

```

F808:90 02 F80C 146 BCC RTMASK
F80A:69 0E 147 ADC #00 ;MASK $F0 IF ODD
F80C:85 2E 148 RTMASK STA MASK
F80E:B1 26 149 PLOT1 LDA (GBASL),Y ;DATA
F810:45 30 150 EOR COLOR ; XOR COLOR
F812:25 2E 151 AND MASK ; AND MASK
F814:51 26 152 EOR (GBASL),Y ; XOR DATA
F816:91 26 153 STA (GBASL),Y ; TO DATA
F818:60 154 RTS
F819: 155 *
F819:20 00 F8 156 HLINE JSR PLOT ;PLOT SQUARE
F81C:C4 2C 157 HLINE1 CPY H2 ;DONE?
F81E:B0 11 F831 158 BCS RTS1 ; YES, RETURN
F820:C8 159 INY ; NO, INCR INDEX (X-COORD)
F821:20 0E F8 160 JSR PLOT1 ;PLOT NEXT SQUARE
F824:90 F6 F81C 161 BCC HLINE1 ;ALWAYS TAKEN
F826:69 01 162 VLINEZ ADC #01 ;NEXT Y-COORD
F828:48 163 VLINE PHA ; SAVE ON STACK
F829:20 00 F8 164 JSR PLOT ; PLOT SQUARE
F82C:68 165 PLA
F82D:C5 2D 166 CMP V2 ;DONE?
F82F:90 F5 F826 167 BCC VLINEZ ; NO, LOOP.
F831:60 168 RTS1 RTS
F832: 169 *
F832:A0 2F 170 CLRSCR LDY #02F ;MAX Y, FULL SCR N CLR
F834:D0 02 F838 171 BNE CLRSC2 ;ALWAYS TAKEN
F836:A0 27 172 CLRTOP LDY #027 ;MAX Y, TOP SCR N CLR
F838:84 2D 173 CLRSC2 STY V2 ;STORE AS BOTTOM COORD
F83A: 174 ; FOR VLINE CALLS
F83A:A0 27 175 LDY #027 ;RIGHTMOST X-COORD (COLUMN)
F83C:A9 00 176 CLRSC3 LDA #000 ;TOP COORD FOR VLINE CALLS
F83E:85 30 177 STA COLOR ;CLEAR COLOR (BLACK)
F840:20 28 F8 178 JSR VLINE ;DRAW VLINE
F843:88 179 DEY ;NEXT LEFTMOST X-COORD
F844:10 F6 F83C 180 BPL CLRSC3 ;LOOP UNTIL DONE.
F846:60 181 RTS
F847: 182 *
F847:48 183 GBASCALC PHA ;FOR INPUT OODEFGH
F848:4A 184 LSR A
F849:29 03 185 AND #003
F84B:09 04 186 ORA #004 ;GENERATE GBASH=000001FG
F84D:85 27 187 STA GBASH
F84F:68 188 PLA ;AND GBASL=HDEDE000
F850:29 18 189 AND #018
F852:90 02 F856 190 BCC GBSCALC
F854:69 7F 191 ADC #07F
F856:85 26 192 GBSCALC STA GBASL
F858:0A 193 ASL A
F859:0A 194 ASL A
F85A:05 26 195 ORA GBASL
F85C:85 26 196 STA GBASL
F85E:60 197 RTS
F85F: 198 *
F85F:A5 30 199 NXTCOL LDA COLOR ;INCREMENT COLOR BY 3

```

```

F861:18 200 CLC
F862:69 03 201 ADC #003
F864:29 0F 202 SETCOL AND #00F ;SETS COLOR=17*A MOD 16
F866:85 30 203 STA COLOR
F868:0A 204 ASL A ;BOTH HALF BYTES OF COLOR EQUAL
F869:0A 205 ASL A
F86A:0A 206 ASL A
F86B:0A 207 ASL A
F86C:05 30 208 ORA COLOR
F86E:85 30 209 STA COLOR
F870:60 210 RTS
F871: 211 *
F871:4A 212 SCR N LSR A ;READ SCREEN Y-COORD/2
F872:08 213 PHP ;SAVE LSB (CARRY)
F873:20 47 F8 214 JSR GBASCALC ;CALC BASE ADDRESS
F876:B1 26 215 LDA (GBASL),Y ;GET BYTE
F878:28 216 PLP ;RESTORE LSB FROM CARRY
F879:90 04 F87F 217 SCR N2 BCC RTMSKZ ;IF EVEN, USE LO H
F87B:4A 218 LSR A
F87C:4A 219 LSR A
F87D:4A 220 LSR A ;SHIFT HIGH HALF BYTE DOWN
F87E:4A 221 LSR A
F87F:29 0F 222 RTMSKZ AND #00F ;MASK 4-BITS
F881:60 223 RTS
F882: 224 *
F882:A6 3A 225 INSDS1 LDX PCL ;PRINT PCL,H
F884:A4 3B 226 LDY PCH
F886:20 96 FD 227 JSR PRYX2
F889:20 48 F9 228 JSR PRBLNK ;FOLLOWED BY A BLANK
F88C:A1 3A 229 LDA (PCL,X) ;GET OPCODE
F88E:A8 230 INSDS2 TAY
F88F:4A 231 LSR A ;EVEN/ODD TEST
F890:90 09 F89B 232 BCC IEVEN
F892:6A 233 ROR A ;BIT 1 TEST
F893:80 10 F8A5 234 BCS ERR ;XXXXXX11 INVALID OP
F895:C9 A2 235 CMP #0A2
F897:F0 0C F8A5 236 BEQ ERR ;OPCODE $89 INVALID
F899:29 87 237 AND #087 ;MASK BITS
F89B:4A 238 IEVEN LSR A ;LSB INTO CARRY FOR L/R TEST
F89C:AA 239 TAX
F89D:BD 62 F9 240 LDA FMT1,X ;GET FORMAT INDEX BYTE
F8A0:20 79 F8 241 JSR SCR N2 ;R/L H-BYTE ON CARRY
F8A3:D0 04 F8A9 242 BNE GETFMT
F8A5:A0 80 243 ERR LDY #080 ;SUBSTITUTE $80 FOR INVALID OPS
F8A7:A9 00 244 LDA #000 ;SET PRINT FORMAT INDEX TO 0
F8A9:AA 245 GETFMT TAX
F8AA:B0 A6 F9 246 LDA FMT2,X ;INDEX INTO PRINT FORMAT TABLE
F8AD:85 2E 247 STA FORMAT ;SAVE FOR ADR FIELD FORMATTING
F8AF: 248 ; (0=1 BYTE, 1=2 BYTE, 2=3 BYTE)
F8AF: 249 *
F8AF: 250 * Move code to C1-C2 because the code
F8AF: 251 * that tests for ROM in slot 3 must be in
F8AF: 252 * the F8 ROM.
F8AF: 253 *

```