

```

1 *****
2 *
3 *           Fast Circle Generator           *
4 *
5 *           by Michael J. Mahon, Copyright 2022       *
6 *
7 * Plots an aspect ratio corrected, clipped circle with *
8 * radius 'r' centered at 'xc,yc'. The time required is *
9 * approximately 0.75r+22 milliseconds.                *
10 *
11 * POKE input parameters 'xclo', 'xchi', 'yc', and 'r' *
12 * into locations 6, 7, 8, and 9 before calling.       *
13 *
14 *****
15
16 *   Apple II ROM subroutines
17 ERROR equ   $D412   ; Print error message.
18 HPLOT0 equ  $F457   ; Plot dot at (X,Y),(A)
19 HGLIN  equ  $F53A   ; Plot line to (A,X),(Y)
20 COSTBL equ  $F5BA   ; 1-byte 1st quadrant cosines.
21 IORTS  equ  $FF58   ; Location of monitor RTS
22
23 *   Input parameters
24     dum  $06
0006: 00 25 xc      db  0      ; Center x   (0..255)
0007: 00 26 xchi   db  0      ; Center xhi (0..1)
0008: 00 27 yc      db  0      ; Center y   (0..255)
0009: 00 28 r       db  0      ; Radius     (0..214) <== NOTE!
29     dend
30
31 *   Page zero variables (swapped with Applesoft)
32     dum  $34
0034: 00 33 pz      equ  *      ; Start of page zero space.
34     x      db  0
0035: 00 35 xhi   db  0
0036: 00 36 y      db  0
0037: 00 37 yhi   db  0      ; Sign of y (00 or FF)
0038: 00 00 38 t      dw  0      ; Temp word
003A: 00 39 prevr  db  0      ; Previous call 'r'
003B: 00 40 clipping db  0      ; >0 = clipping, 0 = plotting.
003C: 00 41 savex  db  0
42     pzsize equ  *-pz   ; Size of 'pz' space.
43     dend

```

==== Page 2 ====

```

45          org    $2400      ; Chosen because $24 and $25 only
46          ; occur between 'fastcirc' and
47          ; 'plot' as hi bytes of addresses
48          ; that require relocation. NOTE:
49          ; Page number *must* be even.
50
2400: 38          51  fastcirc sec      ; Initial branch to relocation
2401: B0 6E      52          bcs    :rell      ; code (patches SEC to CLC).
2403: A5 09      53          lda    r
2405: C9 D7      54          cmp    #215      ; Is r < 215?
2407: 90 05      55          bcc    :rok      ; -Yes, r is OK.
2409: A2 35      56          ldx    #$35      ; -No, "ILLEGAL QUANTITY ERROR"
240B: 4C 12 D4   57          jmp    ERROR
58
240E: 20 73 24   59  :rok    jsr    :swappz   ; Swap pz space with 'mypz'.
2411: A0 FF      60          ldy    #$FF
2413: 84 3B      61          sty    clipping   ; Initial clipping state = true.
2415: A5 09      62          lda    r
2417: C5 3A      63          cmp    prevr      ; Is 'r' same as last call?
2419: F0 21      64          beq    :circle   ; -Yes, use cached tables.
65
66          * Construct first quadrant cosine tables scaled by
67          * radius, based on ROM 'COSTBL' at $F5BA.
68
241B: 85 3A      69          sta    prevr      ; Remember cached table 'r'.
241D: A2 0F      70          ldx    #15      ; Gen dy,x & dx,x for x = 15..0
241F: BD BA F5   71  :genloop lda    COSTBL,x
2422: A4 09      72          ldy    r
2424: 20 83 24   73          jsr    :mpyay      ; A.t = r * COSTBL,x
2427: 06 38      74          asl    t          ; Round dy.
2429: 69 00      75          adc    #0
242B: 9D 3A 25   76          sta    dy,x      ; dy,x = int(r * COSTBL,x + .5)
77
78          * Adjust 'dx' for Apple II pixel aspect ratio, computed by
79          * Sather in "Understanding the Apple II" as X:Y = 1.19:1
80          * on page 8-28. Here this is approximated by:
81          *      1 + (1/8 + 1/16) = 1.1875
82
242E: A0 30      83  :xfactor ldy    #$30      ; Y = 0.1875 (can be POKEd for
2430: 20 83 24   84          jsr    :mpyay      ; different pixel aspect ratio)
2433: 7D 3A 25   85          adc    dy,x
2436: 9D 2A 25   86          sta    dx,x      ; dx = int(dy * 1.1875 + .5)
2439: CA          87          dex
243A: 10 E3      88          bpl    :genloop
89
90          * Generate 64 points on circle, connected by lines.
91
243C: A2 00      92  :circle ldx    #0
243E: 20 9A 24   93  :q1    jsr    :adddx
2441: 20 BA 24   94          jsr    :adddy
2444: E8          95          inx
2445: E0 10      96          cpx    #16      ; Done with 1st quadrant?
2447: D0 F5      97          bne    :q1      ; -No, continue.
2449: CA          98          dex          ; -Yes, reset X to 15.
244A: 20 A9 24   99  :q2    jsr    :subdx
244D: 20 BA 24  100          jsr    :adddy
2450: CA          101          dex          ; Done with 2nd quadrant?
2451: 10 F7      102          bpl    :q2      ; -No, continue.
2453: E8          103          inx          ; -Yes, reset X to 0.
2454: 20 A9 24  104  :q3    jsr    :subdx
2457: 20 D1 24  105          jsr    :subdy
245A: E8          106          inx
245B: E0 10      107          cpx    #16      ; Done with 3rd quadrant?
245D: D0 F5      108          bne    :q3      ; -No, continue.
245F: CA          109          dex          ; -Yes, reset X to 15.
2460: 20 9A 24  110  :q4    jsr    :adddx
2463: 20 D1 24  111          jsr    :subdy
```

```

2466: CA      112      dex          ; Done with 4th quadrant?
2467: 10 F7   113      bpl      :q4    ; -No, continue.
2469: E8      114      inx          ; -Yes, close circle by
246A: 20 9A 24 115      jsr      :adddx ;          plotting to 1st point.
246D: 20 BA 24 116      jsr      :addy   ; Plot final point and
                               117      ; fall into 'swappz'.
                               118
2470: 18      119      clc          ; Fall through relay branch.
2471: B0 45   120 :rell     bcs      :rel2 ; Relay branch
                               121
2473: A2 08   122 :swappz   ldx      #pzsize-1 ; Swap 'pz' with 'mypz'.
2475: B5 34   123 :swap     lda      pz,x
2477: BC 21 25 124          ldy      mypz,x
247A: 94 34   125          sty      pz,x
247C: 9D 21 25 126          sta      mypz,x
247F: CA      127      dex
2480: 10 F3   128      bpl      :swap
2482: 60      129      rts
                               130
                               131 * 8x8 multiply subroutine
                               132
2483: 85 38   133 :mpyay    sta      t          ; On entry: A = multiplicand
2485: 8C 92 24 134          sty      :plier+1 ;          Y = multiplier
2488: A9 00   135          lda      #0        ; On exit: A = hi product
248A: A0 08   136          ldy      #8        ;          t = lo product
248C: 46 38   137          lsr      t
248E: 90 03   138 :mloop    bcc      :noadd
2490: 18      139          clc
2491: 69 00   140 :plier    adc      #0-0      ; (Multiplier stored here)
2493: 6A      141 :noadd    ror
2494: 66 38   142          ror      t
2496: 88      143          dey
2497: D0 F5   144          bne      :mloop
2499: 60      145          rts
                               146
249A: 18      147 :adddx    clc          ; Add dx to xc.
249B: A5 06   148          lda      xc
249D: 7D 2A 25 149          adc      dx,x
24A0: 85 34   150          sta      x
24A2: A5 07   151          lda      xchi
24A4: 69 00   152          adc      #0
24A6: 85 35   153          sta      xhi
24A8: 60      154          rts
                               155
24A9: 38      156 :subdx    sec          ; Subtract dx from xc.
24AA: A5 06   157          lda      xc
24AC: FD 2A 25 158          sbc      dx,x
24AF: 85 34   159          sta      x
24B1: A5 07   160          lda      xchi
24B3: E9 00   161          sbc      #0
24B5: 85 35   162          sta      xhi
24B7: 60      163          rts
                               164
24B8: B0 67   165 :rel2     bcs      relocate ; Relay branch
                               166
24BA: 86 3C   167 :addy     stx      savex
24BC: 8A      168          txa          ; X = 15 - X
24BD: 49 0F   169          eor      #$0F
24BF: AA      170          tax
24C0: 18      171          clc          ; y = yc + dy
24C1: A5 08   172          lda      yc
24C3: 7D 3A 25 173          adc      dy,x
24C6: 85 36   174          sta      y
24C8: A9 00   175          lda      #0
24CA: 69 00   176          adc      #0
24CC: 85 37   177          sta      yhi
24CE: 18      178          clc

```

```

24CF: 90 14      179      bcc  plot      ; (always)
                               180
24D1: 86 3C      181  :subdy stx  savex
24D3: 8A         182      txa              ; X = 15 - X
24D4: 49 0F      183      eor  #$0F
24D6: AA         184      tax
24D7: 38         185      sec              ; y = yc - dy
24D8: A5 08      186      lda  yc
24DA: FD 3A 25   187      sbc  dy,x
24DD: 85 36      188      sta  y
24DF: A9 00      189      lda  #0
24E1: E9 00      190      sbc  #0
24E3: 85 37      191      sta  yhi         ; Fall into 'plot'.
                               192
24E5: A4 35      193  plot  ldy  xhi         ; Plot (with clipping)
24E7: F0 09      194      beq  :noclipx    ; No clip if X < 256.
24E9: 88         195      dey
24EA: D0 10      196      bne  :clip       ; Clip if xhi <> 1.
24EC: A0 17      197      ldy  #280-1-256
24EE: C4 34      198      cpy  x
24F0: 90 0A      199      bcc  :clip       ; Clip if X > 279.
24F2: A4 37      200  :noclipx ldy  yhi         ; y < 0 or y > 255?
24F4: D0 06      201      bne  :clip       ; -Yes, clip.
24F6: A0 BF      202      ldy  #192-1     ; -No, is Y in range?
24F8: C4 36      203      cpy  y
24FA: B0 05      204      bcs  :noclip
24FC: 84 3B      205  :clip  sty  clipping ; Remember we're clipping,
24FE: A6 3C      206      ldx  savex       ; restore X,
2500: 60         207      rts              ; and return.
                               208
2501: A0 00      209  :noclip ldy  #0
2503: A5 3B      210      lda  clipping    ; Were we clipping?
2505: 84 3B      211      sty  clipping    ; (We aren't any more!)
2507: D0 0C      212      bne  :point      ; -Yes, HPLOT X,Y.
2509: A6 35      213      ldx  xhi         ; -No, HPLOT TO X,Y.
250B: A5 34      214      lda  x
250D: A4 36      215      ldy  y
250F: 20 3A F5   216      jsr  HGLIN       ; Plot line to X,Y,
2512: A6 3C      217      ldx  savex       ; restore X reg,
2514: 60         218      rts              ; and return.
                               219
2515: A4 35      220  :point  ldy  xhi         ; -Yes, HPLOT X,Y.
2517: A6 34      221      ldx  x
2519: A5 36      222      lda  y
251B: 20 57 F4   223      jsr  HPLOT0     ; Plot point at X,Y,
251E: A6 3C      224      ldx  savex       ; restore X reg,
2520: 60         225      rts              ; and return.
                               226
                               227      dum  *           ; (Overwrites relocation code)
2521: 00 00 00   228  mypz  ds    pzsize    ; Page zero swap space
2524: 00 00 00 00 00 00
252A: 00 00 00   229  dx     ds    16
252D: 00 00 00 00 00 00 00 00 00
2535: 00 00 00 00 00
253A: 00 00 00   230  dy     ds    16
253D: 00 00 00 00 00 00 00 00 00
2545: 00 00 00 00 00
                               231      dend

```

```

233 * Self-relocation code executed on first call.
234
235 offset equ $FB ; Page zero unused by Applesoft
236 loadadr equ $FD ; so safe before page 0 swap.
237 asmpage equ fastcirc/256 ; Unique page ==> abs address.
238 CLC_op equ $18 ; CLC to turn off relocation.
239
2521: 20 58 FF 240 relocate jsr IORTS ; Get load address
2524: BA 241 :asm tsx
2525: EA 242 nop ; Align $FF in following 'ldaa'
2526: EA 243 nop ; to cause 'swappz' to init
2527: EA 244 nop ; 'prevr' to $FF.
2528: BD FF 00 245 ldaa $100-1,x ; Lo byte of return address
252B: 38 246 sec
252C: E9 23 247 sbc #<:asm-1 ; Offset = Actual adr - asm adr
252E: 85 FB 248 sta offset
2530: BD 00 01 249 lda $100,x
2533: E9 25 250 sbc #>:asm-1
2535: 85 FC 251 sta offset+1
2537: A5 FB 252 lda offset
2539: 85 FD 253 sta loadadr
253B: 18 254 clc
253C: A9 24 255 lda #>plot ; Compute 'plot' page.
253E: 65 FC 256 adc offset+1
2540: 85 FE 257 sta loadadr+1
2542: 18 258 clc
2543: A9 24 259 lda #>fastcirc
2545: 65 FC 260 adc offset+1
2547: AA 261 tax ; X = Load page of 'fastcirc'.
2548: A0 E5 262 ldy #<plot ; Start reverse scan at 'plot'.
254A: B1 FD 263 :reloc lda (loadadr),y
254C: 29 FE 264 and #$FE ; Turn off low bit first.
254E: C9 24 265 cmp #asmpage ; Unique abs addr hi bytes?
2550: D0 0F 266 bne :noreloc ; -No, don't relocate.
2552: 88 267 dey ; -Yes, point to lo addr byte
2553: 18 268 clc ; and relocate address.
2554: B1 FD 269 lda (loadadr),y
2556: 65 FB 270 adc offset
2558: 91 FD 271 sta (loadadr),y
255A: C8 272 iny
255B: B1 FD 273 lda (loadadr),y
255D: 65 FC 274 adc offset+1
255F: 91 FD 275 sta (loadadr),y
2561: 88 276 :noreloc dey
2562: C0 FF 277 cpy #$FF ; Done relocating page?
2564: D0 E4 278 bne :reloc ; -No, keep going.
2566: E4 FE 279 cpx loadadr+1 ; First (final) page?
2568: F0 04 280 beq :done ; -Yes, finish.
256A: C6 FE 281 dec loadadr+1 ; -No, do next lower page.
256C: D0 DC 282 bne :reloc ; (always)
283
256E: A9 18 284 :done lda #CLC_op ; Close 'relocate' door.
2570: C8 285 iny ; Y = 0
2571: 91 FD 286 sta (loadadr),y
2573: 6C FD 00 287 jmp (loadadr) ; and draw first circle!

```

--End assembly, 374 bytes, Errors: 0

Symbol table - alphabetical order:

CLC_op	=\$18	COSTBL	=\$F5BA	ERROR	=\$D412	HGLIN	=\$F53A
HPLOT0	=\$F457	IORTS	=\$FF58	asmpage	=\$24	clipping	=\$3B
dx	=\$252A	dy	=\$253A	fastcirc	=\$2400	loadadr	=\$FD
mypz	=\$2521	offset	=\$FB	plot	=\$24E5	prevr	=\$3A
pz	=\$34	pzsize	=\$09	r	=\$09	relocate	=\$2521
savex	=\$3C	t	=\$38	x	=\$34	xc	=\$06
xchi	=\$07	xhi	=\$35	y	=\$36	yc	=\$08
yhi	=\$37						

Symbol table - numerical order:

xc	=\$06	xchi	=\$07	yc	=\$08	r	=\$09
pzsize	=\$09	CLC_op	=\$18	asmpage	=\$24	pz	=\$34
x	=\$34	xhi	=\$35	y	=\$36	yhi	=\$37
t	=\$38	prevr	=\$3A	clipping	=\$3B	savex	=\$3C
offset	=\$FB	loadadr	=\$FD	fastcirc	=\$2400	plot	=\$24E5
mypz	=\$2521	relocate	=\$2521	dx	=\$252A	dy	=\$253A
ERROR	=\$D412	HPLOT0	=\$F457	HGLIN	=\$F53A	COSTBL	=\$F5BA
IORTS	=\$FF58						