

```

1 *****
2 *
3 *           F P E P A T C H           *
4 *
5 *           Michael J. Mahon         *
6 *           Sep 21, 2010             *
7 *           Copyright 2010          *
8 *
9 * Patch Applesoft BASIC (FPBASIC) to use FP replacement *
10 * routines using the Floating Point Engine.             *
11 *
12 * Computes ARG op FAC --> FAC for dyadic FP ops, and  *
13 * fn (FAC) --> FAC for monadic functions.              *
14 *
15 *****
16
17 FPEslot equ 4           ; FPE in slot 4
18 * 'fptr' not defined; should not be referenced!
19
20
21
22
23
24
25
26 * Applesoft Page Zero Definitions
27
28 extra    equ  $03      ; Temp for extra byte
29 FAC      equ  $9D      ; Floating-Point Accumulator
30 FACSGN   equ  FAC+5    ; FAC unpacked sign
31 EXTRAFAC equ  $AC      ; FAC guard byte
32 ARG      equ  $A5      ; Secondary FP accumulator
33
34 * Constants
35
36 fbias    equ  128      ; Applesoft FP exponent bias
37 Xbias    equ  $3FFE    ; IEEE FP extended exp bias
38
39 * Apple ROM routines
40
41 IQERR    equ  $E199    ; "Illegal quantity error"
42 ZEROFAC  equ  $E84E    ; Zero out FAC (underflow)
43 OVERFLOW equ  $E8D5    ; "Overflow error"
44
45 * Patcher page zero locations
46
47 src      equ  6         ; Ptr to patch code
48 targ     equ  src+2     ; Ptr to RAM image
49 segptr   equ  $FC      ; Ptr to segtbl entry
50
51 * Patcher RAM addresses
52
53 patcher  equ  $2000     ; RAM addr of this program
54 FPBASIC  equ  $D000     ; ROM addr of FPBASIC
55 FP       equ  $3000     ; RAM addr of FPBASIC
56 targadj  equ  FPBASIC-FP ; Offset to RAM buf
57
58 * Patch segment macros
59
60 segm     mac  &romadsym
61 ]lc      equ  *
62 ]nsegs   equ  ]nsegs+1
63 ]org     org  ]1
64 ]start   equ  *
65 ]eom     eom
66
67 eseg     mac
68 ]len     equ  *-]start
69 ]org     org  ]lc+]]len
70 ]eom     eom ; *****
71
72 segtb    mac  &segname;&romadsym
73 ]da      da  ]1         ; RAM address
74 ]da      da  ]2         ; ROM address

```

==== Page 2 ====

```
75      db      e]2-]2      ; Length  
76      eom                ; *****
```

==== Page 3 ====

```

79          org    patcher
80 ]nsegs   equ    0          ; Initial segment counter
81
82 * Patch FPBASIC with patch segments below
83
2000: A9 58 84 fpepatch lda    #<segtbl+1 ; Point to first src, targ
2002: 85 FC 85          sta    segptr
2004: A9 21 86          lda    #>segtbl+1
2006: 85 FD 87          sta    segptr+1
2008: A0 00 88 :nxseg  ldy    #0          ; Set up src & targ
200A: B1 FC 89 :mvptr  lda    (segptr),y ; pointers.
200C: 99 06 00 90          sta    src,y
200F: C8      91          iny
2010: C0 04 92          cpy    #4
2012: D0 F6 93          bne    :mvptr
2014: 38      94          sec          ; Adjust targ to RAM buf
2015: A5 09 95          lda    targ+1
2017: E9 A0 96          sbc    #>targadj
2019: 85 09 97          sta    targ+1
201B: B1 FC 98          lda    (segptr),y ; Segment length
201D: A8      99          tay
201E: 88      100         dey          ; length - 1
201F: B1 06 101 :mvseg  lda    (src),y   ; Move the segment
2021: 91 08 102          sta    (targ),y
2023: 88      103         dey
2024: C0 FF 104         cpy    #$FF
2026: D0 F7 105         bne    :mvseg
2028: CE 57 21 106        dec    segtbl   ; More segments?
202B: F0 0F 107         beq    :exit    ; -No, quit.
202D: 18      108         clc          ; -Yes, advance segptr
202E: A5 FC 109         lda    segptr
2030: 69 05 110         adc    #5
2032: 85 FC 111         sta    segptr
2034: A5 FD 112         lda    segptr+1
2036: 69 00 113         adc    #0
2038: 85 FD 114         sta    segptr+1
203A: D0 CC 115         bne    :nxseg   ; (always)
      116
203C: 60      117 :exit  rts
```

```

119 *****
120 *                               Patch segments                               *
121 *****
122
123 * FPBASIC entry points to patch
124
125 FSUBT    equ    $E7AA          ; ARG - FAC  --> FAC
126 FADDT    equ    $E7C1          ; ARG + FAC  --> FAC
127 LOG      equ    $E94B          ; ln (FAC>0)  --> FAC
128 FMU      equ    $E987          ; ARG * FAC  --> FAC
129 FDIVT    equ    $EA6B          ; ARG / FAC<>0) --> FAC
130 SQRT     equ    $EE8D          ; sqrt (FAC) --> FAC  (>=0)
131 EXP      equ    $EF09          ; e^FAC      --> FAC
132 RND      equ    $EFAE          ; random     --> FAC
133 COS      equ    $EFEA          ; cos (FAC)  --> FAC
134 SIN      equ    $EFF1          ; sin (FAC)  --> FAC
135 TAN      equ    $F03A          ; tan (FAC)  --> FAC (in SIN)
136 ATAN     equ    $F09E          ; atan (FAC) --> FAC (in SIN)
137
138 subseg   segm   FSUBT
138 ]lc      equ    *
138 ]nsegs   equ    ]nsegs+1
138 ]start   org    FSUBT
138 ]start   equ    *
138          eom
E7AA: A9 28 139          lda    #$28
E7AC: 4C 66 F0 140          jmp    dyadic
141 eFSUBT   eseg
141 ]len     equ    *-]start
141          org    ]lc+]len
141          eom ; *****
142
143 addseg   segm   FADDT
143 ]lc      equ    *
143 ]nsegs   equ    ]nsegs+1
143          org    FADDT
143 ]start   equ    *
143          eom
E7C1: A9 22 144          lda    #$22
E7C3: 4C 66 F0 145          jmp    dyadic
146 eFADDT   eseg
146 ]len     equ    *-]start
146          org    ]lc+]len
146          eom ; *****
147
148 logseg   segm   LOG
148 ]lc      equ    *
148 ]nsegs   equ    ]nsegs+1
148          org    LOG
148 ]start   equ    *
148          eom
E94B: A9 14 149          lda    #$14
E94D: 4C F3 EF 150          jmp    monadic
151 eLOG     eseg
151 ]len     equ    *-]start
151          org    ]lc+]len
151          eom ; *****
152
153 mulseg   segm   FMU
153 ]lc      equ    *
153 ]nsegs   equ    ]nsegs+1
153          org    FMU
153 ]start   equ    *
153          eom
E987: A9 23 154          lda    #$23
E989: 4C 66 F0 155          jmp    dyadic
156 eFMU     eseg

```

```

156 ]len      equ    *-]start
156          org    ]lc+]len
156          eom          ; *****
157
158 divseg    segm   FDIVT
158 ]lc      equ    *
158 ]nsegs   equ    ]nsegs+1
158          org    FDIVT
158 ]start    equ    *
158          eom
EA6B: A9 20   159          lda    #$20
EA6D: 4C 66 F0 160          jmp    dyadic
161 eFDIVT    eseg
161 ]len      equ    *-]start
161          org    ]lc+]len
161          eom          ; *****
162
163 sqrseg    segm   SQRT
163 ]lc      equ    *
163 ]nsegs   equ    ]nsegs+1
163          org    SQRT
163 ]start    equ    *
163          eom
EE8D: A5 A2   164          lda    FACSGN    ; FAC < 0?
EE8F: 30 7D   165          bmi    IQER      ; -Yes, Illegal Quantity!
EE91: A9 04   166 :ok      lda    #$04
EE93: 4C F3 EF 167          jmp    monadic
168 eSQRT     eseg
168 ]len      equ    *-]start
168          org    ]lc+]len
168          eom          ; *****
169
170 expseg    segm   EXP
170 ]lc      equ    *
170 ]nsegs   equ    ]nsegs+1
170          org    EXP
170 ]start    equ    *
170          eom
EF09: A9 10   171          lda    #$10
EF0B: 4C F3 EF 172          jmp    monadic
173
EF0E: 4C 99 E1 174 IQER      jmp    IQERR      ; Relay jmp from SQRT
175 eEXP      eseg
175 ]len      equ    *-]start
175          org    ]lc+]len
175          eom          ; *****
176
177 cosseg    segm   COS
177 ]lc      equ    *
177 ]nsegs   equ    ]nsegs+1
177          org    COS
177 ]start    equ    *
177          eom
EFEA: A9 1D   178          lda    #$1D
EFEC: 4C F3 EF 179          jmp    monadic
180 eCOS      eseg
180 ]len      equ    *-]start
180          org    ]lc+]len
180          eom          ; *****
181
182 sinseg    segm   SIN      ; (also includes TAN and ATAN)
182 ]lc      equ    *
182 ]nsegs   equ    ]nsegs+1
182          org    SIN
182 ]start    equ    *
182          eom
EFF1: A9 0E   183          lda    #$0E

```

```

EFF3: A6 AC      184 monadic ldx  EXTRAFAC  ; Move EXTRAFAC
EFF5: 86 03      185         stx  extra    ; for ZPX2FR0.
EFF7: A2 9D      186         ldx  #FAC     ; Aop (FAC) --> FAC
EFF9: 20 74 F0   187         jsr  ZPX2FR0  ; Aop (FAC) to FR0
EFFC: A0 68      188 FR02FAC ldy  #>$6000+fX-fF ; Fmove FR0 to FAC
EF FE: 8C C8 C0  189         sty  FPcmd
F001: A0 00      190         ldy  #0       ; (No kfactor)
F003: 8C C9 C0   191         sty  FPcmd+1
                192         Fwtdata      ; Wait for data transfer
F006: AD C0 C0   192 wait   lda  FPresp   ; Check FPE status.
F009: AC C1 C0   192         ldy  FPresp+1
F00C: D0 04      192         bne  done    ; Ready if FPresp <> $8900
F00E: C9 89      192         cmp  #$89
F010: F0 F4      192         beq  wait    ; else keep waiting...
                192 done   eom
F012: AD CC C0   193         lda  FPopnd   ; Sign + hi exponent
F015: 85 A2      194         sta  FACSGN
F017: AD CD C0   195         lda  FPopnd+1 ; lo exponent
F01A: 38         196         sec          ; Adjust bias
F01B: E9 7E      197         sbc  #<Xbias-fbias
F01D: 85 9D      198         sta  FAC     ; Save adjusted exponent.
F01F: A5 A2      199         lda  FACSGN   ; Recover hi exp
F021: 29 7F      200         and  #$7F    ; without mant sign.
F023: E9 3F      201         sbc  #>Xbias-fbias
F025: 08         202         php          ; Save exponent flags.
F026: AD CE C0   203         lda  FPopnd+2 ; Discard 2 zero
F029: AD CF C0   204         lda  FPopnd+3 ; bytes.
F02C: AD CC C0   205         lda  FPopnd   ; Store mantissa
F02F: 85 9E      206         sta  FAC+1
F031: AD CD C0   207         lda  FPopnd+1
F034: 85 9F      208         sta  FAC+2
F036: 4C 3F F0   209         jmp  sktan    ; Jump past TAN entry.
                210
                211 ]v     equ  TAN+1
                212         err  */]v   ; Can't be past TAN.
F039: 00         213         ds    TAN-*   ; Pad to TAN entry.
F03A: A9 0F      214 tan    lda  #$0F
F03C: 4C F3 EF   215         jmp  monadic
                216
F03F: AD CE C0   217 sktan  lda  FPopnd+2
F042: 85 A0      218         sta  FAC+3
F044: AD CF C0   219         lda  FPopnd+3
F047: 85 A1      220         sta  FAC+4
F049: AD CC C0   221         lda  FPopnd   ; Save extra byte
F04C: 85 AC      222         sta  EXTRAFAC
F04E: AD CD C0   223         lda  FPopnd+1 ; Discard low 3 bytes
F051: AD CE C0   224         lda  FPopnd+2
F054: AD CF C0   225         lda  FPopnd+3
F057: 28         226         plp          ; Restore exponent flags.
F058: 90 06      227         bcc  unflow   ; Exponent underflow
F05A: D0 07      228         bne  overflow ; Exponent overflow
F05C: A5 9D      229         lda  FAC     ; Set flags from FAC
F05E: 18         230         clc
F05F: 60         231         rts          ; Return.
                232
F060: 4C 4E E8   233 unflow jmp  ZEROFAC  ; Flush underflow to zero.
F063: 4C D5 E8   234 overflow jmp  OVERFLOW  ; Overflow signals error.
                235
F066: 48         236 dyadic pha          ; Save dyadic op
F067: A9 00      237         lda  #0       ; Clear "EXTRA" byte
F069: 85 03      238         sta  extra    ; (A) = 0 = Fmove
F06B: A2 A5      239         ldx  #ARG
F06D: 20 74 F0   240         jsr  ZPX2FR0  ; ARG --> FR0
F070: 68         241         pla          ; recover dyadic
F071: 4C F3 EF   242         jmp  monadic  ; op and do it.
                243
                244 * Load FAC or ARG (pointed to by X) to FPE's FR0

```

```

245 * (X+6 must be set to EXTRAFAAC or 0.)
246
F074: A0 48 247 ZPX2FR0 ldy #>$4000+fX-fF ; MR op, X fmt --> FR0
F076: 8C C8 C0 248 sty FPCmd
F079: 8D C9 C0 249 sta FPCmd+1 ; A = op
250 Fwtdata ; Wait for data xfer
F07C: AD C0 C0 250 wait lda FPresp ; Check FPE status.
F07F: AC C1 C0 250 ldy FPresp+1
F082: D0 04 250 bne done ; Ready if FPresp <> $8900
F084: C9 89 250 cmp #$89
F086: F0 F4 250 beq wait ; else keep waiting...
251 done eom
F088: B5 00 251 lda 0,x ; exp low byte
F08A: 18 252 clc
F08B: 69 7E 253 adc #<Xbias-fbias ; correct exp bias
F08D: 48 254 pha
F08E: A9 00 255 lda #0
F090: 69 3F 256 adc #>Xbias-fbias
F092: 0A 257 asl ; Insert mant sign
F093: 36 05 258 rol 5,x
F095: 6A 259 ror
F096: 8D CC C0 260 sta FPopnd ; Sign & high exp
F099: 68 261 pla
F09A: 4C A3 F0 262 jmp skatn ; Skip ATAN entry.
263
264 ]v equ ATAN+1
265 err */]v ; Can't be past ATAN.
F09D: 00 266 ds ATAN-* ; Pad to ATAN entry.
F09E: A9 0A 267 atn lda #$0A
F0A0: 4C F3 EF 268 jmp monadic
269
F0A3: 8D CD C0 270 skatn sta FPopnd+1 ; low exp
F0A6: A9 00 271 lda #0
F0A8: 8D CE C0 272 sta FPopnd+2 ; 2 zero bytes
F0AB: 8D CF C0 273 sta FPopnd+3
F0AE: B5 01 274 lda 1,x ; Load mantissa
F0B0: 8D CC C0 275 sta FPopnd
F0B3: B5 02 276 lda 2,x
F0B5: 8D CD C0 277 sta FPopnd+1
F0B8: B5 03 278 lda 3,x
F0BA: 8D CE C0 279 sta FPopnd+2
F0BD: B5 04 280 lda 4,x
F0BF: 8D CF C0 281 sta FPopnd+3
F0C2: A5 03 282 lda extra ; EXTRAFAAC or 0 (if ARG)
F0C4: 8D CC C0 283 sta FPopnd
F0C7: A9 00 284 lda #0
F0C9: 8D CD C0 285 sta FPopnd+1 ; Zero final 3 bytes
F0CC: 8D CE C0 286 sta FPopnd+2 ; of FPE mantissa.
F0CF: 8D CF C0 287 sta FPopnd+3
288 Fwait ; wait for completion
F0D2: AD C0 C0 288 wait lda FPresp ; Keep waiting until
F0D5: 0A 288 asl
F0D6: AD C1 C0 288 lda FPresp+1
F0D9: B0 F7 288 bcs wait ; FPresp hi bit off.
288 eom
F0DB: 60 289 rts
290 eSIN eseg
290 ]len equ *-]start
290 org ]lc+]len
290 eom ; *****

```

```

2157: 09      292  segtbl  db    ]nsegs      ; [cnt]
                293      segtb  subseg;FSUBT ; [locl.addr][ROM .addr][len]
2158: 3D 20   293      da    subseg      ; RAM address
215A: AA E7   293      da    FSUBT       ; ROM address
215C: 05      293      db    eFSUBT-FSUBT ; Length
                293      eom                    ; *****
                294      segtb  addseg;FADDT
215D: 42 20   294      da    addseg      ; RAM address
215F: C1 E7   294      da    FADDT       ; ROM address
2161: 05      294      db    eFADDT-FADDT ; Length
                294      eom                    ; *****
                295      segtb  logseg;LOG
2162: 47 20   295      da    logseg      ; RAM address
2164: 4B E9   295      da    LOG          ; ROM address
2166: 05      295      db    eLOG-LOG    ; Length
                295      eom                    ; *****
                296      segtb  mulseg;FMU
2167: 4C 20   296      da    mulseg      ; RAM address
2169: 87 E9   296      da    FMU          ; ROM address
216B: 05      296      db    eFMU-FMU    ; Length
                296      eom                    ; *****
                297      segtb  divseg;FDIVT
216C: 51 20   297      da    divseg      ; RAM address
216E: 6B EA   297      da    FDIVT       ; ROM address
2170: 05      297      db    eFDIVT-FDIVT ; Length
                297      eom                    ; *****
                298      segtb  sqrseg;SQRT
2171: 56 20   298      da    sqrseg      ; RAM address
2173: 8D EE   298      da    SQRT         ; ROM address
2175: 09      298      db    eSQRT-SQRT  ; Length
                298      eom                    ; *****
                299      segtb  expseg;EXP
2176: 5F 20   299      da    expseg      ; RAM address
2178: 09 EF   299      da    EXP          ; ROM address
217A: 08      299      db    eEXP-EXP    ; Length
                299      eom                    ; *****
                300      segtb  cosseg;COS
217B: 67 20   300      da    cosseg      ; RAM address
217D: EA EF   300      da    COS          ; ROM address
217F: 05      300      db    eCOS-COS    ; Length
                300      eom                    ; *****
                301      segtb  sinseg;SIN ; (includes TAN & ATAN)
2180: 6C 20   301      da    sinseg      ; RAM address
2182: F1 EF   301      da    SIN          ; ROM address
2184: EB      301      db    eSIN-SIN    ; Length
                301      eom                    ; *****

```

--End assembly, 389 bytes, Errors: 0

Symbol table - alphabetical order:

ARG	=\$A5	ATAN	=\$F09E	COS	=\$EFEA	EXP	=\$EF09
EXTRAFAC	=\$AC	? F10t0	=\$32	? F10t1	=\$33	? F10t1024	=\$3D
? F10t128	=\$3A	? F10t16	=\$37	? F10t2	=\$34	? F10t2048	=\$3E
? F10t256	=\$3B	? F10t32	=\$38	? F10t4	=\$35	? F10t4096	=\$3F
? F10t512	=\$3C	? F10t64	=\$39	? F10t8	=\$36	FAC	=\$9D
FACSGN	=\$A2	FADDT	=\$E7C1	? FCcnt1	=\$04	? FCstat	=\$02
FDIVT	=\$EA6B	FMU	=\$E987	FP	=\$3000	FPBASIC	=\$D000
FPE	=\$C0C0	FPslot	=\$04	? FPNaN	=\$01	FPcmd	=\$C0C8
? FPcond	=\$C0CA	? FPctrl	=\$C0C2	? FPdz	=\$04	? FPdzA	=\$10
? FPinexA	=\$08	? FPinexd	=\$01	? FPinexop	=\$02	? FPinf	=\$02
? FPinop	=\$20	? FPinvA	=\$80	? FPneg	=\$08	FPopnd	=\$C0CC
? FPovfl	=\$10	? FPovflA	=\$40	? FPqsign	=\$80	FPresp	=\$C0C0
? FPrst	=\$C0C6	? FPrnear	=\$00	? FPrninf	=\$20	? FPrpD	=\$80
? FPrpS	=\$40	? FPrpX	=\$00	? FPrpinf	=\$30	? FPrzero	=\$10

? FPsNaN = \$40	? FPsave = \$C0C4	? FPunfl = \$08	? FPunfla = \$20
? FPzero = \$04	? FR0 = \$00	? FR02FAC = \$E7FC	? FR1 = \$01
? FR2 = \$02	? FR3 = \$03	? FR4 = \$04	? FR5 = \$05
? FR6 = \$06	? FR7 = \$07	FSUBT = \$E7AA	MD?Fabs = \$8000
MD?Facos = \$8000	MD?Fadd = \$8000	MD?Fasin = \$8000	MD?Fatan = \$8000
MD?Fatanh = \$8000	MD?Fcmp = \$8000	MD?Fcond = \$8000	MD?Fcos = \$8000
MD?Fcosh = \$8000	MD?Fdiv = \$8000	? Fe = \$0C	MD?Fetox = \$8000
MD?Fetoxml = \$8000	MD?Fgetexp = \$8000	MD?Fgetman = \$8000	MD?Fidx = \$8000
MD?Fint = \$8000	MD?Fldcs = \$8000	MD?Fln = \$8000	? Fln10 = \$31
? Fln2 = \$30	MD?Flnp1 = \$8000	MD?Flog10 = \$8000	? Flog102 = \$0B
? Flog10e = \$0E	MD?Flog2 = \$8000	? Flog2e = \$0D	MD?Fmod = \$8000
MD?Fmove = \$8000	MD?Fmul = \$8000	MD?Fneg = \$8000	? Fpi = \$00
MD?Fprint = \$8000	MD?Frem = \$8000	MD?Freset = \$8000	MD?From = \$8000
MD?Fscale = \$8000	MD?Fsin = \$8000	MD?Fsincos = \$8000	MD?Fsinh = \$8000
MD?Fsqrt = \$8000	MD?Fstcs = \$8000	MD?Fsub = \$8000	MD?Ftan = \$8000
MD?Ftanh = \$8000	MD?Ftentox = \$8000	MD?Ftst = \$8000	MD?Ftwotox = \$8000
MD Fwait = \$8000	MD Fwtdata = \$8000	? Fzero = \$0F	IQER = \$E70E
IQERR = \$E199	LOG = \$E94B	OVERFLOW = \$E8D5	? RND = \$EFAE
SIN = \$E7F1	SQRT = \$EE8D	TAN = \$F03A	Xbias = \$3FFE
ZEROFAC = \$E84E	ZPX2FR0 = \$F074	V?]Fdebug = \$00	V]Fdeflt = \$00
V?]FfCOND = \$00	V?]FfIDX = \$00	V?]FfMR = \$00	V?]FfPRINT = \$00
V?]FfRM = \$00	V?]FfRR = \$00	V?]Fkfac = \$11	MV]lc = \$206C
MV]len = \$EB	V]nsegs = \$09	MV]start = \$E7F1	V]v = \$F09F
addseg = \$2042	? atn = \$F09E	cosseg = \$2067	divseg = \$2051
M done = \$F088	dyadic = \$F066	eCOS = \$E7EF	eEXP = \$E711
eFADDT = \$E7C6	eFDIVT = \$EA70	eFMU = \$E98C	eFSUBT = \$E7AF
eLOG = \$E950	eSIN = \$F0DC	eSQRT = \$EE96	MD eseg = \$8000
expseg = \$205F	extra = \$03	? fB = \$180F	? fBb = \$40
? fBl = \$01	? fCb = \$80	? fCl = \$04	? fD = \$140F
? fDb = \$20	? fDl = \$08	fF = \$0F	MD?fFPTR = \$8000
? fL = \$0F	? fLb = \$01	? fLl = \$04	MD?fMR = \$8000
MD?fOP = \$8000	MD?fOPC = \$8000	M? fOpcod = \$8000	? fP = \$0C0F
? fPb = \$08	? fPl = \$0C	MD?fRR = \$8000	? fS = \$040F
? fSb = \$02	? fSl = \$04	MD?fVECT = \$8000	? fW = \$100F
? fWb = \$10	? fWl = \$02	fX = \$080F	? fXb = \$04
? fXl = \$0A	fbias = \$80	? fpepatch = \$2000	logseg = \$2047
monadic = \$E7F3	mulseg = \$204C	ovflow = \$F063	patcher = \$2000
MD segm = \$8000	segptr = \$FC	MD segtb = \$8000	segtbl = \$2157
sinseg = \$206C	skatn = \$F0A3	sktan = \$F03F	sqrseg = \$2056
src = \$06	subseg = \$203D	? tan = \$F03A	targ = \$08
targadj = \$A000	unflow = \$F060	M wait = \$F0D2	

Symbol table - numerical order:

? FR0 = \$00	? Fpi = \$00	? FPrnear = \$00	? FPrpX = \$00
V]Fdeflt = \$00	V?]FfRR = \$00	V?]FfMR = \$00	V?]FfRM = \$00
V?]FfCOND = \$00	V?]FfIDX = \$00	V?]FfPRINT = \$00	V?]Fdebug = \$00
? fBl = \$01	? FR1 = \$01	? FPNaN = \$01	? FPinexd = \$01
? fLb = \$01	? fWl = \$02	? FR2 = \$02	? FCstat = \$02
? FPinf = \$02	? FPinexop = \$02	? fSb = \$02	? FR3 = \$03
extra = \$03	FPESlot = \$04	? fLl = \$04	? fSl = \$04
? fCl = \$04	? FR4 = \$04	? FPzero = \$04	? fPdZ = \$04
? FCentl = \$04	? fXb = \$04	? FR5 = \$05	? FR6 = \$06
src = \$06	? FR7 = \$07	? fDl = \$08	? FPneg = \$08
? FPunfl = \$08	? FPinexA = \$08	? fPb = \$08	targ = \$08
V]nsegs = \$09	? fXl = \$0A	? Flog102 = \$0B	? fPl = \$0C
? Fe = \$0C	? Flog2e = \$0D	? Flog10e = \$0E	ff = \$0F
? fL = \$0F	? Fzero = \$0F	? FPovfl = \$10	? fPdZA = \$10
? FPrzero = \$10	? fWb = \$10	V?]Fkfac = \$11	? FPinop = \$20
? FPunfla = \$20	? FPrninf = \$20	? fDb = \$20	? Fln2 = \$30
? FPrpinf = \$30	? Fln10 = \$31	? F10t0 = \$32	? F10t1 = \$33
? F10t2 = \$34	? F10t4 = \$35	? F10t8 = \$36	? F10t16 = \$37
? F10t32 = \$38	? F10t64 = \$39	? F10t128 = \$3A	? F10t256 = \$3B
? F10t512 = \$3C	? F10t1024 = \$3D	? F10t2048 = \$3E	? F10t4096 = \$3F
? FPsNaN = \$40	? FPovfla = \$40	? FPrpS = \$40	? fBb = \$40
? FPqsign = \$80	? FPinvA = \$80	? FPrpD = \$80	? fCb = \$80

fbias	=\$80	FAC	=\$9D	FACSGN	=\$A2	ARG	=\$A5
EXTRAFAC	=\$AC	MV llen	=\$EB	segptr	=\$FC	? fs	=\$040F
fx	=\$080F	? fP	=\$0C0F	? fw	=\$100F	? fd	=\$140F
? fb	=\$180F	patcher	=\$2000	? fpepatch	=\$2000	subseg	=\$203D
addseg	=\$2042	logseg	=\$2047	mulseg	=\$204C	divseg	=\$2051
sqrseg	=\$2056	expseg	=\$205F	cosseg	=\$2067	MV jlc	=\$206C
sinseg	=\$206C	segtbl	=\$2157	FP	=\$3000	Xbias	=\$3FFE
M? fOpod	=\$8000	MD segm	=\$8000	MD eseg	=\$8000	MD segtb	=\$8000
targadj	=\$A000	MD?fVECT	=\$8000	MD?fOPC	=\$8000	MD?fFPTR	=\$8000
MD?fRR	=\$8000	MD?fMR	=\$8000	MD?fOP	=\$8000	MD Fwtdata	=\$8000
MD Fwait	=\$8000	MD?fstcs	=\$8000	MD?From	=\$8000	MD?Freset	=\$8000
MD?fprint	=\$8000	MD?fmove	=\$8000	MD?Fldcs	=\$8000	MD?Fidx	=\$8000
MD?fcond	=\$8000	MD?Ftwotox	=\$8000	MD?Ftst	=\$8000	MD?Ftentox	=\$8000
MD?Ftanh	=\$8000	MD?Ftan	=\$8000	MD?Fsub	=\$8000	MD?Fsqr	=\$8000
MD?Fsinh	=\$8000	MD?Fsincos	=\$8000	MD?Fsin	=\$8000	MD?Fscale	=\$8000
MD?Frem	=\$8000	MD?Fneg	=\$8000	MD?Fmul	=\$8000	MD?Fmod	=\$8000
MD?Flog10	=\$8000	MD?Flog2	=\$8000	MD?Flnp1	=\$8000	MD?Fln	=\$8000
MD?Fint	=\$8000	MD?Fgetman	=\$8000	MD?Fgetexp	=\$8000	MD?Fetoxm1	=\$8000
MD?Fetox	=\$8000	MD?Fdiv	=\$8000	MD?Fcosh	=\$8000	MD?Fcos	=\$8000
MD?Fcmp	=\$8000	MD?Fatanh	=\$8000	MD?Fatan	=\$8000	MD?Fasin	=\$8000
MD?Fadd	=\$8000	MD?Facos	=\$8000	MD?Fabs	=\$8000	FPE	=\$C0C0
FPresp	=\$C0C0	? FPctrl	=\$C0C2	? FPsave	=\$C0C4	? FPrest	=\$C0C6
FPcmd	=\$C0C8	? FPcond	=\$C0CA	FPopnd	=\$C0CC	FPBASIC	=\$D000
IQERR	=\$E199	FSUBT	=\$E7AA	eFSUBT	=\$E7AF	FADDT	=\$E7C1
eFADDT	=\$E7C6	ZEROFAC	=\$E84E	OVERFLOW	=\$E8D5	LOG	=\$E94B
eLOG	=\$E950	FMU	=\$E987	eFMU	=\$E98C	FDIVT	=\$EA6B
eFDIVT	=\$EA70	SQRT	=\$EE8D	eSQRT	=\$EE96	EXP	=\$EF09
IQER	=\$EF0E	eEXP	=\$EF11	? RND	=\$EFAE	COS	=\$EFEA
eCOS	=\$EFFF	SIN	=\$EFF1	MV lstart	=\$EFF1	monadic	=\$EFF3
? FR02FAC	=\$EFFF	TAN	=\$F03A	? tan	=\$F03A	sktan	=\$F03F
unflow	=\$F060	ovflow	=\$F063	dyadic	=\$F066	ZPX2FR0	=\$F074
M done	=\$F088	ATAN	=\$F09E	? atn	=\$F09E	V jv	=\$F09F
skatn	=\$F0A3	M wait	=\$F0D2	eSIN	=\$F0DC		