F09	: ONLY VALID IN BASIC ; IGNORE IF PASCAL : GET SETUP TO GUIT ; DONE, CALLER WON'T RETURN
CD90: CD90: CD90: CD90: CD90: CD90: AD CD95: D09: AD CD93:29 CD98:	20 03 AA 00 22 20 50 21 18	CD90 CD9A CD9A CD9A CD9A CD9A	184 185 186 187 187 197 197 197 197 197 197 197 197 197 19	* EXECUTE * EXECUTE * X. NAK X. NAKRET X. DC2RET * NAME * FUNCTION * INPUT * UTPUT * UTPUT * UTPUT * FULLSO * NAME * FUNCTION * INPUT * UTPUT * UTPUT	GUIT ': EQU LDA LDA BNE BNE EQU EQU EQU EQU EQU EQU EQU EQU LDA STA STA STA STA STA STA STA STA STA ST	* MDDE MDDE MP.PASCAL X.NAKRT GUIT * ULL BOCOL WIT PARAMETERS * O ULL BOCOL WIT W PARAMETERS * O UNDUFT #0 HNDUFT #	: ONLY VALID IN BASIC ; IGNORE IF PASCAL : GET SETUP TO GUIT ; DONE, CALLER WON'T RETURN
CD90: CD90: CD90: CD90: AD CD93: 29 CD93: 29 CD93: 29 CD93: CD CD93: 20 CD93: CD CD93: CD CD93: CD CD93: CD CD93: CD98:	00 00 20 50 21 18 23	CD90 04 CD9A CD CD9A CD9A	184 185 186 187 187 197 197 197 197 197 197 197 197 197 19	* EXECUTE * EXECUTE * X. NAK X. NAKRET X. DC2RET * NAME * FUNCTION * INPUT * UTPUT * UTPUT * UTPUT * FULLSO * NAME * FUNCTION * INPUT * UTPUT * UTPUT	GUIT ': EQU LDA AND BNE EQU EQU EQU EQU EQU EQU AC STA STA LDA STA LDA STA LDA STA LDA STA LDA STA STA LDA STA STA LDA STA EQU EQU EQU EQU EQU EQU EQU	* * MODE * MODE * MODE * MODE * MODE * MODE * * * * * * * * * * * * * * * * * * *	; ONLY VALID IN BASIC ; IGNORE IF PASCAL ; GET SETUP TO QUIT ; DONE; CALLER WON'T RETURN NDOW CARD
CD90: CD90: CD90: CD90: CD90: CD90: CD93: 29 CD93: 20 CD93: 20 CD93: 20 CD93: 20 CD93: 20 CD93: 20 CD93: 20 CD93: CD93: CD98: CD98: CD98: CD98: CD98: CD98: CD98: CD98: 49 CD98: 40 CD98: 40 CD9	00 22 20 20 21 23 00 22 20 21 23 00 22 00 22 00 22 00 22 00 20 20 20 20	CD90 CD9A CD9A CD9A CD9A CD9A	184 185 187 188 189 197 198 197 198 197 193 194 200 201 202 203 204 207 202 203 204 207 201 211 213 214 213 214 213 214 213 214 213 214 213 214 203 207 213 214 215 216 217 217 218 207 217 218 219 219 219 211 219 211 219 219	* EXECUTE * EXECUTE * X. NAK X. NAKRET X. DC2RET * NAME * FUNCTION * INPUT * UTPUT * UTPUT * UTPUT * FULLSO * NAME * FUNCTION * INPUT * UTPUT * UTPUT	GUIT ': EQU LDA LDA BNE BNE EQU EQU EQU EQU EQU EQU EQU EQU LDA STA STA STA STA STA STA STA STA STA ST	* * MODE * MODE * MODE * MODE * MODE * MODE * * * * * * * * * * * * * * * * * * *	: ONLY VALID IN BASIC ; IGNORE IF PASCAL : GET SETUP TO GUIT ; DONE, CALLER WON'T RETURN
CD90: CD90: CD90: CD90: AD CD93: 20 CD93: 20 CD93: 20 CD93: 20 CD93: CD93: CD9	00 22 20 20 20 20 20 20 20	CD90 CD9A CD9A CD9A CD9A CD9A	184 185 186 187 188 189 197 197 197 197 197 197 197 19	* EXECUTE * EXECUTE * X. NAK X. NAKRET X. DC2RET * NAME * FUNCTION * INPUT * UTPUT * UTPUT * UTPUT * FULLSO * NAME * FUNCTION * INPUT * UTPUT * UTPUT	GUIT ': EQU LDA LDA BNE BNE EQU EQU EQU EQU EQU EQU EQU EQU LDA STA STA STA STA STA STA STA STA STA ST	* * * * * * * * * * * * * * * * * * *	; ONLY VALID IN BASIC ; IGNORE IF PASCAL ; GET SETUP TO QUIT ; DONE; CALLER WON'T RETURN NDOW CARD
CD90: CD90: CD90: CD90: CD90: CD90: CD93:	00 20 20 20 20 20 20 20 20 20 20 20 20 2	CD90 CD9A CD9A CD9A CD9A CD9A	184 185 186 187 188 189 197 193 194 197 193 194 201 201 201 201 201 201 201 201	* EXECUTE * EXECUTE * X. NAK X. NAKRET X. DC2RET * NAME * FUNCTION * INPUT * UTPUT * UTPUT * UTPUT * FULLSO * NAME * FUNCTION * INPUT * UTPUT * UTPUT	GUIT ': EGU LDA AND BNE EGU EGU EGU EGU EGU AC STA STA LDA EGU LDA STA STA STA LDA EGU LDA STA STA LDA EGU LDA	* * * * * * * * * * * * * * * * * * *	; ONLY VALID IN BASIC ; IGNORE IF PASCAL ; GET SETUP TO QUIT ; DONE; CALLER WON'T RETURN NDOW CARD
CD90: CD90: CD90: CD90: CD90: AD CD95: D00: AD CD95: D07: CD93: CD94: CD93: CD94: CD	002200 2002200 200220 200220 200220 200220 200220 200220 200220 200220 200220 200220 200220 200220 200220 200220 200200	CD90 CD9A CD9A CD9A CD9A CD9A	1844 1855 1867 1887 1887 1887 1897 1973 1974 1975 1974 1975 2000 2012 2023 2024 2032 2045 2032 2045 2072 2088 2072 2088 2072 2012 212 213 212 213 214 215 215 216 217 212 212 213 212 212 212 212 212 212 212	* EXECUTE * EXECUTE * X. NAK X. NAKRET X. DC2RET * NAME * FUNCTION * INPUT * UTPUT * UTPUT * UTPUT * FULLSO * NAME * FUNCTION * INPUT * UTPUT * UTPUT	GUIT ': EQUIT ': EQUIT ': EQUIT ': EQUIT EQUIT EQUIT EQUIT EQUIT EQUIT EQUIT EQUIT EQUIT STA STA STA STA STA STA STA STA STA STA	* MDDE MDDE MP.PASCAL X.NAKRT GUIT V VLL BOCOL WI V PARAMETERS V O ULL BOCOL WI V PARAMETERS V O ULL BOCOL WI V V NDU FT V BO V ULL BO V V ULL BO V V V V V V V V V V V V V V V V V V	; ONLY VALID IN BASIC ; IGNORE IF PASCAL ; GET SETUP TO QUIT ; DONE; CALLER WON'T RETURN NDOW CARD
CD90: CD90: CD90: CD90: AD CD90: AD CD93: 20 CD93: 20 CD93: 20 CD93: 20 CD93: CD93: CD93: 20 CD93: CD93: SC03: A9 CD93: CD93: SC03: A9 CD93: A9 CD93: A9 CD93: A9 CD94: CD3A: CDAA:	002 200 200 200 200 200 200 200 200 200	CD90 CD9A CD9A CD9A CD9A CD9A	184 185 186 187 188 189 197 193 194 197 193 194 201 201 201 201 201 201 201 201	* EXECUTE * EXECUTE * X. NAK X. NAKRET X. DC2RET * NAME * FUNCTION * INPUT * UTPUT * UTPUT * UTPUT * FULLSO * NAME * FUNCTION * INPUT * UTPUT * UTPUT	GUIT ': EGU LDA AND BNE EGU EGU EGU EGU EGU AC STA STA LDA EGU LDA STA STA STA LDA EGU LDA STA STA LDA EGU LDA	* * * * * * * * * * * * * * * * * * *	; ONLY VALID IN BASIC ; IGNORE IF PASCAL ; GET SETUP TO QUIT ; DONE; CALLER WON'T RETURN NDOW CARD

CDBB: 2C 1F CO CDBB: 10 03 CDCO CDBD: 20 DB CD CDCO:	230 231 232 233 *	BIT BPL JSR	RD80VID QUIT2 SCRN84	;WHAT WIDTH? ;=>NG CVT NEEDED IF 40 ;CONVERT 40>80
	234 QUIT2	EQU		
CDC0: A9 17	235	LDA	#23	VTAB TO THE
CDC2: BD FB 05	236	STA	DURCV	3 BOTTOM LINE
CDC5: 20 51 CB	237	JSR	BASCALC	
CDCB: A9 00	238	LDA	#O DURCH	AND PLACE CURSOR
CDCA: BD 7B 05 CDCD: BD OE CO	239 240	STA	CL BAL TCHAR	AT LEFT SIDE
CDDO: A9 FF	241	LDA	#\$FF	DESTROY THE
CDD2: BD FB 04	242	STA	MUDE	MODE BYTE
CDD5: 20 93 FE	243	JSR	SETVID	; PR#O
CDD8: 4C 89 FE	244 245	JMP	SETKBD	RETURN VIA INHO (UGH!)
CDDB	246 * NAME :	SCRNE	4	
CDDB:	247 * FUNCTION:	CONVE	RT BOVID>4	QVID
CDDB:	249 * INPUT : 249 * OUTPUT :	NONE		
CDDB: CDDB:	249 * OUTPUT : 250 * VOLATILE:	NONE	COLOTERS	
CDDB:	251 * NOTE :	USES	'BAS2H/L' AS	TEMPS
CDDB:	252			
CODB:	253 *		17	
CDDB: CDDB CDDB: AD FB 05	254 SCRN84 255	EQU	* OURCV	SAVE CURRENT
CDDE: 48	256	PHA	donev	TARVE CORRERT
CDDF: AD 7B 05	257	LDA	OURCH	; SETTINGS
CDE2: 48	258	PHA		
CDE3: CDE3: A9 17	257 * 260	LDA	#23	
CDE5: 85 2A	261	STA	BAS2L	USE AS A TEMP
CDE7: BD 01 CO	262	STA	SETBOCOL	
CDEA: A5 2A	263 SCR40	LDA	BAS2L	AND THE OTHER PROPERTY AND THE ADDRESS OF
CDEC: 20 54 CB CDEF: 20 0A CE	264 265	JSR JSR	BASCALCZ	; BEGIN AT BOTTOM AND WORK UP ; DO THIS LINE
		DEC	BAS2L	DU THIS LINE
CDF4: 30 0B CE01	267	BMI	SCR40RFT	;=>DONE (HIT TOP)
UDID. EG IM UU	200	BIT	RDTEXT	ARE WE IN MIDEXMODE?
CDF9: 30 EF CDEA	269 270	BMI LDA	SCRAU	=>ND, DD ENTIRE SCREEN
CDFB: A5 2A CDFD: C7 14	271	CMP	BAS2L #20	; IF SO, ONLY DO BOTTOM ; FOUR (4) LINES OF WINDOW
CDFF: BO E9 CDEA	272	BCS	SCR40	, took (4) Eineo of Winbow
CEO1: CEO1	273 SCR40RET	EQU	#	
CE01: 80 00 CO	274	STA	CLRBOCOL	
CE04: BD OC CO CE07: 4C 58 CE	275 276	STA	CLR80VID SCRNRET	RETURN VIA SIMILAR CODE
CEOA:	277 *	U	SUMMEN	THE FURTH VIEL OF THE ENDE
CEOA: CEOA		EQU	*	
CEOA: 08	279	PHP SEI		LOCK IRQ WHILE
CEOB: 78 CEOC: AO 28	280 281	LDY	#40	; SCREENHOLES ARE WRONG
CEOE: 84 28	282	STY	BAS2H	
CE10: 2C 54 CO	283	BIT	TXTPAGE1	
CE13: 20 22 CE	284 ATEFOR1	JSR	GET84	
CE16:2C 55 CO CE19:20 22 CE	285 286	BIT JSR	TXTPAGE2 GET84	
CE1C: A4 2B	287	LDY	BAS2H	;DONE? ;=>ND; DD WHOLE LINE
CE1E: DO F3 CE13		BNE	ATEFOR 1	
CE20: 28	289	PLP		RESTORE IRG NOW
CE21:60 CE22:	290 291 *	RTS		
CE22: C6 2B	292 GET84	DEC	BAS2H	
CE24: A5 28	293	LDA	BA52H	
CE26: 4A CE27: A8	294 295	LSR	A	
CE28: 01 28	295	LDA	(BASL), Y	
CE2A: A4 2B	297	LDY	BAS2H	
CE2C: 2C 54 CO	298	BIT	TXTPAGE1	
CE2F: 91 28	299 300	STA	(BASL), Y	
CE31: 60 CE32:	301			
CE32:	302 # NAME ·	SCRNA	8	
CE32:	303 * FUNCTION 304 * INPUT 305 * OUTPUT	CONVE	RT 40VID>8	OVID
CE32: CE32:	304 * INPUT :	NONE		
CE32:	306 * VOLATILE:	ALL R	EGISTERS	
CE32:	307 * NOTE :	USES	'BAS2H/L' AS	TEMPS
CE32:	308			
CE32: CE32: CE32	309 * 310 SCRN48	EQU	4	
CE32: AD FB 05	310 SCRN48 311	LDA	DURCV	SAVE CV
CE35: 48	312	PHA		1.2010- 21
CE36: AD 78 05	313	LDA	DURCH	AND CH
CE39: 48 CE3A:	314 315 *	PHA		
CEBA: A9 17	315 *	LDA	#23	
CE3C: 85 2A	317	STA	BAS2L	USE AS A TEMP
CEBE: A5 2A	318 SCR80	LDA	BAS2L	
CE40: 20 54 CB CE43: 20 63 CE	317 320	JSR JSR	BASCALCZ FORATE	;BEGIN AT BOTTOM AND WORK UP ;DO THIS LINE
		DEC	BASEL	DO HING EINE
CE46: C6 2A	321			

CE48:30 0B CE5 CE4A:2C 1A CO CE4D:30 EF CE3 CE4F:A5 2A CE51:C7 14 CE53:B0 E7 CE3 CE55:	323 E 324 325 326	BMI BIT BMI LDA CMP BCS	SCRBORET RDTEXT SCRBO BAS2L #20 SCRBO	:=>DONE (HIT TOP) ;ARE WE IN MIXEDMODE? ;MO, DO FULL SCREEN ;IF SG, DNLY DO BOTTOM ;FOUR (4) LINES OF WINDOW
CE55: CE55 CE55: 8D OD CO	327 SCREORET	EQU	* SETBOVID	DICOLAY IN COMODE
CE58: CE58	3 331 SCRNRET	STA EQU	*	; DISPLAY IN 80-MODE ; USED BY SCRN84
CE58:48 CE59:8D 7B 05	332 333	PLA	OURCH	CH AND
CE5C: 48 CE5D: 80 FB 05	334 335	PLA	DURCV) CV
CE60: 4C 51 CB	336	JMP	BASCALC	RETURN VIA BABCALC (UGH!)
CE63: CE63:	337 * 338 *			
	3 339 FORATE 340	EGU	*	
CE64: 78	341	SEI		DON'T ALLOW IRQ WHILE
CE65: A0 00 CE67: 84 29	342 343	LDY STY	#0 BAS2H	
CE69: 8C 01 CO	344	STY	SETBOCOL	
CE6C: 2C 54 CO CE6F: B1 28	345 346 FORATE1	BIT	TXTPAGE1 (BASL), Y	
CE71:2C 55 CO CE74:20 A3 CE	347	BIT	TXTPAGE2	
CE74: 20 A3 CE CE77: 2C 54 CO	348 349	JSR BIT	TXTPAGE1	
CE7A: B1 28	350	LDA	(BASL), Y	
CE7C: 20 A3 CE CE7F: C0 28	351 352	JSR CPY	DO48 #40	
CE81: 90 EC CE6F CE83:	- 353 354 *	BCC	FORATE1	
CE83: 20 91 CE	355	JSR	CLRHALF	CLEAR RIGHT HALF
CE86: 2C 55 CO CE89: 20 91 CE	356 357	BIT JSR	TXTPAGE2 CLRHALF	OF BOTH PAGES
CE8C: 2C 54 CO	358	BIT	TXTPAGE1	
CEBF: 28 CE90: 60	359 360	PLP RTS		; OK TO ALLOW IRG NOW
CE91: CE91: CE91	361 *	EQU		
CE91: AO 14	362 CLRHALF	LDY	* * 20	
CE93: A9 A0	364 365	LDA BIT	#' INVFLG	, WHICH MODE?
CE95: 24 32 CE97: 30 02 CE98	366	BMI	CLRHALF2	;=>NORMAL
CE99: 29 7F CE98: CE98	367 368 CLRHALF2	AND	#\$7F	; INVERSE
CE98: 91 28	369	STA	(BASL), Y	STUFF THE BLANK
CE9D: C8 CE9E: C0 28	370 371	INY CPY	#40	
CEAO: DO F9 CE9E	372	BNE	CLRHALF2	
CEA2: 60 CEA3:	373 374 *	RTS		
CEA3: 48 CEA4: 98	375 DO48 376	PHA TYA		
CEA5: 4A	377	LSR	A	
CEA6: A8 CEA7: 68	378 379	TAY		
CEA8: 91 28	380	STA	(BASL), Y	
CEAA: E6 2B CEAC: A4 2B	381 382	INC LDY	BAS2H BAS2H	
CEAE: 60	383	RTS		
CEAF: CEAF:	14 2		DE SUBS3	
CEAF: CEAF:	3 * NAME :	SETCH		
CEAF:	4 * FUNCTION: 5 * INPUT 6 * OUTPUT	AC=CH	VALUE	
CEAF: CEAF:	6 * OUTPUT : 7 * VOLATILE:	NOTHI	, CH MOD 40 NG	
CEAF: CEAF:	7 * VOLATILE: B * CALLS 9	NOTHI		
CEAF:	10 *			
CEAF: CEAF CEAF: 8D 7B 05	11 SETCH 12	EQU	* OURCH	STUFF OURCH
CEB2: 85 24	13	STA	CH	STUFF IN CASE WE'RE 40 MODE
CEB4: 8D 7B 04 CEB7: 2C 1F CO	14 15	STA	DLDCH RDBOVID	IN BO-MODE?
CEBA: 10 1D CED9	16	BPL	SETCHRTS	;=>NO, DONE
CEBC: CEBC:	17 * 18 * IF WE'RE	NEAR T	HE END OF DU	R
CEBC:	17 * 80COL LI	NE MO	VE CH UP. IF D AT ZERO	NOT,
CEBC: CEBC:	20 * LEAVE CH 21 *	PINNE	D AT ZERO	
CEBC: A9 00	22	LDA	#0	PIN CH AT ZERO
CEBE: 85 24 CECO: 8D 7B 04	23 24	STA	CH OLDCH	REMEMBER THE SETTING
CEC3: A5 21 CEC5: 38	25	LDA	WNDWDTH	CHECK IF NEAR THE END
CEC6: ED 78 05	26 27	SEC	OURCH	GET ABS CH
CEC9: C9 OB CEC5: BO OC CED9	28	CMP BCS	#8 SETCHRTS	NEAR THE END?
CECD: 85 24	30	STA	CH	YES, MOVE CH UP NEAR RIGHT

CECF: A9 28 CED1: 38 CED2: E5 24 CED4: 85 24 CED4: 85 7B 04	31 32 33 34 35	LDA #40 SEC SBC CH STA CH STA DLDCH	; BASIC WILL SEE THAT NOW ; REMEMBER THE SETTING
CED9: CED9: CED9 CED9: AD 7B 05 CEDC: 60	36 * 37 SETCHRTS 38 39	EQU * LDA OURCH RTS	RESTORE AC
CEDD: CEDD: CEDD: CEDD: CEDD: CEDD: CEDD: CEDD:	43 * INPUT : 44 * OUTPUT : 45 * VOLATILE: 46 * CALLS :	INVERT INVERT CHAR AT NOTHING CHAR AT CH/CV NOTHING PICK, STORCHAR	INVERTED
CEDD: CEDD: CEDD: CEDD CEDD: 48 CEDE: 98 CEDF: 48	47 48 * 49 INVER1 50 51 52	EQU * PHA TYA PHA	; SAVE AC ; AND Y
CEE0: AC 7B 05 CEE3: 20 01 CF CEE4: 49 80 CEE8: 2C 00 CF CEE8: 20 06 CF CEEE: 68 CEEF: A8 CEEF: A8	53 55 55 56 57 58 59 59 60	LDY OURCH JSR PICK EOR #\$E00 BIT SEV JSR SCREENIT PLA TAY PLA	;GET CH ;GET CHARACTER ;FLIP INVERSE/NORMAL ;PUT DIRECTLY BACK ;ONTO SCREEN ;RESTORE Y ; AND AC
CEF1:60 CEF2: CEF2: CEF2: CEF2: CEF2: CEF2: CEF2: CEF2: CEF2: CEF2:	64 * FUNCTION: 65 * INPUT :	CHAR ON SCREEN	
CEF2: CEF2: CEF2: CEF2: CEF2 CEF2:48 CEF3:2432 CEF5:3002 CEF7	70 71 * 72 STORCHAR 73 74 75	EQU * PHA BIT INVFLG BMI STOR2	; SAVE AC ; NORMAL OR INVERSE? ; =>NORMAL
CEF7: 49 80 CEF9: CEF9 CEF9: 2C 00 CF CEFC: 20 06 CF CEFF: 68 CF00: 60	76 77 STOR2 78 79 80 81 SEV	EQU * BIT SEV JSR SCREENIT PLA RTS	; INVERSE ; V SET FOR STORE ;=>DD IT! ;RESTORE AC
CF01: CF01: CF01: CF01: CF01: CF01: CF01: CF01:	82 83 * NAME : 84 * FUNCTION: 85 * INPUT : 86 * OUTPUT : 87 * VOLATILE: 88 * CALLS : 89	PICK GET A CHAR FRO Y=CH POSITION AC=CHARACTER NOTHING	M SCREEN
CF01: CF01: CF01 CF01:B8 CF02:20 06 CF CF05:60 CF06:	90 * 91 PICK 92 93 94 95	EQU * CLV JSR SCREENIT RTS	;V CLEAR FOR PICK ;DO IT!
CF06: CF06: CF06: CF06: CF06: CF06: CF06: CF06: CF06: CF06: CF06: CF06: CF06:	96 * NAME : 97 * FUNCTION: 98 * INPUT : 99 * : 100 * : 101 * : 102 * OUTPUT : 103 * VOLATILE:	STORE OR PICK V CLR FOR PICK V SET FOR STOR AC=CHAR FOR ST Y=CH POSITION AC=CHAR (PICK)	Ë ORE
CF06:	106 * 107 SCREENIT 108 109 110 * AVDID CHAN	EQU * STY YSAV1 PHA	SAVE Y SAVE CHARACTER IF STORING
CF07: AD 1F C0 CF0C: 10 32 CF40 CF0E: CF0E: CF0E:	111 112 113 * 114 * 80-COLUMN 115 *	LDA RDBOVID BPL SCRN40	↓WHAT DISPLAY MODE? ↓=>40-COL MODE
CFOE: A5 1F CF10: 4A CF11: A8 CF12: 70 16 CF2A CF14:	115 * 116 117 118 119 120 *	LDA YSAV1 LSR A TAY BVS STORBO	;GET CURSOR HORIZ ;DIVIDE BY TWO FOR PAGE ;CH TO YREG ;=>GONNA STORE THE CHAR
CF14:	121 * 80-COL PI	CK:	

CF14: CF14:08	122		PHP		LOCK INTERRUPTS WHILE
CF15:78 CF16:AD 55 CO	124		SET		SCREENHOLES ARE WRONG
CF16: AD 55 CO	125		LDA	TXTPAGE2	ASSUME PAGE 2 (EVENS)
CF19:90 03 CF1 CF1B:AD 54 CO	126		BCC	SCRN2 TXTPAGE1	;=>IT IS ;ODDS GO TO PAGE1
CF1E: CF1	129	SCRN2	EQU	*	TODDS GO TO PAGET
CF1E:01 28	129	0071110	LDA	(BASL), Y	PICK THE CHARACTER
CF20:A8 CF21:AD 54 CO	130		TAY	TXTPAGE 1	;HOLD CHAR TEMPORARILY ;RESTORE PAGE1
CF24:28	132		PLP	TXTPAGET	AND ALLOW IRQ AGAIN
CF25: 68	133		PLA		TRASH SAVED AC
CF26: 98 CF27: 48	134 135		TYA		MAKE CHAR GET RESTORED TO AC
CF28: 50 24 CF4	135		BVC	STPKEXIT	=>DONE (ALWAYS TAKEN)
CE24	137	÷			
CF2A: CF2 CF2A: 68	138		EQU	*	RESTORE CHARACTER
CF2B: 48	140		PHA		(LEAVE ON STACK)
CF2C: 08	141		PHP		LOCK INTERRUPTS WHILE
CF2D: 78 CF2E: 48	142 143		SEI PHA		LOCK INTERRUPTS WHILE THE SCREENHOLES ARE WRONG HOLD THE CHAR TEMPORARILY
CF2F: AD 55 CO	144		LDA	TXTPAGE2	; ASSUME PAGE2 (EVENS)
CF32: 90 03 CF3	143 144 145 146		BCC	SCRN3	;=>IT IS ;ODDS GO TO PAGE1
CF34: AD 54 CO CF37: CF3	146	COND	LDA EQU	TXTPAGE1	ODDS GO TO PAGE1
CF37: 68	147	36443	PLA	*	GET CHAR TO BE STORED
CE38:91 28	149		STA	(BASL) / Y	STUFF ONTO SCREEN
CF3A: AD 54 CO	150		LDA	TXTPAGE1	RESTORE PAGE1
CF3D: 28 CF3E: 70 OE CF4	121		PLP BVS	STPKEXIT	; AND ALLOW IRQ AGAIN ;=>DONE (ALWAYS TAKEN)
CF40:	153	*	-		
CF40:	154	* 40-COLUMN	MODE:		
CF40: CF40: CF44	155	* SCRN40	FQU	*	
CF40: A4 1F	157		LDY	YSAV1	GET CURSOR HORIZ
CF42: 70 06 CF4			BVS	STOR40	; =>STORE IT
CF44:68 CF45:B1 28	159		PLA	(BASL), Y	TRASH SAVED CHAR
CF47:48	161		PHA	(BH3C/) (PICK THE CHARACTER
CF48: 50 04 CF4	162		BVC	STPKEXIT	DONE (ALWAYS TAKEN)
CF4A: CF4A: CF4	163		EQU	*	
CF4A: 68	165	5101040	PLA		GET THE CHARACTER
CF4B: 48	166		PHA		(LEAVE ON STACK)
CF4C:91 28	167		STA	(BASL), Y	STUFF ONTO SCREEN
CF4E: CF4E: CF4	168	* STPKEXIT		(BASL), Y	STUFF ONTO SCREEN
CF4E: CF4E: CF4I CF4E:68	168 169 170	STPKEXIT	EQU	*	STUFF ONTO SCREEN
CF4E: CF4E: CF4H CF4E:68 CF4F:64 1F	168 169 170 171	STPKEXIT	EQU PLA LDY	(BASL), Y * YSAV1	STUFF ONTO SCREEN
CF4E: CF4E: CF44 CF4E: 68 CF4F: 68 CF4F: 60 CF51: 60 CF52:	168 169 170 171 172	STPKEXIT	EQU PLA LDY RTS	¥ YSAV1	STUFF ONTO SCREEN
CF4E: CF4 CF4E: CF4 CF4E: 68 CF4F: A4 1F CF51: 60 CF52: CF52:	168 169 170 171 172 173 174	STPKEXIT	EQU PLA LDY RTS ESCON	¥ YSAV1	;STUFF ONTO SCREEN ;RESTORE AC ;RESTORE Y
CF4E: CF4E: CF4 CF4E:68 CF4F:64 CF51:60 CF52: CF52: CF52: CF52:	168 169 170 171 172 173 174 175	STPKEXIT * NAME : * FUNCTION;	EQU PLA LDY RTS ESCON	* YSAV1	, STUFF ONTO SCREEN ; RESTORE AC , RESTORE Y
CF4E: CF4E: CF4E:68 CF4F:A4 1F CF51:60 CF52: CF52: CF52: CF52: CF52: CF52:	168 169 170 171 172 173 174 175 176	STPKEXIT * NAME : * FUNCTION: * INPUT : * OUTPUT	EQU PLA LDY RTS ESCON TURN (NONE	* YSAV1 DN 'ESCAPE'	, STUFF ONTO SCREEN ; RESTORE AC , RESTORE Y
CF4E: CF4 CF4E: CF4 CF4E: 68 CF4F: A4 1F CF51: 60 CF52: CF52: CF52: CF52: CF52: CF52: CF52: CF52:	168 169 170 171 172 173 174 175 176	STPKEXIT * NAME : * FUNCTION: * INPUT : * OUTPUT	EQU PLA LDY RTS ESCON TURN (NONE	* YSAV1 DN 'ESCAPE'	, STUFF ONTO SCREEN ; RESTORE AC , RESTORE Y
CF4E: CF4E: CF4E:68 CF4F:A4 1F CF51:60 CF52: CF52: CF52: CF52: CF52: CF52:	168 169 170 171 172 173 174 175 176	STPKEXIT * NAME : * FUNCTION: * INPUT : * OUTPUT	EQU PLA LDY RTS ESCON TURN (NONE	* YSAV1 DN 'ESCAPE'	, STUFF ONTO SCREEN ; RESTORE AC , RESTORE Y
CF4E: CF4 CF4E: CF4 CF4E: 68 CF4F: A4 1F CF51: 60 CF52: CF52: CF52: CF52: CF52: CF52: CF52: CF52: CF52: CF52: CF52: CF52: CF52: CF52: CF52:	168 169 170 171 172 173 174 175 176 177 178 179 180 181	STPKEXIT * NAME : * FUNCTION: * INPUT : * OUTPUT : * VOLATELE: * CALLS : *	EQU PLA LDY RTS ESCON TURN (NONE 'CHAR NOTHIN PICK, S	* YSAV1 DN 'ESCAPE'	, STUFF ONTO SCREEN ; RESTORE AC , RESTORE Y
CF4E: CF4E: CF4I CF4E: CF4I CF4F: A4 1F CF51: 60 CF52: CF52	168 169 170 171 172 173 174 175 176 177 178 179 180 181	STPKEXIT * NAME : * FUNCTION: * UVTPUT : * OUTPUT : * CALLS : * ESCON	EQU PLA LDY RTS ESCON TURN (NONE 'CHAR NOTHIN PICK.S EQU	* YSAV1 DN 'ESCAPE'	, STUFF ONTO SCREEN ; RESTORE AC , RESTORE Y CURSOR :HAR
CF4E: CF4 CF4E: CF4 CF4E: 68 CF4F: A4 1F CF51: 60 CF52: CF52: CF52: CF52: CF52: CF52: CF52: CF52: CF52: CF52: CF52: CF52: CF52: CF52: CF52:	168 169 170 171 172 173 174 175 176 177 178 179 180 181	STPKEXIT * NAME : * FUNCTION: * UVTPUT : * OUTPUT : * CALLS : * ESCON	EQU PLA LDY RTS ESCON TURN (NONE 'CHAR NOTHIN PICK, S	* YSAV1 DN 'ESCAPE' '=ORIGINAL C NG STORCHAR	, STUFF ONTO SCREEN , RESTORE AC , RESTORE Y CURSOR CHAR , SAVE AC
CF4E: CF4E: CF4I CF4E: 68 CF4F: A4 1F CF51: 60 CF52: C	168 169 170 171 172 173 174 175 176 177 178 177 181 181 182 183 184 185	STPKEXIT * NAME : * FUNCTION: * INPUT : * OULTPUT : * VOLATELE: * ESCON	EQU PLA LDY RTS ESCON TURN (NONE 'CHAR NOTHIN PICK.S EQU PHA TYA	* YSAV1 DN 'ESCAPE' '=DRIGINAL C IG STORCHAR *	, STUFF ONTO SCREEN ; RESTORE AC ; RESTORE Y CURSOR CHAR ; SAVE AC ; AND Y
CF4E: CF4E: CF4I CF4E: CF4I CF4F: A4 1F CF51: 60 CF52: CF52	168 169 170 171 172 173 174 175 176 177 178 177 180 181 180 181 182 182 183 184 185	STPKEXIT * NAME : FUNCTION: * FUNCTION: * UOUTPUT : * UOLATELE: * CALLS : * ESCON	EQU PLA LDY RTS ESCON TURN (NONE 'CHAR NOTHI PICK.S EQU PHA TYA PHA	* YSAV1 DN 'ESCAPE' '=DRIGINAL C IG STORCHAR *	, STUFF ONTO SCREEN ; RESTORE AC ; RESTORE Y CURSOR CHAR ; SAVE AC ; AND Y
CF4E: CF4E: CF4E: CF4F: CF4F: CF52: CF54: CF	168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 182 182 184 185 184	STPKEXIT * NAME : * FUNCTION: * INPUT : * OUTPUT : * CALLS : * CALLS : * * ESCON	EGU PLA LDY RTS ESCON TURN (NONE 'CHAR NOTHI PICK.S EGU PHA TYA PHA LDY SR	* YSAV1 DN 'ESCAPE' '=DRIGINAL C IG STORCHAR *	, STUFF ONTO SCREEN ; RESTORE AC ; RESTORE Y CURSOR CHAR ; SAVE AC ; AND Y
CF4E: CF4E: CF4E: CF4I: CF4F: A1 CF52: CF52: CF53: 78 CF53: 78 CF58: 01 CF58: 01 CF58: 01 CF58: 01	168 149 170 171 172 173 174 175 176 177 178 177 178 179 180 181 185 188 185 186 187 188 189	STPKEXIT * NAME : * FUNCTION: * INPUT : * OUTPUT : * CALLS : * CALLS : * * ESCON	EGU PLA LDY RTS ESCON TURN (NONE 'CHAR NOTHI PICK.S EGU PHA TYA PHA LDY SR	* YSAV1 DN 'ESCAPE' '=DRIGINAL C IG STORCHAR *	, STUFF ONTO SCREEN ; RESTORE AC ; RESTORE Y CURSOR CHAR ; SAVE AC ; AND Y
CF4E: CF4E: CF44 CF4E: 68 CF52: C	168 169 170 171 172 172 174 175 174 175 176 177 181 181 182 182 183 184 185 184 185 186 187 188 189	STPKEXIT * NAME : * FUNCTION: * INPUT : * OUTPUT : * OUTPUT : * CALLS : * ESCON	EGU PLA LDY RTS ESCON TURN (NONE 'CHAR NOTHI PICK.S EGU PHA LDY JSR STA AND	* YSAV1 SON 'ESCAPE' '=ORIGINAL C NG STORCHAR * PICK CHAR #\$E0	, STUFF ONTO SCREEN ; RESTORE AC ; RESTORE Y
CF4E: CF4E: CF4E: CF4I: CF4F: A1 CF52: CF52: CF53: 78 CF53: 78 CF58: 01 CF58: 01 CF58: 01 CF58: 01	168 169 170 171 172 173 174 175 174 175 176 177 178 177 181 181 183 184 185 186 187 188 189 190	STPKEXIT * NAME : * FUNCTION: * INPUT : * UDUTUT : * VOLATELE: * CALLS : * ESCON	EQU PLA LDY RTS ESCON TURN (NONE 'CHAR VCHAR PICK.S EQU PHA LDY JSR STA AND EOR JMP	* YSAV1 	, STUFF ONTO SCREEN , RESTORE AC , RESTORE Y
CF4E: CF4E: CF4E: CF4E: A 1 CF4F: A 1 CF52: CF52: CF52: CF53: R0 CF54: CF53: N N CF58: N N CF58: N N CF58: N N CF58: N N CF54: 48 CF54: CF54: CF 80 CF64: 42 C CF62: CF 67 CF65: CF CF65:	168 169 170 171 172 173 174 175 174 175 176 177 178 177 181 181 183 184 185 186 187 188 189 190	STPKEXIT * NAME : * FUNCTION: * INPUT : * UDLATELE: * CALLS : * ESCON	EQU PLA LDY RTS ESCON TURN (NONE 'CHAR VCHAR PICK.S EQU PHA LDY JSR STA AND EOR JMP	* YSAV1 	, STUFF ONTO SCREEN , RESTORE AC , RESTORE Y
CF4E: CF4E: CF44 CF4E: CF4 CF4F: A4 1F CF51: 60 CF52:	168 169 170 171 172 173 174 175 174 175 176 177 178 177 181 181 183 184 185 186 187 188 189 190	STPKEXIT * NAME : * FUNCTION: * INPUT : * UDLATELE: * CALLS : * ESCON	EQU PLA LDY RTS ESCON TURN (NONE 'CHAR VCHAR PICK.S EQU PHA LDY JSR STA AND EOR JMP	* YSAV1 	, STUFF ONTO SCREEN , RESTORE AC , RESTORE Y
CF4E: CF4E: CF4E: CF4E: A 1 CF4F: A 1 CF52: CF52: CF52: CF53: R0 CF54: CF53: N N CF58: N N CF58: N N CF58: N N CF58: N N CF54: 48 CF54: CF54: CF 80 CF64: 42 C CF62: CF 67 CF65: CF CF65:	1689 1689 1700 1711 1731 1731 174 1755 176 1777 178 182 1884 185 184 185 186 187 1882 189 190 191 192 1934 195 195	STPKEXIT * NAME : * FUNCTION: * INPUT : * OUTPUT : * CALLS : * ESCON * NAME : * FUNCTION: * FUNCTION: * INPUT : * OUTPUT :	EQU PLA LDY RTS ESCON TURN (NONE 'CHAR NOTHI PICK.S EQU PHA TYA PHA TYA PHA LDY JSR STA AND EOR JMP ESCOFF TURN ('CHAR	* YSAV1 DN 'ESCAPE' '=ORIGINAL C 40 STORCHAR * OURCH PICK * OURCH PICK * OURCH PICK SCRET - - - - - - - - - - - - -	, STUFF ONTO SCREEN , RESTORE AC , RESTORE Y
CF4E: CF4E: CF41 CF4E: CF CF52: CF52: CF52: CF53: CF55: CF53: NB CF54: AB CF54: AB CF54: CF55: CF55: CF55: CF65:	1688 1689 1700 1711 1722 1731 174 175 176 1775 176 1775 178 1775 1810 1821 1833 184 1853 186 187 1994 1995 1996	STPKEXIT * NAME : * FUNCTION: * UNPUT : * OUTPUT : * CALLS : * ESCON * NAME : * FUNCTION: * INPUT : * UNDTION: * UNDTION: * UNDTION: * UNDTIE: *	EGU PLA LDY RT5 ESCON TURN (NONE 'CHAR NOTHI PICK/S EGU PHA LDY PHA LDY PHA LDY PHA LDY CHAR STA AND ECR JMP ESCOFF TURN ('CHAR NOTHIN	* YSAV1 ON 'ESCAPE' '=ORIGINAL C 00 STORCHAR * OURCH PICK CHAR #SD #SD #SD FF 'ESCAPE' 	, STUFF ONTO SCREEN , RESTORE AC , RESTORE Y
CF4E: CF4E: CF4F: CF4E: A4 1F CF51: 60 CF52: C	168 2 169 170 171 172 173 174 175 176 177 177 177 177 177 177 177 177 177	STPKEXIT * NAME : * FUNCTION: * INPUT : * UDIPUT : * CALLS : * ESCON * NAME : * FUNCTION: * FUNCTION: * INPUT : * OUTPUT : * CALLS : CALLS : * CALLS : * * OUTPUT : * * CALLS : * * CALLS : * * * * * * * * * * * * * * * * * * *	EQU PLA LDY RTS ESCON TURN (NONE 'CHAR NOTHI PICK.S EQU PHA TYA PHA TYA PHA LDY JSR STA AND EOR JMP ESCOFF TURN ('CHAR	* YSAV1 ON 'ESCAPE' '=ORIGINAL C 00 STORCHAR * OURCH PICK CHAR #SD #SD #SD FF 'ESCAPE' 	, STUFF ONTO SCREEN , RESTORE AC , RESTORE Y
CF4E: CF4E: CF41 CF4E: CF CF52: CF52: CF52: CF53: CF55: CF53: NB CF54: AB CF54: AB CF54: CF55: CF55: CF55: CF65:	168 2 169 170 171 172 173 174 175 176 177 177 177 177 177 177 177 177 177	STPKEXIT * NAME : * FUNCTION : * UNUTUT : * OULATELE : * CALLS : * ESCON * NAME : * FUNCTION : * FUNCTION : * UNUTUT : * UNUTUT : * UNUTUT : * UNUTUT : * UNUTUT : * CALLS :	EGU PLA LDY RT5 ESCON TURN (NONE 'CHAR NOTHI PICK/S EGU PHA LDY PHA LDY PHA LDY PHA LDY CHAR STA AND ECR JMP ESCOFF TURN ('CHAR NOTHIN	* YSAV1 ON 'ESCAPE' '=ORIGINAL C 00 STORCHAR * OURCH PICK CHAR #SD #SD #SD FF 'ESCAPE' 	, STUFF ONTO SCREEN , RESTORE AC , RESTORE Y
CF4E: CF4E: CF41 CF4E: CF CF52: C	168 2 169 170 171 172 173 174 175 177 178 175 176 177 178 180 177 178 180 182 183 187 179 181 181 181 185 186 187 198 187 198 199 191 191 192 193 194 195 197 198 197 197 198 199 193 197 198 199 193 197 198 199 193 199 200 199 199 199 200 201 199 200 190 200 190 200 190 200 190 200 190 200 190 200 190 200 190 200 200 200 200 200 200 200 200 200 2	STPKEXIT * NAME : * FUNCTION: * NUTUT : * OUTPUT : * CALLS : * ESCON * * NAME : * FUNCTION: * NUTUT : * OUTPUT : * OUTPUT : * CALLS : * CALLS : *	EGU PLA LDY RTS ESCON TURN (NOTHEIN PICK.S EGU PICK.S EGU JMP EGU JSR STA AND JSR STA AND JMP ESCOFF CHAR NOTHIN STORC EGU	* YSAV1 ON 'ESCAPE' '=ORIGINAL C 00 STORCHAR * OURCH PICK CHAR #SD #SD #SD FF 'ESCAPE' 	, STUFF ONTO SCREEN , RESTORE AC , RESTORE Y
CF4E: CF4E: CF4E: A CF4E: A4 CF4F: A4 CF52: CF52: CF53: CF53: CF53: CF54: CF53: CF65: CF63: CF65: CF63: CF65: CF63: CF65: CF65:	168 1170 1172 1173 1170 1172 1173 1170 1172 1173 1174 1175 1176 1177 1178 1177 1178 1181 1184 1185 1184 1185 1184 1185 1189 1191 1193 1194 1195 1196 1197 1197	STPKEXIT * NAME : * FUNCTION: * NUTUT : * OUTPUT : * CALLS : * ESCON * * NAME : * FUNCTION: * NUTUT : * OUTPUT : * OUTPUT : * CALLS : * CALLS : *	EGU PLA RTS ESCON TURN (CHAR YCHAR PHA LDY PHA LDY PHA AND ESCOPF TURN (CHAR STA AND ESCOPF TURN (CHAR STA NOTHIN STAC EGU PHA	* YSAV1 ON 'ESCAPE' '=ORIGINAL C 00 STORCHAR * OURCH PICK CHAR #SD #SD #SD FF 'ESCAPE' 	, STUFF ONTO SCREEN ; RESTORE AC ; RESTORE Y
CF4E: CF4E: CF4E: A CF4E: A4 CF4F: A4 CF52: CF52: CF53: CF53: CF53: CF54: CF53: CF65: CF63: CF65: CF63: CF65: CF63: CF65: CF65:	1688 1697 1701 1712 1733 1722 1733 174 1753 174 175 176 1801 1811 1811 1811 1811 1835 1846 1855 1867 1921 1931 1941 1955 1977 1985 1972 1972 1972 1972 1972 1972 1975 1972 1975 1975 1975 1975 1975 1975 1975 1975	STPKEXIT * NAME : * FUNCTION: * INPUT : * OUTPUT : * CALLS : * ESCON * NAME : * FUNCTION: * FUNCTION: * OUTPUT : *	EQU PLA ESCON TURN (CHAR PHA NONE EQU PHA AND STA AND STA ESCOFF TURN (CCHAR NONE ESCOFF TURN (CCHAR NOTHI STORC) EQU PHA STORC)	* YSAV1 YSAV1 ON 'ESCAPE' ' ORIGINAL C ' ORIGINAL C ' OURCH PICK * OURCH * OURCH * OURCH SECRET ' ' OFF 'ESCAPE' ' AR * *	, STUFF ONTO SCREEN , RESTORE AC , RESTORE Y
CF4E: CF4E: CF4E: A CF4E: A4 CF52: CF52: CF52: CF53: CF53: R0 CF53: R0 CF54: R0 CF54: R0 CF64: CF65: CF65:	1688 1699 1700 1711 1721 1733 174 1753 174 1753 174 1753 174 1801 1855 1867 1881 1845 1855 1845 1855 1845 1902 1912 1912 1913 1942 1952 2011 2022 2032 2045 2055 2055 2055 2055 2055 2055 205	STPKEXIT * NAME : * FUNCTION: * INPUT : * OUTPUT : * CALLS : * ESCON * NAME : * FUNCTION: * FUNCTION: * FUNCTION: * OUTPUT : * OUTPUT : * OUTPUT : * CALLS : * ESCOFF	EQU PLA LDY RTS ESCON TURN (CHAR PHA NOTHI PICK 5 CHAR PHA AND JSR EQU JSR EQU JSR ECOR CHAR NOTHI STORC EQU ECOR CHAR STORC CHAR S	* YSAV1 '=GRIGINAL C ' OURCH Y OURCH PICK CHAR * SOO ESCRET	, STUFF ONTO SCREEN , RESTORE AC , RESTORE Y
CF4E: CF4E: CF4E: A CF4E: A4 CF4E: A4 CF52: CF52: CF53: CF54: CF53: CF67: CF63: CF65: CF63: CF65: CF63: CF65: CF63: CF65: CF65: CF65: CF63: CF65: CF65: CF65: CF63: CF65: CF65: CF65: CF65: CF65: CF65: CF65: CF65: CF65: CF65:	168 1170 1172 1173 1172 1173 1172 1173 1172 1173 1174 1175 1174 1175 1174 1175 1174 1175 1174 1175 1174 1175 1174 1175 1175	STPKEXIT * NAME : * FUNCTION: * INPUT : * UDLATELE: * CALLS : * ESCON * * NAME : * PUNCTION: * NUNCTION: * NUPUT : * OUTPUT : * CALLS : * * CALLS : * * * ESCOFF	EQU PLA LDY RTS ESCON TURN (CHAR PHA NOTHI PICK 5 CHAR PHA AND JSR EQU JSR EQU JSR ECOR CHAR NOTHI STORC EQU ECOR CHAR STORC CHAR S	* YSAV1 '=GRIGINAL C ' OURCH Y OURCH PICK CHAR * SOO ESCRET	, STUFF ONTO SCREEN , RESTORE AC , RESTORE Y
CF4E: CF4E: CF4E: A CF4E: A4 CF4F: A4 CF52: CF52: CF52: CF53: CF53: R0 CF53: R0 CF53: R0 CF64: R0 CF65: CF65: CF65: CF6	168 1170 1172 1173 1172 1173 1172 1173 1172 1173 1174 1175 1174 1175 1174 1175 1174 1175 1174 1175 1174 1175 1174 1175 1175	STPKEXIT A NAME : FUNCTION: FUNCTION: OUTPUT : OUTPUT : CALLS : FUNCTION: FUNCTION: FUNCTION: FUNCTION: CALLS : FUNCTION: CALLS : FUNCTION: FUNCT	EQU PLA LDY RTS ESCON TURN (CHAR PHA NOTHI PICK 5 CHAR PHA AND JSR EQU JSR EQU JSR ECOR CHAR NOTHI STORC EQU ECOR CHAR STORC CHAR S	* YSAV1 '=GRIGINAL C ' OURCH Y OURCH PICK CHAR * SOO ESCRET	, STUFF ONTO SCREEN , RESTORE AC , RESTORE Y
CF4E: CF4E: CF4E: A CF4F: A4 CF52: CF52: CF52: CF53: CF53: R0 CF53: R0 CF53: R0 CF54: R0 CF64: CF65: CF65: CF65: CF64: CF64: CF64: CF65: CF64: CF64: CF64: CF64: CF64: CF64: CF64:	168 169 169 169 169 169 169 169 169 169 169	STPKEXIT * NAME : * FUNCTION: * INPUT : * UDLATELE: * CALLS : * ESCON * * NAME : * PUNCTION: * NUNCTION: * NUPUT : * OUTPUT : * CALLS : * * CALLS : * * * ESCOFF	EQU PLA ESCON TURN (CHAR PHA NOTHI PICK : CHAR PHA NOTHI PICK : CHAR PHA AND STA CHAR PHA AND STA CHAR PHA STA CHAR PHA STA CHAR PHA STA CHAR PHA STA CHAR STA STA STA CHAR STA STA STA STA STA STA STA STA STA STA	* YSAV1 YSAV1 ON 'ESCAPE' ' ORIGINAL C ' ORIGINAL C ' OURCH PICK * OURCH * OURCH * OURCH SECRET ' ' OFF 'ESCAPE' ' AR * *	, STUFF ONTO SCREEN , RESTORE AC , RESTORE Y CURSOR CHAR CHAR SAVE AC ; AND Y ; GET CH ; GET OR IGINAL CHARACTER ; AND REMEMBER FOR ESCOFF ; AND REMEMBER FOR ESCOFF ; AND REMEMBER FOR ESCOFF ; AND REMEMBER FOR ESCOFF ; SAVE AC ; CURSOR ; AND Y ; GET CH ; GET
CF4E: CF4E: CF4E: A CF4E: A CF52: CF52: CF53: R0 CF53: CF53: CF53: NB CF54: 48 CF55: CF65: CF65: CF65: CF64: <t< td=""><td>1688 1699 1700 1711 1722 1731 1742 1754 1774 1775 1744 1777 1781 1800 1811 1811 1811 1811 1811</td><td>STPKEXIT * NAME : * FUNCTION: * INPUT : * UDLATELE: * CALLS : * ESCON * * NAME : * PUNCTION: * NUNCTION: * NUPUT : * OUTPUT : * CALLS : * * CALLS : * * * ESCOFF</td><td>EGU PLA RTS ESCON TURN (PCHAR PHA STA DAN EGU PHA STA STA AND ESCOFF TURN (CHAR NOTHI STORCC PHA STO STORCC PHA STO STORCC PHA STO STORCC PHA STO STORCC PHA STO STORCC PHA STO STORCC PHA STO STORCC PHA STO STORCC PHA STORCC PHA STORCC PHA STORCC PHA STORCC PHA STORCC PHA STORCC PHA STORCC PHA STORCC PHA STORCC PHA STORCC PHA STO STO STO STO STORCC PHA STO STO STO STO STO STO STO STO STO STO</td><td>* YSAV1 YSAV1 YSAV1 Y=ORIGINAL C G STORCHAR * OURCH PICK CHAR ESCRET F F CHAIGINAL C G URCH AR * OURCH CHAR * SEV</td><td>, STUFF ONTO SCREEN , RESTORE AC , RESTORE Y </td></t<>	1688 1699 1700 1711 1722 1731 1742 1754 1774 1775 1744 1777 1781 1800 1811 1811 1811 1811 1811	STPKEXIT * NAME : * FUNCTION: * INPUT : * UDLATELE: * CALLS : * ESCON * * NAME : * PUNCTION: * NUNCTION: * NUPUT : * OUTPUT : * CALLS : * * CALLS : * * * ESCOFF	EGU PLA RTS ESCON TURN (PCHAR PHA STA DAN EGU PHA STA STA AND ESCOFF TURN (CHAR NOTHI STORCC PHA STO STORCC PHA STO STORCC PHA STO STORCC PHA STO STORCC PHA STO STORCC PHA STO STORCC PHA STO STORCC PHA STO STORCC PHA STORCC PHA STORCC PHA STORCC PHA STORCC PHA STORCC PHA STORCC PHA STORCC PHA STORCC PHA STORCC PHA STORCC PHA STO STO STO STO STORCC PHA STO STO STO STO STO STO STO STO STO STO	* YSAV1 YSAV1 YSAV1 Y=ORIGINAL C G STORCHAR * OURCH PICK CHAR ESCRET F F CHAIGINAL C G URCH AR * OURCH CHAR * SEV	, STUFF ONTO SCREEN , RESTORE AC , RESTORE Y
CF4E: CF4E: CF4E: A CF4F: A4 CF52: CF52: CF52: CF53: CF53: R0 CF53: R0 CF53: R0 CF54: R0 CF64: CF65: CF65: CF65: CF64: CF64: CF64: CF65: CF64: CF64: CF64: CF64: CF64: CF64: CF64:	168 169 169 169 169 169 169 169 169 169 169	STPKEXIT * NAME : * FUNCTION: * INPUT : * UDLATELE: * CALLS : * ESCON * * NAME : * PUNCTION: * NUNCTION: * NUPUT : * OUTPUT : * CALLS : * * CALLS : * * * ESCOFF	EQU PLA ESCON TURN (CHAR PHA NOTHI PICK : CHAR PHA NOTHI PICK : CHAR PHA AND STA CHAR PHA AND STA CHAR PHA STA CHAR PHA STA CHAR PHA STA CHAR PHA STA CHAR STA STA STA CHAR STA STA STA STA STA STA STA STA STA STA	* YSAV1 YSAV1 YSAV1 Y=ORIGINAL C G STORCHAR * OURCH PICK CHAR ESCRET F F CHAIGINAL C G URCH AR * OURCH CHAR * SEV	, STUFF ONTO SCREEN , RESTORE AC , RESTORE Y CURSOR CHAR CHAR SAVE AC ; AND Y ; GET CH ; GET OR IGINAL CHARACTER ; AND REMEMBER FOR ESCOFF ; AND REMEMBER FOR ESCOFF ; AND REMEMBER FOR ESCOFF ; AND REMEMBER FOR ESCOFF ; SAVE AC ; CURSOR ; AND FUT IN INVERSE 11 ; SAVE AC ; AND Y ; GET CH ; MAR ; MA

47

CF77: 60	213	RTS		
CF78: CF78:	214	COPYR	CM	nin ang man ang mga mga mga man ang man ang
CF78:	216 * FUNCTION:	COPY	F8 ROM TO LO	CARD
CF78:	217 * INPUT :	NOTHI	NG	
CF78: CF78:	218 * VOLATILE: 219 * CALLS :		NO	
CF78:	220	NUTHI	~~~~~~	
CF78:	221 *			
CF78: CF78	222 COPYROM	EQU	*	
CF78: 48 CF79: 08	223	PHA PHP		SAVE AC
CF7A: 78	225	SEI		IENSURE IRG INHIBITED
CF7B:	226 *			
CF7B: AD 11 CO CF7E: 48	227 228	LDA PHA	RDLCBNK2	GET BANK2
CF7F: 46	229 *	P.HM		
CF7F: AE 12 CO	230	LDX	RDLCRAM	AND RAM FLAGS
CF92: AD 81 CO	231	LDA	\$C081	SET READ-ROM
CF85: AD 81 CO CF88:	232 233 *	LDA	\$C081	WRITE-RAM MODE
CF88: A0 00	234	LDY	#0	
CF8A: A9 F8	235	LDA	#\$F8	
CF9C: 85 37	236	STA	CSWH	/USE HOOK FOR MOVE /PRESERVE LO BYTE
CF8E: A5 36 CF90: 48	237 238	PHA	CSWL	PRESERVE LU BYTE
CF91: A9 00	239	LDA	#0	
CF93:85 36	240	STA	CSWL	
CF95: CF95	241 COPYROM2 242	EGU	* (CSWL.), Y	COPY ONLY PATCHED PAGES
CF95: B1 36 CF97: 91 36	243	STA	(CSWL),Y	HOVE THE RON
CF99: C8	244	INY		
CF9A: DO F9 CF95	245	BNE	COPYROM2	
CF9C: E6 37 CF9E: D0 F5 CF95	246 247	INC	CSWH COPYROM2	
CFA0:	248 *	DIAC	COLLEGE	
CFA0: 68	249	PLA		RESTORE THE
CFA1:85 36	250 251	STA	CSWL # <cnoo< td=""><td>; HOOK</td></cnoo<>	; HOOK
CFA3: A9 C3 CFA5: 85 37	252	STA	CSWH	
CFA7:	253 *	0.111	U GHI I	
CFA7: 68	254	PLA		WHICH LC BANK?
CFAB: 10 OF CFB9	255	BPL	LCB1	;=>BANK1 ;RAM OR ROM READ?
CFAA: BA CFAB: 10 06 CFB3	257	BPL	LCB2ROM	i=>ROM
CFAD: AD BO CO	258	LDA	\$C080	BANK2, RAM
CFBO: 4C C5 CF	259	JMP	COPYRET	
CFB3: AD 81 CO CFB6: 4C C5 CF	260 LCB2ROM 261	LDA	\$COB1 COPYRET	; BANK2, ROM
CFB9: 8A	262 LCB1	TXA	COR THE T	RAM DR ROM READ?
CFBA: 10 06 CFC2	263	BPL	LCBIROM	; =>ROM
CFBC: AD 88 CO CFBF: 4C C5 CF	264 265	LDA	\$C088	; BANK1, RAM
CFC2: AD 89 CO	266 LCB1ROM	LDA	COPYRET \$CO89	BANK 1, ROM
CFC5:	267 *	CON	+0007	/ Deliver / Nort
CFC5: CFC5	268 COPYRET	EQU	*	
CFC5: 28	269 270	PLP		RESTORE IRQ STATE NOW
CFC6: 68 CFC7: 60	271	RTS		, AND AC
CFCB:	272			
CFCB:	273 * NAME : 274 * FUNCTION:	PSETU	P	
CFCB: CFCB:	274 * FUNCTION: 275 * INPUT	NONE	ZP FOR PAS	JAL
CFCB:	276 * OUTPUT :	NONE		
CFCB:	277 * VOLATILE:			
CFCB: CFCB-	278 * CALLS : 279	NOTHI	NG	
CFC8:	280 *			
CFC8: CFC8	281 PSETUP	EQU	*	
CFCB: AD FB 04	282	LDA	MODE	; TRANSPARENT MODE?
CFCB: 29 01 CFCD: DO 03 CFD2	283 284	AND BNE	#M. TRANS PSETUP2	;=>YES, TRUST WINDOW
CFCF: 20 78 CD	285	JSR	FULLBO	SET FULL BOCOL WINDOW
CFD2:	286 *			
CFD2: CFD2	287 PSETUP2 288	EQU	* #255	
CFD2: A9 FF CFD4: 85 32	288	LDA STA	#255 INVFLG	ASSUME NORMAL MODE
CFD6:	290 *	018	11471 68	ABOULT HONTHE HODE
CFD6: AD FB 04	291	LDA	MODE	
CFD9: 29 04	292	AND	#M. VMODE	
CFDB: FO 02 CFDF CFDD: 46 32	293 294	BEQ	PSETUPRET INVFLQ)=>IT'S NORMAL ;MAKE IT INVERSE
CFDF:	295 *	_		
CFDF: CFDF	296 PSETUPRET	EQU	*	
CFDF: AD 7B 07	297	LDA	OLDBASL BASL	; SET UP BASE ADDRESS
CFE2:85 28 CFE4:AD F8 07	298 299	STA	BASL OLDBASH	
CFE7:85 29	300	STA	BASH	
CFE9: 60	301	RTS		
CFEA: CFEA:	302	RIFS 4	-7 OF THEEF	TABLES
or ER.	SUG A ROLE, ENI		. of THEOE	

	12 844 1001	and the second second	2714 P. 2010 T. 2010	with the second s
CFEA:				THUS THERE ARE
CFEA:	305	* SOME OTH	HER VAL	UES STUFFED IN.
CFEA:	306	*		
CFEA: 28	307	F. TABLE	DFB	>F. CLREOP-1
CFEB: 42	306	3	DFB	>F. HOME-1
CFEC: 4C	309	7	DFB	>F. SCROLL-1
CFED: 7C	310	>	DFB	>F. CLREOL-1
CFEE: 9B	311		DFB	>F. CLEOLZ-1
CFEF: E9	312	2	DFB	>B. RESET-1 ; USE SAME RESET
CFFO: FF 01	313	PLUSMINUS1	DFB	-1,1 SCROLL USES THIS
CFF2: 89	314	ł	DFB	>F. SETWND-1
CFF3:	315	5 *		
CFF3: EO	314	B. TABLE	DFB	>B. CLREOP-1
CFF4: EC	317	7	DFB	>8. HOME-1
CFF5: CC	318	3	DFB	>B. SCROLL-1
CFF6: D2	315	7	DFB	>B. CLREOL-1
CFF7: D8	320)	DFB	>8. CLEOLZ-1
CFF8: E9	321		DFB	>B. RESET-1 / USE SAME RESET
CFF9: 23 22	322	WNDTAB	DFB	WNDETM, WNDTOP ; SCROLL USES THIS
CFFB: E6	323	3	DFB	>B. SETWND-1
CFFC: 00	324	1	DFB	0 AVOID CFFF PIPELINING
CFFD:	CFFD 15	ZZEND	EQU	*

80-Column Symbol Table, Sorted by Symbol

	127723						
3D	A1H	30	AIL	ЗF	A2H	ЗE	A2L
43	A4H	42	A4L	CE13	ATEFOR1	CEOA	ATEFOR
CA02	B. CANLIT	C9DF	B. CHKCAN	C1D9	B. CLEOLZ	C1D3	B. CLREOL
C1E1	B. CLREOP	C26E	B. ESCFIX	C272	8. ESCETX2	C27A	B ESCETX3
CAOA	8 FIXCHR	CRET	BELTP	C144	B FUNCO	CIOF	B FUNCHE
20100	B FLINC	0211	P FUNC1	C107	D FUNCMU	CODE	B OFTON
CIED	P LIDME	CROE	B. INDUT	0107	B. FUNCINA	0202	D. WEIGH
CIED	B. NUME	6905	B. INPUT	CA24	B. INRE!	0588	B. KEYIN
C29C	B. KEYIN2	C9C6	B. NOPICK	CIIF	B. OLDFUNC	C1EA	B. RESET
C234	B. RESETX	CICD	B. SCROLL	C221	B. SETWND2	C219	B. SETWNDX
C1E7	B. SETWND	CFF3	8. TABLE	C1FF	B. VECTOR	28	BAS2H
24	BAS2L	CB54	BASCALCZ	CB51	BASCALC	CB7E	BASCLC3
CB97	BASCLCX	29	BASH	C317	BASICENT	0336	BASICENT2
6803	RASICINIT	C205	BASICIN	20200	BACTCINT	6307	PARICOUT
2000	DACI	0000	DELLO	1000	DELINICIAL	0307	BASICOUT
2011	BHOL	CBC3	BELLe	0100	BFONCPG	6831	BINITIA
0816	BINITI	C850	BINIT2	C8F6	BINPUT	CBE2	BIORET
C252	BLAST	CB96	BOUT	0300	BPNCTL	C8A1	BPRINT
?CBE2	BS40	CB5B	BSCLC1A	CB55	BSCLC1	CB6D	BSCLC2
CBEB	BSDONE	C398	CO1	C3A3	C03	CB74	C682
C87E	CBB3	C890	CBB4	0866	CBBASIC	07FB	CASLOT
24	СН	0678	CHAR	6850	CLEARIT	C181	CLEDI 2
C170	CLEORI	0075	CI BROCOL	0000	CLEMAIN	0101	CLEUL2
CIED	CLEUFI	0000	CERBOCOL	COUC	CLRBUVID	CODE	CERAL TCHAR
CEAR	CLRHALF2	CE91	CLRHALF	C300	CNOO	CFC5	COPYRET
CF95	COPYROM2	CF78	COPYROM	37	CSWH	36	CSWL
CC78	CTLADH	CC5F	CTLADL	CB99	CTLCHAR	CBAB	CTLCHARX
CBAE	CTLOD	CBB2	CTLRET	CBB6	CTLXFER	25	CV
C261	DIAGS	CEAR	D048	6929	ESC1	C928	ESC2
0735	ESC3	C918	ESCAPING	0983	FSCCHAR	0280	ESCIN
0700	ECCHIONE	0011	COUNT LINE	0703	COOPER	0260	EBCIN
6780	ESCHURE.	0011	ESCNOR	CF85	ESCUPP	CF 52	ESCON
C284	ESCOUT	CF6E	ESCRET	C945	ESCSPEC	C954	ESCSPEC2
C963	ESCSPEC3	C972	ESCTAB	?FBC1	F. BASCALC	C19C	F. CLEDLZ
C17D	F. CLREOL	C129	F. CLREOP	C1A1	F. GORET	C143	F. HOME
C2F1	F. RET1	C2EB	F. RETURN	C14D	F. SCROLL	C1BA	F. SETWND
CFEA	F. TABLE	FC22	F VTAR	EC24	E VTAR7	FRRG	FRVERSION
CEAE	FORATE1	CE42	FORATE	CDOP	FULLEO	Enge	FUNCEVIT
CEDO	OFTOA	0203	OFTIC	0078	PULLOU	F D & 7	ACTORITOR
GEEE	GE 104	CBIB	GEINZ	CBID	GEINEY	CAR/	GETERIOR
CAAF	GETY	CSSE	GOBACK	06	QOODF8	C27D	GORETN
C2B5	GOTKEY	CA49	GPX	C813	HANG	CSCC	IK1
C2D5	IK2A	CSCE	IK2	C2DB	IK3	CEDD	INVERT
32	INVFLG	FF5B	IORTS	C348	JBASINIT	C34B	JPINIT
C351	JPREAD	C350	JPSTAT	C357	JPWRITE	C010	KBDSTRR
0000	KBD	CRBA	KROWATT	COEA	KDRET	20254	KORETN
COER	KORETY	0004	KEVNIV	20	KELL	I VELD	Kell
CZE7	LORA	Care	KETULT	37	NEWH	38	NOWL
CF89	LUBI	CFC2	LCBIROM	CFB3	LC85KOW	40	M. BINPUT
80	M. ESCR	08	M. GOXY	01	M. IRG	10	M. LIT
02	M. PAS1. 0	20	M. PASCAL	01	M. TRANS	04	M. VMODE
04FB	MODE	C378	MOVEC 2M	C380	MOVELOOP	CBAC	MOVERET
C37E	MOVESTRT	C363	MOVE	CCDD	MSCRI O	CCE1	MSCRL 1
CCE9	MSCRI 2	cnog	MECHIPET	CDIO	MRCOL DTC	0987	NOFEC
C105	NOT	0000	MOUATT	0000	NYTAI	0759	OL DRACH
0778	OL DBACL	COCO	OLDOWAL I	CJOA	NA LAL	0/68	ULUBRON
0778	DEDBAGE	04/8	ULDCA	0009	UNEMURE	0578	UURCH
0518	OURCV	CF01	PICK	CA62	PIGOOD	CA4A	PINIT1.0
CA51	PINIT2	CA4F	PINIT	CFFO	PLUSMINUS1	CA74	PREAD
CABA	PREADRET2	CFD2	PSETUP2	CFC8	PSETUP	CFDF	PSETUPRET
C99E	PSTATUS2	C994	PSTATUS	C780	PSTATUS3	C784	PSTATUS4
2CA9E	PWRITE2	CACE	PWRITES	CABE	PWRITE	CAFR	PWRITE4
CBOF	PWRITERET	CROP	PUURAP	CDCO	OUTTO	CDAA	GUTT
CON	PDBOCOL	COLE	PROUTD	0000	BDCABDBAM	DEDOC	DDVEV
COIB		COIF	RDOUVID	6003		AP DOC	RUNET
C011	RULCBNKZ	C012	RULCRAM	C005	RUMAINRAM	COIC	RDPAGE2
C013	RDRAMRD	C014	RDRAMWRT	C01A	RDTEXT	2019	RDVBLBAR
C264	RESETRET	4F	RNDH	4E	RNDL	CE01	SCR40RET
CDEA	SCR40	CEGE	SCRBO	CE55	SCRBORET	CF06	SCREENIT
C153	SCRL1	C169	SCRL2	C172	SCRL3	CCD1	SCRLSUB
CF1E	SCRN2	CE37	SCRN3	CE40	SCRNAO	CE32	SCRN48
CDDB	SCRNBA	CESP	SCRNRET	CCAE	SCROLL1	CCPO	SCROLL 2
CCAA	SCOOL DN	CCAA	BODDI LUD	CORE	SETROCOL	0000	CETOOUTD
COAT	OCTAL TOULO	CCA4	SCRULLUP	0001	SEIBUCUL	COOD	SEIBOVID
COOF	DE IALICHAR	C009	BETALTZP	COED	SEICB	CEAF	SEICH
CED9	SETCHRTS	?0007	SETINTCXROM	FEB9	SETKBD	COOB	SETSLOTCOROM
C008	SETSTDZP	FE93	SETVID	CFOO	SEV	FC75	SNIFFIRG
C030	SPKR	CADC	STARTXY	CB48	STAY2	CB4D	STAY80
CEF9	STOR2	CF44	STOR40	CF24	STORBO	CEFP	STORCHAR
CEAF	STPRETT	CCSP	BTUFFINV	0479	TEMP 1	00	TEST
CROA	TERTCARD	CRAF	TERTEAT	0476	TYTRACEL	000	TYTRADED
0.024	LATT	CD+E	LEGITMIL	0034	INTERNEL	0035	INICHWER .
CBCF	WAIT	CBDO	WAITZ	CBD1	ETIAW	23	WINDETM
20	WNDLFT	CFF9	WNDTAB	22	WNDTOP	21	WNDWDTH
C003	WRCARDRAM	C004	WRMAINRAM	CBBC	X. BELL	CBDB	X. BS
C2F6	X. CLEOL2	C2F4	X. CLEOLZ	CCOC	X. CRRET	CBEC	X. CR
			AIL A4L A4L B. CHKCAN B. SCRDIX B. FLIP B. FUNC1 B. FLIP B. FUNC1 B. INPUT B. NDPICK B. SCRDLL B. SCRDLL B. SCRDLL2 BASCALC2 BASCALC2 BASCALC2 BASCALC2 BASCALC2 COMPACT BOU				

80-Column Symbol Table, Sorted by Symbol

	X. CRPAS	CD64	X. DC1B	CD76	X. DC1RTS	CD59	X. DC1
CD88	X. DC2B	CD7A	X. DC2RET	CD77	X. DC2	CCOD	X. EM
CD42	X. FF	0033	X. FSRET	CC26	X.FS	CD48	X. GS
CD54	X. GS2	CD4E	X. GSEOLZ	CC91	X. LF	CC9E	X. LF2
CD20	X. LERET	CD90	X. NAK	CD9A	X. NAKRET	CD11	X. SCRLRET
CD17	X. SCRLRET2	CC52	X.SI	CC49	X. 50	2001D	X. SUBBO
CC1F	X. SUELP	CC1A	X. SUB	CC34	X. US	CC40	X. US1
CC45	X. US2	CC48	X. USRET	CD23	X. VT	CD2C	X. VTLOOP
CD32	X. VINEXT	06FB	XCOORD	C3BO	XFER		XFERAZP
C3C5	XFERC2M	CODC	XFERSZP	1F	YSAV1		ZSPAREC2
?CFFD	ZZEND						
** SUC	CESSFUL ASSEMBL	Y :=	NO ERRORS				
** ASS	EMBLER CREATED	ON OS	-JAN-82 000004				
** TOT	AL LINES ASSEMI	LED	2419				
## FRE	E SPACE PAGE CO	UNT	49				
2	EQUATES						
з	BFUNC						
4	COSPACE						
5	CBSPACE						
6	BPRINT						
7	BINPUT						
8	PINIT						
9	PREAD						
10	PWRITE						
11	SUBS1						
12	SUBS2						
13	SUBS3						
10							

80-Column Symbol Table, Sorted by Address

~~		01	FI. 18
02	ZSPAREC2	04	M. VN
10		0011	ESCH
20		21	WND
24	CH	25	CV
24		28	BASa
	COUL		
37	CSWH	38	
ЭD	A1H	3E	A2L
42	A4L	43	
80		0478	TEMP
057B	OURCH	05FB	OURC
0770	OLDBASL	07F8	COPI
			COOL
C000		C001	SETE
C004	WRMAINRAM	C005	WRCA
0009	SETALTZP CLRALTCHAR	COOB	CETC
0007	OCT PICT AT	0000	aLIC
COOF	CLRALTCHAR	COOF	SETA
C012	RDLCRAM	C013	RDRA
010	RDVBLBAR	COIA	
		COIR	NUTE
CO30	SPKR	C054 C107 C12D	TXTE
C100	BFUNCPG	C107	B. FL
C170	F. CLREOP	C100	
0127	F. GLREOF	CIED	ULEL
	SCRL1	C169	SCRL
C181	CLEOL2	C18A	E SE
0.1 mm	B. FUNCO	C1C5 C1E1 C1FF C221 C261 C27A C288	NUT
C1D7	B. CLEOLZ	C1E1	B. CL
CIED	B. HOME	C1EE	B VE
0010	D. OCTUMBY	0001	
C514	B. SETWINDA	6221	B. 55
C252	BLAST	C261	DIAG
C272	B. ESCEIX2	C27A	B ES
0004	FREQUE	0000	
6204	B. FUNCO B. CLEOLZ B. HOME B. SETWNDX BLAST B. ESCFIX2 ESCOUT KEYDLY IK3 F. RETURN	C268	D. NE
C2C6	KEYDLY	C2CC	IK1
C2DB	IK3 F. RETURN	2C2E6	KDRE
COED	E DETURN	2226 2271 20300 2336 2357 2375	F
		CZFI	F. RE
C300	CNOO BASICENT	?0300	BASI
C317	BASICENT	6336	BAST
COLL	IDDEAD	0057	
166.0	JPREAD MOVEC2M	C357 C37E	JPWR
C378	MOVEC2M	C37E	MOVE
C398	CO1	C343	003
C3C5		C3CD	
6363	AFERGen	6360	AFER
CB03	BASICINIT	C813	HANG
C850	BINIT2 C8B3	C85D	CLEA
0075	CORO		000.4
COVE	0000	C870	C884
C884	KBDWAIT	CBCO	NOWA
CBEA	BINPUT	C905	R TN
0000	ESC2	0000	
0720	COUR	C935	ESCJ
C960	ESCNONE	C963	ESCS
0994	PSTATUS NOESC	C99E	PSTA
007	NOFOC	0776	
C481	NUESC	C7C6	B. NU
CA02	B. CANLIT	CAOA	B.FI
CA49	CPY	CA4A	
		01111	-
CHOZ	PIGOOD	CA74 CAAF	PREA
CA9E	PWRITE2	CAAF	GETY
CAFR	PWRITE4	CBO9	PLUD
CDID	GETK2	0007	
CB1B CB4E	VEINE	CB24	
CB4E	TESTFAIL	CB51	BASC
CRSB	BSCLC1A CTLCHAR	CB6D	RSCI
	OTL OULO	0000	
0844	CILCHAR	CBAB	CILC
CBB6	CTLXFER	CBBC	X. BE
RNO	WAIT2		
	DODOUT	0001	
CBEB	BSDONE	CBEC	X. CR
CCOD	X, EM	CC1A	X. SU
0522	YES	6622	VEC
		0000	2.10
CC 45	X. EM X. FS X. US2	CC48	X. US
CC 59	STUFFINV	CBD1 CBEC CC1A CC33 CC48 CC5F	CTLA
CC7E	X. LF2	CCA4	SCRO
CDE	0000110		JUNU
-088	SCRULL2	CCD1	SCRL
CCF9	MSCRL2	CD02	MSCR
DII	Y SCRI RET	CD17	Y SC
	A. BUREREI	CD1/	A. 30
:020	X. VTLOOP	CD32	X. VT
D4E	X. GSEDLZ	CD54	X. G5
774	X. SCRLRET X. VTLOOP X. GSEDLZ X. DC1RTS X. NAKRET	CD77	X. SC X. VT X. GS X. DC X. DC
	A. DETKIS	CD//	A. DC
:D7A	X. NAKRET	CD9A	X. DC:
ODCO	GUIT2	CDDB	SCRN
FOA	ATEEOD	CELE	ATE
LUA	MIEFUN	CE13	ATEF
CE3E	SCRBO	CE55	SCR8
E6F	X.LF2 SCROLL2 MSCRL2 X.SCRLRET X.VTLOOP X.QSEOLZ X.DCIRTS X.NAKRET GUIT2 ATEFOR SCR80 FORATE1		CLRH
		5271	where the

00 TEST

2

?

?(

20

6

C

¢

¢

c

01 M TRANS OI M TRO MODE NUM WDTH 28 BASL 2н 39 KSWH 3F A2H 4E RNDL P1 047B OLDCH cv 0678 CHAR OT 07FB OLDBASH BOCOL RDRAM BLOTCOROM COLO KBDSTRB LTCHAR AMRD EXT AGE 1 UNCNK DP 1 C172 SCRL3 2 ETWND CICD B. SCROLL REOP C20E B. GETCH CTOR ETWND2 C22E GOBACK 35 C264 RESETRET C27D GORETN SCFIX3 C29C B. KEYIN2 C2CE IK2 EYIN TN Τ1 CINT C305 BASICIN C348 JBASINIT CENT2 ITE C35D C380 MOVELOOP STRT COAC MOVERET RAZP C816 BINIT1 C866 C88A5IC -ARIT C896 BOUT CBCC BPNCTL AIT PUT C918 ESCAPING C945 ESCSPEC C972 ESCTAB C980 PSTATUS3 PEC3 TUS2 PICK C9DF B. CHKCAN CA24 B. INRET XCHR T1. 0 AD AP CARD CB48 STAY2 CB54 BASCALCZ ALC C2 CB7E BASCLC3 HARX CBAE CTLGO LL CBC3 BELL2 3 CBDB X.85 CBFD X. CRPAS ?CC1D X. SUBBO JΒ CC34 X.US RET RET CC78 CTLADH CCAA SCROLLON SUB CCDD MSCRLO CD09 DNEMORE LRET RLRET2 CD20 X. LFRET CD42 X. FF NEXT CD59 X. DC1 CD88 X. DC28 :2 2 2RET CD9B FULL80 CDEA SCR40 84 mp 1 CE22 GET84 ORET CE58 SCRNRET ALF CE9B CLRHALF2

02 M. PAS1. 0 02 M. PASI. 0 08 M. GOXY 20 M. PASCAL 23 WNDBTM 04 GOODFS 1F YSAV1 22 WNDTOP 29 BASH 32 INVELO 36 CSWL ЭC 40 M. BINPUT 4F RNDH 04FB MODE 06FB XCOORD COOD CLRBOCOL COO3 RDCARDRAM COO2 RDMAINRAM 20007 SETINTCIROM COOB SETSTDZP COOC CLRBOVID COOD SETROVID CO11 CO14 RDRAMWRT CO18 RDBOCOL COIC RDPAGE2 COSS TXTPAGE2 COIF 20100 B. FUNC CIOE B. FUNCNE C11F C143 F. HOME C14D F. SCROLL C17D F. CLREOL C19C F. CLEDLZ CIAL F. GORET CIDS B. CLREOL CIEZ B. SETWND CIEA B. RESET C211 B. FUNC1 C234 B. RESETX C26E B. ESCFIX C285 GOTKEY C2D5 1K2A C2E9 KDRETY C2F4 X. CLEDLZ CZEA KORET C2F6 X. CLEOL2 C307 BASICOUT C363 MOVE C38A NXTA1 JPSTAT C3BO XFER C3EB SETCO C831 BINITIA C874 C882 CBA1 BPRINT CBE2 BIDRET C929 ESC1 C954 ESCSPEC2 C783 ESCCHAR C984 PSTATUS4 C9F7 B.FLIP CA27 GETPRIOR CA4F PINIT CABA PREADRET2 CA51 PINIT2 CA8E PWRITE CACE PWRITE3 CBOF PWRITERET CADC STARTXY **CB15 GETKEY** CB4D STAY80 CB55 BSCLC1 CB97 BASCLCX CBB2 CTLRET CBCF WAIT ?CBE2 BS40 CCOC X. CRRET CC40 X. US1 CC52 X. SI CC91 X. LF CCAE SCROLL1 CCE1 MSCRL1 CD10 MSCRLRTS CD23 X.VT CD48 X.GS CD64 X. DC18 CD90 X. NAK CDAA QUIT CEO1 SCR4ORET CE32 SCRN48 CE63 FORATE CEA3 DD48

AIL

RDLCBNK2

RDBOVID

B. DLDFUNC

80-Column Symbol Table, Sorted by Address

	SETCH STOR2		SETCHRTS		INVERT		STORCHAR
		CFOO			PICK		SCREENIT
	SCRNZ		STOR80		SCRN3	CF40	SCRN40
	STOR40		STPKEXIT	CF52	ESCON	CF65	ESCOFF
	ESCRET	CF78	COPYROM	CF95	COPYROM2	CFB3	LC82ROM
CFB9	LCB1	CFC2	LCBIROM	CFC5	COPYRET	CFC8	PSETUP
CFD2	PSETUP2	CFDF	PSETUPRET	CFEA	F. TABLE	CFFO	PLUSMINUS1
CFF3	B. TABLE	CFF9	WNDTAB	2CFFD	ZZEND		FOVERSION
?FBC1	F. BASCALC	FC22			F. VTABZ		SNIFFIRG
?FDOC		FD29			SETKED		SETVID
FF58	IORTS						
	CESSFUL ASSEMBL	V ·=	NO FRRORS				
	EMBLER CREATED						
	AL LINES ASSEME						
	E SPACE PAGE CO						
2	EQUATES	10111	47				
	BFUNC						
4	COSPACE						
5	CBSPACE						
6	BPRINT						
7	BINPUT						
8	PINIT						
9	PREAD						
10	PWRITE						
11	SUBS1						
12	SUBS2						
13	SUB53						

