```
                    1    ************************************************************
                    2    *                                                          *
                    3    *               N.BODY inner loop code                     *
                    4    *                                                          *
                    5    *                  Michael J. Mahon                        *
                    6    *                    Aug 25, 2010                          *
                    7    *                                                          *
                    8    ************************************************************
                    9
                   10    * FPE parameters
                   11
                   12    FPEslot  equ   4
                   13    fptr     equ   6          ; (6,7) FPE scratch pointer
                   14
                   20
                   21             org   $2000
                   22
                   23    * N.BODY variables
                   24
                   25    Nmax     equ   5          ; Maximum number of bodies - 1
                   26
2000: 00           27    N        db    0          ; Number of bodies
2001: 00           28    i        db    0          ; Body index variable
2002: 00           29    j        db    0          ; Body index Variable
2003: 00 00 00     30    dt       ds    fDl        ; Delta t
200B: 00 00 00     31    G        ds    fDl        ; Gravitational constant
2013: 00 00 00     32    BCD      ds    fPl        ; BCD for input-output
                   33
201F: 00 00 00     34    X        ds    Nmax*fDl   ; Body X array
2047: 00 00 00     35    Y        ds    Nmax*fDl   ; Body Y array
206F: 00 00 00     36    VX       ds    Nmax*fDl   ; Body X velocity array
2097: 00 00 00     37    VY       ds    Nmax*fDl   ; Body Y velocity array
20BF: 00 00 00     38    M        ds    Nmax*fDl   ; Body mass array
                   39
                   40    * Test I/O code
                   41
                   42             From  Fpi;FR1    ; FR1 = pi
                   42    ]FfRR    equ   1          ; Acts like a RegReg op
                   42             fOPC  $5C00;0;FR1;Fpi ; Load opcode
                   42    fOpcod   equ   FR1*$80+$5C00+0+Fpi
20E7: A2 5C        42             ldx   #>fOpcod
20E9: A0 80        42             ldy   #<fOpcod
                   42             eom
20EB: 20 C4 22     42             jsr   fRegReg
                   42             eom
                   43             Fmove FR1;fP;BCD ; BCD = pi
                   43             do    FR1/8      ; If MR op
                   43             fOP   0;]1;]2;]3 ;  do MR Fmove.
                   43             else
                   43             do    fP/8       ; If RM op
                   43    ]FfRM    equ   ]FfRM.fPb  ; Include fRegMem fP code
                   43             fFPTR BCD        ;  set fptr (if needed)
                   43             if    BCD=* ; Already set if addr="*".
                   43             else             ; If addr <> "*"
20EE: A9 13        43             lda   #<BCD      ;  set fptr = &addr
20F0: 85 06        43             sta   fptr
20F2: A9 20        43             lda   #>BCD
20F4: 85 07        43             sta   fptr+1
                   43             fin
                   43             eom
                   43             do    fP-fP
                   43             fOPC  $6000;]2-fF;]1;0 ; Kfac = 0 if not fP
                   43             else
                   43             fOPC  $6000;fP-fF;FR1;]fkfac ; Use Kfac if fP
                   43    fOpcod   equ   FR1*$80+$6000+fP-fF+]fkfac
20F6: A2 6C        43             ldx   #>fOpcod
20F8: A0 91        43             ldy   #<fOpcod
                   43             eom
                   43             fin
```

```
20FA: 20 73 23  43            jsr   fRegMem    ;  and do it.
                43            else              ; If RR op
                43            fRR   0;]1;]2     ;  do RR Fmove.
                43            fin
                43            fin
                43            eom
                44            Fmove fP;BCD;FR2 ; FR2 = pi
                44            do    fP/8        ; If MR op
                44            fOP   0;fP;BCD;FR2 ;  do MR Fmove.
                44            do    fP/8        ; If MR op
                44            fFPTR BCD         ;  set fptr (if needed)
                44            if    BCD=* ; Already set if addr="*".
                44            else              ; If addr <> "*"
20FD: A9 13     44            lda   #<BCD       ;  set fptr = &addr
20FF: 85 06     44            sta   fptr
2101: A9 20     44            lda   #>BCD
2103: 85 07     44            sta   fptr+1
                44            fin
                44            eom
                44            fMR   0;fP;FR2    ;  do MR &op;&fmt;&FRd
                44   ]FfMR    equ   ]FfMR.fPb   ; Include fMemReg fP code
                44            fOPC  $4000;fP-fF;FR2;0 ; Load opcode
                44   fOpcod   equ   FR2*$80+$4000+fP-fF+0
2105: A2 4D     44            ldx   #>fOpcod
2107: A0 00     44            ldy   #<fOpcod
                44            eom
2109: 20 06 23  44            jsr   fMemReg     ;  and do it.
                44            eom
                44            else              ; If RR op
                44            fRR   ]1;]2;]3    ;  do RR &op;&FRs;&FRd
                44            fin
                44            eom
                44            else
                44            do    ]2/8        ; If RM op
                44   ]FfRM    equ   ]FfRM.]2b   ; Include fRegMem ]2 code
                44            fFPTR ]3          ;  set fptr (if needed)
                44            do    ]2-fP
                44            fOPC  $6000;]2-fF;]1;0 ; Kfac = 0 if not fP
                44            else
                44            fOPC  $6000;]2-fF;]1;]Fkfac ; Use Kfac if fP
                44            fin
                44            jsr   fRegMem     ;  and do it.
                44            else              ; If RR op
                44            fRR   0;]1;]2     ;  do RR Fmove.
                44            fin
                44            fin
                44            eom
                45            Fsincos FR2;FR3;FR4 ; sin(FR2)->FR3; cos(FR2)->FR4
                45            do    FR2/8       ; If MR op
                45            fFPTR ]2          ;  set fptr (if needed)
                45            fMR   $30+]4;]1;]3 ;  do MR &op+&FRc;&fmt;&FRd
                45            else              ; If RR op
                45            fRR   $30+FR4;FR2;FR3 ;  do RR &op+&FRc;&FRs;&FRd
                45   ]FfRR    equ   1           ; Include fRegReg code.
                45   ]Freg    equ   FR2*$400    ; Shift FRs
                45            fOPC  0;]Freg;FR3;$30+FR4 ; Load opcode
                45   fOpcod   equ   FR3*$80+0+]Freg+$30+FR4
210C: A2 09     45            ldx   #>fOpcod
210E: A0 B4     45            ldy   #<fOpcod
                45            eom
2110: 20 C4 22  45            jsr   fRegReg     ;  and do it.
                45            eom
                45            fin
                45            eom
                46            Fsincos fD;X;FR3;FR4
                46            do    fD/8        ; If MR op
                46            fFPTR X           ;  set fptr (if needed)
```

```
                    46            if    X=* ; Already set if addr="*".
                    46            else              ; If addr <> "*"
2113: A9 1F         46            lda   #<X         ;  set fptr = &addr
2115: 85 06         46            sta   fptr
2117: A9 20         46            lda   #>X
2119: 85 07         46            sta   fptr+1
                    46            fin
                    46            eom
                    46            fMR   $30+FR4;fD;FR3 ;  do MR &op+&FRc;&fmt;&FRd
                    46   ]FfMR     equ   ]FfMR.fDb  ; Include fMemReg fD code
                    46            fOPC  $4000;fD-fF;FR3;$30+FR4 ; Load opcode
                    46   fOpcod   equ   FR3*$80+$4000+fD-fF+$30+FR4
211B: A2 55         46            ldx   #>fOpcod
211D: A0 B4         46            ldy   #<fOpcod
                    46            eom
211F: 20 06 23      46            jsr   fMemReg     ;  and do it.
                    46            eom
                    46            else              ; If RR op
                    46            fRR   $30+]3;]1;]2 ;  do RR &op+&FRc;&FRs;&FRd
                    46            fin
                    46            eom
                    47            Ftst  fD;X
                    47            do    fD/8        ; If MR op
                    47            fFPTR X           ;  set fptr (if needed)
                    47            if    X=* ; Already set if addr="*".
                    47            else              ; If addr <> "*"
2122: A9 1F         47            lda   #<X         ;  set fptr = &addr
2124: 85 06         47            sta   fptr
2126: A9 20         47            lda   #>X
2128: 85 07         47            sta   fptr+1
                    47            fin
                    47            eom
                    47            fMR   $3A;fD;0    ;  do MR &op;&fmt;0
                    47   ]FfMR     equ   ]FfMR.fDb  ; Include fMemReg fD code
                    47            fOPC  $4000;fD-fF;0;$3A ; Load opcode
                    47   fOpcod   equ   0*$80+$4000+fD-fF+$3A
212A: A2 54         47            ldx   #>fOpcod
212C: A0 3A         47            ldy   #<fOpcod
                    47            eom
212E: 20 06 23      47            jsr   fMemReg     ;  and do it.
                    47            eom
                    47            else              ; If RR op
                    47            fRR   $3A;]1;0    ;  do RR &op;&FRs;0
                    47            fin
                    47            eom
                    48            Ftst  FR1
                    48            do    FR1/8       ; If MR op
                    48            fFPTR ]2          ;  set fptr (if needed)
                    48            fMR   $3A;]1;0    ;  do MR &op;&fmt;0
                    48            else              ; If RR op
                    48            fRR   $3A;FR1;0   ;  do RR &op;&FRs;0
                    48   ]FfRR     equ   1           ; Include fRegReg code.
                    48   ]Freg     equ   FR1*$400    ; Shift FRs
                    48            fOPC  0;]Freg;0;$3A ; Load opcode
                    48   fOpcod   equ   0*$80+0+]Freg+$3A
2131: A2 04         48            ldx   #>fOpcod
2133: A0 3A         48            ldy   #<fOpcod
                    48            eom
2135: 20 C4 22      48            jsr   fRegReg     ;  and do it.
                    48            eom
                    48            fin
                    48            eom
                    49
                    50   * Inner loop code
                    51
                    52   loop     Fidx  fD;X;i      ; fptr -> X(i)
                    52   ]FfIDX   equ   ]FfIDX.fDb ; Include fIDXfD
```

```
2138: A2 1F     52            ldx   #<X
213A: A0 20     52            ldy   #>X
213C: AD 01 20  52            lda   i
213F: 20 AE 23  52            jsr   fIDXfD
                52            eom
                53            Fmove fD;*;FR0   ; FR0 = X(i)
                53            do    fD/8       ; If MR op
                53            fOP   0;fD;*;FR0 ;  do MR Fmove.
                53            do    fD/8       ; If MR op
                53            fFPTR *          ;  set fptr (if needed)
                53            if    *=* ; Already set if addr="*".
                53            else             ; If addr <> "*"
                53            lda   #<]1       ;  set fptr = &addr
                53            sta   fptr
                53            lda   #>]1
                53            sta   fptr+1
                53            fin
                53            eom
                53            fMR   0;fD;FR0   ;  do MR &op;&fmt;&FRd
                53   ]FfMR    equ   ]FfMR.fDb  ; Include fMemReg fD code
                53            fOPC  $4000;fD-fF;FR0;0 ; Load opcode
                53   fOpcod   equ   FR0*$80+$4000+fD-fF+0
2142: A2 54     53            ldx   #>fOpcod
2144: A0 00     53            ldy   #<fOpcod
                53            eom
2146: 20 06 23  53            jsr   fMemReg    ;  and do it.
                53            eom
                53            else             ; If RR op
                53            fRR   ]1;]2;]3   ;  do RR &op;&FRs;&FRd
                53            fin
                53            eom
                53            else
                53            do    ]2/8       ; If RM op
                53   ]FfRM    equ   ]FfRM.]2b  ; Include fRegMem ]2 code
                53            fFPTR ]3         ;  set fptr (if needed)
                53            do    ]2-fP
                53            fOPC  $6000;]2-fF;]1;0 ; Kfac = 0 if not fP
                53            else
                53            fOPC  $6000;]2-fF;]1;]Fkfac ; Use Kfac if fP
                53            fin
                53            jsr   fRegMem    ;  and do it.
                53            else             ; If RR op
                53            fRR   0;]1;]2    ;  do RR Fmove.
                53            fin
                53            fin
                53            eom
                54            Fidx  fD;X;j     ; fptr -> X(j)
                54   ]FfIDX   equ   ]FfIDX.fDb ; Include fIDXfD
2149: A2 1F     54            ldx   #<X
214B: A0 20     54            ldy   #>X
214D: AD 02 20  54            lda   j
2150: 20 AE 23  54            jsr   fIDXfD
                54            eom
                55            Fsub  fD;*;FR0   ; FR0 = X(i)-X(j) = dx
                55            fOP   $28;fD;*;FR0
                55            do    fD/8       ; If MR op
                55            fFPTR *          ;  set fptr (if needed)
                55            if    *=* ; Already set if addr="*".
                55            else             ; If addr <> "*"
                55            lda   #<]1       ;  set fptr = &addr
                55            sta   fptr
                55            lda   #>]1
                55            sta   fptr+1
                55            fin
                55            eom
                55            fMR   $28;fD;FR0 ;  do MR &op;&fmt;&FRd
                55   ]FfMR    equ   ]FfMR.fDb  ; Include fMemReg fD code
```

```
                  55            fOPC  $4000;fD-fF;FR0;$28 ; Load opcode
                  55   fOpcod   equ   FR0*$80+$4000+fD-fF+$28
2153: A2 54       55            ldx   #>fOpcod
2155: A0 28       55            ldy   #<fOpcod
                  55            eom
2157: 20 06 23    55            jsr   fMemReg    ;  and do it.
                  55            eom
                  55            else            ; If RR op
                  55            fRR   ]1;]2;]3   ;  do RR &op;&FRs;&FRd
                  55            fin
                  55            eom
                  55            eom
                  56
                  57            Fidx  fD;Y;i     ; fptr -> Y(i)
                  57   ]FfIDX   equ   ]FfIDX.fDb ; Include fIDXfD
215A: A2 47       57            ldx   #<Y
215C: A0 20       57            ldy   #>Y
215E: AD 01 20    57            lda   i
2161: 20 AE 23    57            jsr   fIDXfD
                  57            eom
                  58            Fmove fD;*;FR1   ; FR1 = Y(i)
                  58            do    fD/8       ; If MR op
                  58            fOP   0;fD;*;FR1 ;  do MR Fmove.
                  58            do    fD/8       ; If MR op
                  58            fFPTR *          ;  set fptr (if needed)
                  58            if    *=* ; Already set if addr="*".
                  58            else            ; If addr <> "*"
                  58            lda   #<]1       ;  set fptr = &addr
                  58            sta   fptr
                  58            lda   #>]1
                  58            sta   fptr+1
                  58            fin
                  58            eom
                  58            fMR   0;fD;FR1   ;  do MR &op;&fmt;&FRd
                  58   ]FfMR    equ   ]FfMR.fDb  ; Include fMemReg fD code
                  58            fOPC  $4000;fD-fF;FR1;0 ; Load opcode
                  58   fOpcod   equ   FR1*$80+$4000+fD-fF+0
2164: A2 54       58            ldx   #>fOpcod
2166: A0 80       58            ldy   #<fOpcod
                  58            eom
2168: 20 06 23    58            jsr   fMemReg    ;  and do it.
                  58            eom
                  58            else            ; If RR op
                  58            fRR   ]1;]2;]3   ;  do RR &op;&FRs;&FRd
                  58            fin
                  58            eom
                  58            else
                  58            do    ]2/8       ; If RM op
                  58   ]FfRM    equ   ]FfRM.]2b  ; Include fRegMem ]2 code
                  58            fFPTR ]3         ;  set fptr (if needed)
                  58            do    ]2-fP
                  58            fOPC  $6000;]2-fF;]1;0 ; Kfac = 0 if not fP
                  58            else
                  58            fOPC  $6000;]2-fF;]1;]Fkfac ; Use Kfac if fP
                  58            fin
                  58            jsr   fRegMem    ;  and do it.
                  58            else            ; If RR op
                  58            fRR   0;]1;]2    ;  do RR Fmove.
                  58            fin
                  58            fin
                  58            eom
                  59            Fidx  fD;Y;j     ; fptr -> Y(j)
                  59   ]FfIDX   equ   ]FfIDX.fDb ; Include fIDXfD
216B: A2 47       59            ldx   #<Y
216D: A0 20       59            ldy   #>Y
216F: AD 02 20    59            lda   j
2172: 20 AE 23    59            jsr   fIDXfD
```

```
                    59            eom
                    60            Fsub  fD;*;FR1   ; FR1 = Y(i)-Y(j) = dy
                    60            fOP   $28;fD;*;FR1
                    60            do    fD/8       ; If MR op
                    60            fFPTR *          ;  set fptr (if needed)
                    60            if    *=* ; Already set if addr="*".
                    60            else             ; If addr <> "*"
                    60            lda   #<]1       ;  set fptr = &addr
                    60            sta   fptr
                    60            lda   #>]1
                    60            sta   fptr+1
                    60            fin
                    60            eom
                    60            fMR   $28;fD;FR1 ;  do MR &op;&fmt;&FRd
                    60   ]FfMR    equ   ]FfMR.fDb  ; Include fMemReg fD code
                    60            fOPC  $4000;fD-fF;FR1;$28 ; Load opcode
                    60   fOpcod   equ   FR1*$80+$4000+fD-fF+$28
2175: A2 54         60            ldx   #>fOpcod
2177: A0 A8         60            ldy   #<fOpcod
                    60            eom
2179: 20 06 23      60            jsr   fMemReg    ;  and do it.
                    60            eom
                    60            else             ; If RR op
                    60            fRR   ]1;]2;]3   ;  do RR &op;&FRs;&FRd
                    60            fin
                    60            eom
                    60            eom
                    61
                    62            Fmove FR0;FR7;   ; FR7 = dx
                    62            do    FR0/8      ; If MR op
                    62            fOP   0;]1;]2;]3 ;  do MR Fmove.
                    62            else
                    62            do    FR7/8      ; If RM op
                    62   ]FfRM    equ   ]FfRM.]2b  ; Include fRegMem ]2 code
                    62            fFPTR ]3         ;  set fptr (if needed)
                    62            do    ]2-fP
                    62            fOPC  $6000;]2-fF;]1;0 ; Kfac = 0 if not fP
                    62            else
                    62            fOPC  $6000;]2-fF;]1;]Fkfac ; Use Kfac if fP
                    62            fin
                    62            jsr   fRegMem    ;  and do it.
                    62            else             ; If RR op
                    62            fRR   0;FR0;FR7  ;  do RR Fmove.
                    62   ]FfRR    equ   1          ; Include fRegReg code.
                    62   ]Freg    equ   FR0*$400   ; Shift FRs
                    62            fOPC  0;]Freg;FR7;0 ; Load opcode
                    62   fOpcod   equ   FR7*$80+0+]Freg+0
217C: A2 03         62            ldx   #>fOpcod
217E: A0 80         62            ldy   #<fOpcod
                    62            eom
2180: 20 C4 22      62            jsr   fRegReg    ;  and do it.
                    62            eom
                    62            fin
                    62            fin
                    62            eom
                    63            Fmul  FR7;FR7;   ; FR7 = dx^2
                    63            fOP   $23;FR7;FR7;
                    63            do    FR7/8      ; If MR op
                    63            fFPTR ]3         ;  set fptr (if needed)
                    63            fMR   ]1;]2;]4   ;  do MR &op;&fmt;&FRd
                    63            else             ; If RR op
                    63            fRR   $23;FR7;FR7 ;  do RR &op;&FRs;&FRd
                    63   ]FfRR    equ   1          ; Include fRegReg code.
                    63   ]Freg    equ   FR7*$400   ; Shift FRs
                    63            fOPC  0;]Freg;FR7;$23 ; Load opcode
                    63   fOpcod   equ   FR7*$80+0+]Freg+$23
2183: A2 1F         63            ldx   #>fOpcod
```

```
2185: A0 A3     63            ldy   #<fOpcod
                63            eom
2187: 20 C4 22  63            jsr   fRegReg    ;  and do it.
                63            eom
                63            fin
                63            eom
                63            eom
                64            Fmove FR1;FR6;   ; FR6 = dy
                64            do    FR1/8      ; If MR op
                64            fOP   0;]1;]2;]3 ;  do MR Fmove.
                64            else
                64            do    FR6/8      ; If RM op
                64   ]FfRM    equ   ]FfRM.]2b  ; Include fRegMem ]2 code
                64            fFPTR ]3         ;  set fptr (if needed)
                64            do    ]2-fP
                64            fOPC  $6000;]2-fF;]1;0 ; Kfac = 0 if not fP
                64            else
                64            fOPC  $6000;]2-fF;]1;]Fkfac ; Use Kfac if fP
                64            fin
                64            jsr   fRegMem    ;  and do it.
                64            else             ; If RR op
                64            fRR   0;FR1;FR6  ;  do RR Fmove.
                64   ]FfRR    equ   1          ; Include fRegReg code.
                64   ]Freg    equ   FR1*$400   ; Shift FRs
                64            fOPC  0;]Freg;FR6;0 ; Load opcode
                64   fOpcod   equ   FR6*$80+0+]Freg+0
218A: A2 07     64            ldx   #>fOpcod
218C: A0 00     64            ldy   #<fOpcod
                64            eom
218E: 20 C4 22  64            jsr   fRegReg    ;  and do it.
                64            eom
                64            fin
                64            fin
                64            eom
                65            Fmul  FR6;FR6;   ; FR6 = dy^2
                65            fOP   $23;FR6;FR6;
                65            do    FR6/8      ; If MR op
                65            fFPTR ]3         ;  set fptr (if needed)
                65            fMR   ]1;]2;]4   ;  do MR &op;&fmt;&FRd
                65            else             ; If RR op
                65            fRR   $23;FR6;FR6 ;  do RR &op;&FRs;&FRd
                65   ]FfRR    equ   1          ; Include fRegReg code.
                65   ]Freg    equ   FR6*$400   ; Shift FRs
                65            fOPC  0;]Freg;FR6;$23 ; Load opcode
                65   fOpcod   equ   FR6*$80+0+]Freg+$23
2191: A2 1B     65            ldx   #>fOpcod
2193: A0 23     65            ldy   #<fOpcod
                65            eom
2195: 20 C4 22  65            jsr   fRegReg    ;  and do it.
                65            eom
                65            fin
                65            eom
                65            eom
                66            Fadd  FR6;FR7;   ; FR7 = dx^2+dy^2 = r^2
                66            fOP   $22;FR6;FR7;
                66            do    FR6/8      ; If MR op
                66            fFPTR ]3         ;  set fptr (if needed)
                66            fMR   ]1;]2;]4   ;  do MR &op;&fmt;&FRd
                66            else             ; If RR op
                66            fRR   $22;FR6;FR7 ;  do RR &op;&FRs;&FRd
                66   ]FfRR    equ   1          ; Include fRegReg code.
                66   ]Freg    equ   FR6*$400   ; Shift FRs
                66            fOPC  0;]Freg;FR7;$22 ; Load opcode
                66   fOpcod   equ   FR7*$80+0+]Freg+$22
2198: A2 1B     66            ldx   #>fOpcod
219A: A0 A2     66            ldy   #<fOpcod
                66            eom
```

```
219C: 20 C4 22  66            jsr   fRegReg    ;  and do it.
                66            eom
                66            fin
                66            eom
                66            eom
                67
                68            Fmove FR7;FR6;   ; FR6 = r^2
                68            do    FR7/8      ; If MR op
                68            fOP   0;]1;]2;]3 ;  do MR Fmove.
                68            else
                68            do    FR6/8      ; If RM op
                68   ]FfRM    equ   ]FfRM.]2b  ; Include fRegMem ]2 code
                68            fFPTR ]3         ;  set fptr (if needed)
                68            do    ]2-fP
                68            fOPC  $6000;]2-fF;]1;0 ; Kfac = 0 if not fP
                68            else
                68            fOPC  $6000;]2-fF;]1;]Fkfac ; Use Kfac if fP
                68            fin
                68            jsr   fRegMem    ;  and do it.
                68            else             ; If RR op
                68            fRR   0;FR7;FR6  ;  do RR Fmove.
                68   ]FfRR    equ   1          ; Include fRegReg code.
                68   ]Freg    equ   FR7*$400   ; Shift FRs
                68            fOPC  0;]Freg;FR6;0 ; Load opcode
                68   fOpcod   equ   FR6*$80+0+]Freg+0
219F: A2 1F     68            ldx   #>fOpcod
21A1: A0 00     68            ldy   #<fOpcod
                68            eom
21A3: 20 C4 22  68            jsr   fRegReg    ;  and do it.
                68            eom
                68            fin
                68            fin
                68            eom
                69            Fmul  FR6;FR6;   ; FR6 = r^4
                69            fOP   $23;FR6;FR6;
                69            do    FR6/8      ; If MR op
                69            fFPTR ]3         ;  set fptr (if needed)
                69            fMR   ]1;]2;]4   ;  do MR &op;&fmt;&FRd
                69            else             ; If RR op
                69            fRR   $23;FR6;FR6 ;  do RR &op;&FRs;&FRd
                69   ]FfRR    equ   1          ; Include fRegReg code.
                69   ]Freg    equ   FR6*$400   ; Shift FRs
                69            fOPC  0;]Freg;FR6;$23 ; Load opcode
                69   fOpcod   equ   FR6*$80+0+]Freg+$23
21A6: A2 1B     69            ldx   #>fOpcod
21A8: A0 23     69            ldy   #<fOpcod
                69            eom
21AA: 20 C4 22  69            jsr   fRegReg    ;  and do it.
                69            eom
                69            fin
                69            eom
                69            eom
                70            Fmul  FR6;FR7;   ; FR7 = r^6
                70            fOP   $23;FR6;FR7;
                70            do    FR6/8      ; If MR op
                70            fFPTR ]3         ;  set fptr (if needed)
                70            fMR   ]1;]2;]4   ;  do MR &op;&fmt;&FRd
                70            else             ; If RR op
                70            fRR   $23;FR6;FR7 ;  do RR &op;&FRs;&FRd
                70   ]FfRR    equ   1          ; Include fRegReg code.
                70   ]Freg    equ   FR6*$400   ; Shift FRs
                70            fOPC  0;]Freg;FR7;$23 ; Load opcode
                70   fOpcod   equ   FR7*$80+0+]Freg+$23
21AD: A2 1B     70            ldx   #>fOpcod
21AF: A0 A3     70            ldy   #<fOpcod
                70            eom
21B1: 20 C4 22  70            jsr   fRegReg    ;  and do it.
```

```
                    70            eom
                    70            fin
                    70            eom
                    70            eom
                    71            Fsqrt FR7;FR7;   ; FR7 = SQRT(r^6)
                    71            fOP   $04;FR7;FR7;
                    71            do    FR7/8      ; If MR op
                    71            fFPTR ]3         ;  set fptr (if needed)
                    71            fMR   ]1;]2;]4   ;  do MR &op;&fmt;&FRd
                    71            else             ; If RR op
                    71            fRR   $04;FR7;FR7 ;  do RR &op;&FRs;&FRd
                    71   ]FfRR    equ   1          ; Include fRegReg code.
                    71   ]Freg    equ   FR7*$400   ; Shift FRs
                    71            fOPC  0;]Freg;FR7;$04 ; Load opcode
                    71   fOpcod   equ   FR7*$80+0+]Freg+$04
21B4: A2 1F         71            ldx   #>fOpcod
21B6: A0 84         71            ldy   #<fOpcod
                    71            eom
21B8: 20 C4 22      71            jsr   fRegReg    ;  and do it.
                    71            eom
                    71            fin
                    71            eom
                    71            eom
                    72
                    73            Fmove fD;dt;FR2  ; FR2 = dt
                    73            do    fD/8       ; If MR op
                    73            fOP   0;fD;dt;FR2 ;  do MR Fmove.
                    73            do    fD/8       ; If MR op
                    73            fFPTR dt         ;  set fptr (if needed)
                    73            if    dt=* ; Already set if addr="*".
                    73            else             ; If addr <> "*"
21BB: A9 03         73            lda   #<dt       ;  set fptr = &addr
21BD: 85 06         73            sta   fptr
21BF: A9 20         73            lda   #>dt
21C1: 85 07         73            sta   fptr+1
                    73            fin
                    73            eom
                    73            fMR   0;fD;FR2   ;  do MR &op;&fmt;&FRd
                    73   ]FfMR    equ   ]FfMR.fDb  ; Include fMemReg fD code
                    73            fOPC  $4000;fD-fF;FR2;0 ; Load opcode
                    73   fOpcod   equ   FR2*$80+$4000+fD-fF+0
21C3: A2 55         73            ldx   #>fOpcod
21C5: A0 00         73            ldy   #<fOpcod
                    73            eom
21C7: 20 06 23      73            jsr   fMemReg    ;  and do it.
                    73            eom
                    73            else             ; If RR op
                    73            fRR   ]1;]2;]3   ;  do RR &op;&FRs;&FRd
                    73            fin
                    73            eom
                    73            else
                    73            do    ]2/8       ; If RM op
                    73   ]FfRM    equ   ]FfRM.]2b  ; Include fRegMem ]2 code
                    73            fFPTR ]3         ;  set fptr (if needed)
                    73            do    ]2-fP
                    73            fOPC  $6000;]2-fF;]1;0 ; Kfac = 0 if not fP
                    73            else
                    73            fOPC  $6000;]2-fF;]1;]Fkfac ; Use Kfac if fP
                    73            fin
                    73            jsr   fRegMem    ;  and do it.
                    73            else             ; If RR op
                    73            fRR   0;]1;]2    ;  do RR Fmove.
                    73            fin
                    73            fin
                    73            eom
                    74            Fmul  fD;G;FR2   ; FR2 = dt*G
                    74            fOP   $23;fD;G;FR2
```

```
                    74            do    fD/8       ; If MR op
                    74            fFPTR G          ;  set fptr (if needed)
                    74            if    G=* ; Already set if addr="*".
                    74            else             ; If addr <> "*"
21CA: A9 0B         74            lda   #<G        ;  set fptr = &addr
21CC: 85 06         74            sta   fptr
21CE: A9 20         74            lda   #>G
21D0: 85 07         74            sta   fptr+1
                    74            fin
                    74            eom
                    74            fMR   $23;fD;FR2 ;  do MR &op;&fmt;&FRd
                    74   ]FfMR     equ   ]FfMR.fDb  ; Include fMemReg fD code
                    74            fOPC  $4000;fD-fF;FR2;$23 ; Load opcode
                    74   fOpcod   equ   FR2*$80+$4000+fD-fF+$23
21D2: A2 55         74            ldx   #>fOpcod
21D4: A0 23         74            ldy   #<fOpcod
                    74            eom
21D6: 20 06 23      74            jsr   fMemReg    ;  and do it.
                    74            eom
                    74            else             ; If RR op
                    74            fRR   ]1;]2;]3   ;  do RR &op;&FRs;&FRd
                    74            fin
                    74            eom
                    74            eom
                    75            Fidx  fD;M;i     ; fptr -> M(i)
                    75   ]FfIDX   equ   ]FfIDX.fDb ; Include fIDXfD
21D9: A2 BF         75            ldx   #<M
21DB: A0 20         75            ldy   #>M
21DD: AD 01 20      75            lda   i
21E0: 20 AE 23      75            jsr   fIDXfD
                    75            eom
                    76            Fmove fD;*;FR3   ; FR3 = M(i)
                    76            do    fD/8       ; If MR op
                    76            fOP   0;fD;*;FR3 ;  do MR Fmove.
                    76            do    fD/8       ; If MR op
                    76            fFPTR *          ;  set fptr (if needed)
                    76            if    *=* ; Already set if addr="*".
                    76            else             ; If addr <> "*"
                    76            lda   #<]1       ;  set fptr = &addr
                    76            sta   fptr
                    76            lda   #>]1
                    76            sta   fptr+1
                    76            fin
                    76            eom
                    76            fMR   0;fD;FR3   ;  do MR &op;&fmt;&FRd
                    76   ]FfMR     equ   ]FfMR.fDb  ; Include fMemReg fD code
                    76            fOPC  $4000;fD-fF;FR3;0 ; Load opcode
                    76   fOpcod   equ   FR3*$80+$4000+fD-fF+0
21E3: A2 55         76            ldx   #>fOpcod
21E5: A0 80         76            ldy   #<fOpcod
                    76            eom
21E7: 20 06 23      76            jsr   fMemReg    ;  and do it.
                    76            eom
                    76            else             ; If RR op
                    76            fRR   ]1;]2;]3   ;  do RR &op;&FRs;&FRd
                    76            fin
                    76            eom
                    76            else
                    76            do    ]2/8       ; If RM op
                    76   ]FfRM     equ   ]FfRM.]2b  ; Include fRegMem ]2 code
                    76            fFPTR ]3         ;  set fptr (if needed)
                    76            do    ]2-fP
                    76            fOPC  $6000;]2-fF;]1;0 ; Kfac = 0 if not fP
                    76            else
                    76            fOPC  $6000;]2-fF;]1;]fkfac ; Use Kfac if fP
                    76            fin
                    76            jsr   fRegMem    ;  and do it.
```

```
                  76            else              ; If RR op
                  76            fRR   0;]1;]2     ;  do RR Fmove.
                  76            fin
                  76            fin
                  76            eom
                  77            Fmul  FR3;FR2;    ; FR2 = dt*G*M(i)
                  77            fOP   $23;FR3;FR2;
                  77            do    FR3/8       ; If MR op
                  77            fFPTR ]3          ;  set fptr (if needed)
                  77            fMR   ]1;]2;]4    ;  do MR &op;&fmt;&FRd
                  77            else              ; If RR op
                  77            fRR   $23;FR3;FR2 ;  do RR &op;&FRs;&FRd
                  77   ]FfRR    equ   1           ; Include fRegReg code.
                  77   ]Freg    equ   FR3*$400    ; Shift FRs
                  77            fOPC  0;]Freg;FR2;$23 ; Load opcode
                  77   fOpcod   equ   FR2*$80+0+]Freg+$23
21EA: A2 0D       77            ldx   #>fOpcod
21EC: A0 23       77            ldy   #<fOpcod
                  77            eom
21EE: 20 C4 22    77            jsr   fRegReg     ;  and do it.
                  77            eom
                  77            fin
                  77            eom
                  77            eom
                  78            Fidx  fD;M;j      ; fptr -> M(j)
                  78   ]FfIDX   equ   ]FfIDX.fDb ; Include fIDXfD
21F1: A2 BF       78            ldx   #<M
21F3: A0 20       78            ldy   #>M
21F5: AD 02 20    78            lda   j
21F8: 20 AE 23    78            jsr   fIDXfD
                  78            eom
                  79            Fmove fD;*;FR4    ; FR4 = M(j)
                  79            do    fD/8        ; If MR op
                  79            fOP   0;fD;*;FR4  ;  do MR Fmove.
                  79            do    fD/8        ; If MR op
                  79            fFPTR *           ;  set fptr (if needed)
                  79            if    *=* ; Already set if addr="*".
                  79            else              ; If addr <> "*"
                  79            lda   #<]1        ;  set fptr = &addr
                  79            sta   fptr
                  79            lda   #>]1
                  79            sta   fptr+1
                  79            fin
                  79            eom
                  79            fMR   0;fD;FR4    ;  do MR &op;&fmt;&FRd
                  79   ]FfMR    equ   ]FfMR.fDb   ; Include fMemReg fD code
                  79            fOPC  $4000;fD-fF;FR4;0 ; Load opcode
                  79   fOpcod   equ   FR4*$80+$4000+fD-fF+0
21FB: A2 56       79            ldx   #>fOpcod
21FD: A0 00       79            ldy   #<fOpcod
                  79            eom
21FF: 20 06 23    79            jsr   fMemReg     ;  and do it.
                  79            eom
                  79            else              ; If RR op
                  79            fRR   ]1;]2;]3    ;  do RR &op;&FRs;&FRd
                  79            fin
                  79            eom
                  79            else
                  79            do    ]2/8        ; If RM op
                  79   ]FfRM    equ   ]FfRM.]2b   ; Include fRegMem ]2 code
                  79            fFPTR ]3          ;  set fptr (if needed)
                  79            do    ]2-fP
                  79            fOPC  $6000;]2-fF;]1;0 ; Kfac = 0 if not fP
                  79            else
                  79            fOPC  $6000;]2-fF;]1;]fkfac ; Use Kfac if fP
                  79            fin
                  79            jsr   fRegMem     ;  and do it.
```

```
                    79            else              ; If RR op
                    79            fRR   0;]1;]2     ;  do RR Fmove.
                    79            fin
                    79            fin
                    79            eom
                    80            Fmul  FR4;FR2;    ; FR2 = dt*G*M(i)*M(j)
                    80            fOP   $23;FR4;FR2;
                    80            do    FR4/8       ; If MR op
                    80            fFPTR ]3          ;  set fptr (if needed)
                    80            fMR   ]1;]2;]4    ;  do MR &op;&fmt;&FRd
                    80            else              ; If RR op
                    80            fRR   $23;FR4;FR2 ;  do RR &op;&FRs;&FRd
                    80   ]FfRR    equ   1           ; Include fRegReg code.
                    80   ]Freg    equ   FR4*$400    ; Shift FRs
                    80            fOPC  0;]Freg;FR2;$23 ; Load opcode
                    80   fOpcod   equ   FR2*$80+0+]Freg+$23
2202: A2 11         80            ldx   #>fOpcod
2204: A0 23         80            ldy   #<fOpcod
                    80            eom
2206: 20 C4 22      80            jsr   fRegReg     ;  and do it.
                    80            eom
                    80            fin
                    80            eom
                    80            eom
                    81            Fdiv  FR7;FR2;    ; FR2 = dt*G*M(i)*M(j)/sqrt(r^6) =
force
                    81            fOP   $20;FR7;FR2;
                    81            do    FR7/8       ; If MR op
                    81            fFPTR ]3          ;  set fptr (if needed)
                    81            fMR   ]1;]2;]4    ;  do MR &op;&fmt;&FRd
                    81            else              ; If RR op
                    81            fRR   $20;FR7;FR2 ;  do RR &op;&FRs;&FRd
                    81   ]FfRR    equ   1           ; Include fRegReg code.
                    81   ]Freg    equ   FR7*$400    ; Shift FRs
                    81            fOPC  0;]Freg;FR2;$20 ; Load opcode
                    81   fOpcod   equ   FR2*$80+0+]Freg+$20
2209: A2 1D         81            ldx   #>fOpcod
220B: A0 20         81            ldy   #<fOpcod
                    81            eom
220D: 20 C4 22      81            jsr   fRegReg     ;  and do it.
                    81            eom
                    81            fin
                    81            eom
                    81            eom
                    82
                    83            Fmove FR2;FR5;    ; FR5 = force
                    83            do    FR2/8       ; If MR op
                    83            fOP   0;]1;]2;]3  ;  do MR Fmove.
                    83            else
                    83            do    FR5/8       ; If RM op
                    83   ]FfRM    equ   ]FfRM.]2b   ; Include fRegMem ]2 code
                    83            fFPTR ]3          ;  set fptr (if needed)
                    83            do    ]2-fP
                    83            fOPC  $6000;]2-fF;]1;0 ; Kfac = 0 if not fP
                    83            else
                    83            fOPC  $6000;]2-fF;]1;]Fkfac ; Use Kfac if fP
                    83            fin
                    83            jsr   fRegMem     ;  and do it.
                    83            else              ; If RR op
                    83            fRR   0;FR2;FR5   ;  do RR Fmove.
                    83   ]FfRR    equ   1           ; Include fRegReg code.
                    83   ]Freg    equ   FR2*$400    ; Shift FRs
                    83            fOPC  0;]Freg;FR5;0 ; Load opcode
                    83   fOpcod   equ   FR5*$80+0+]Freg+0
2210: A2 0A         83            ldx   #>fOpcod
2212: A0 80         83            ldy   #<fOpcod
                    83            eom
```

```
2214: 20 C4 22  83            jsr   fRegReg    ;  and do it.
                83            eom
                83            fin
                83            fin
                83            eom
                84            Fdiv  FR3;FR5;   ; FR5 = force/M(i) = impi
                84            fOP   $20;FR3;FR5;
                84            do    FR3/8      ; If MR op
                84            fFPTR ]3         ;  set fptr (if needed)
                84            fMR   ]1;]2;]4   ;  do MR &op;&fmt;&FRd
                84            else             ; If RR op
                84            fRR   $20;FR3;FR5 ;  do RR &op;&FRs;&FRd
                84   ]FfRR    equ   1          ; Include fRegReg code.
                84   ]Freg    equ   FR3*$400   ; Shift FRs
                84            fOPC  0;]Freg;FR5;$20 ; Load opcode
                84   fOpcod   equ   FR5*$80+0+]Freg+$20
2217: A2 0E     84            ldx   #>fOpcod
2219: A0 A0     84            ldy   #<fOpcod
                84            eom
221B: 20 C4 22  84            jsr   fRegReg    ;  and do it.
                84            eom
                84            fin
                84            eom
                84            eom
                85
                86            Fmove FR2;FR6;   ; FR6 = force
                86            do    FR2/8      ; If MR op
                86            fOP   0;]1;]2;]3 ;  do MR Fmove.
                86            else
                86            do    FR6/8      ; If RM op
                86   ]FfRM    equ   ]FfRM.]2b  ; Include fRegMem ]2 code
                86            fFPTR ]3         ;  set fptr (if needed)
                86            do    ]2-fP
                86            fOPC  $6000;]2-fF;]1;0 ; Kfac = 0 if not fP
                86            else
                86            fOPC  $6000;]2-fF;]1;]Fkfac ; Use Kfac if fP
                86            fin
                86            jsr   fRegMem    ;  and do it.
                86            else             ; If RR op
                86            fRR   0;FR2;FR6  ;  do RR Fmove.
                86   ]FfRR    equ   1          ; Include fRegReg code.
                86   ]Freg    equ   FR2*$400   ; Shift FRs
                86            fOPC  0;]Freg;FR6;0 ; Load opcode
                86   fOpcod   equ   FR6*$80+0+]Freg+0
221E: A2 0B     86            ldx   #>fOpcod
2220: A0 00     86            ldy   #<fOpcod
                86            eom
2222: 20 C4 22  86            jsr   fRegReg    ;  and do it.
                86            eom
                86            fin
                86            fin
                86            eom
                87            Fdiv  FR4;FR6;   ; FR6 = force/M(j) = impj
                87            fOP   $20;FR4;FR6;
                87            do    FR4/8      ; If MR op
                87            fFPTR ]3         ;  set fptr (if needed)
                87            fMR   ]1;]2;]4   ;  do MR &op;&fmt;&FRd
                87            else             ; If RR op
                87            fRR   $20;FR4;FR6 ;  do RR &op;&FRs;&FRd
                87   ]FfRR    equ   1          ; Include fRegReg code.
                87   ]Freg    equ   FR4*$400   ; Shift FRs
                87            fOPC  0;]Freg;FR6;$20 ; Load opcode
                87   fOpcod   equ   FR6*$80+0+]Freg+$20
2225: A2 13     87            ldx   #>fOpcod
2227: A0 20     87            ldy   #<fOpcod
                87            eom
2229: 20 C4 22  87            jsr   fRegReg    ;  and do it.
```

```
                    87            eom
                    87            fin
                    87            eom
                    87            eom
                    88
                    89            Fneg  FR5;FR3;   ; FR3 = -impi
                    89            fOP   $1A;FR5;FR3;
                    89            do    FR5/8      ; If MR op
                    89            fFPTR ]3         ;  set fptr (if needed)
                    89            fMR   ]1;]2;]4   ;  do MR &op;&fmt;&FRd
                    89            else             ; If RR op
                    89            fRR   $1A;FR5;FR3 ;  do RR &op;&FRs;&FRd
                    89   ]FfRR    equ   1          ; Include fRegReg code.
                    89   ]Freg    equ   FR5*$400   ; Shift FRs
                    89            fOPC  0;]Freg;FR3;$1A ; Load opcode
                    89   fOpcod   equ   FR3*$80+0+]Freg+$1A
222C: A2 15         89            ldx   #>fOpcod
222E: A0 9A         89            ldy   #<fOpcod
                    89            eom
2230: 20 C4 22      89            jsr   fRegReg    ;  and do it.
                    89            eom
                    89            fin
                    89            eom
                    89            eom
                    90            Fmul  FR0;FR3;   ; FR3 = -impi*dx
                    90            fOP   $23;FR0;FR3;
                    90            do    FR0/8      ; If MR op
                    90            fFPTR ]3         ;  set fptr (if needed)
                    90            fMR   ]1;]2;]4   ;  do MR &op;&fmt;&FRd
                    90            else             ; If RR op
                    90            fRR   $23;FR0;FR3 ;  do RR &op;&FRs;&FRd
                    90   ]FfRR    equ   1          ; Include fRegReg code.
                    90   ]Freg    equ   FR0*$400   ; Shift FRs
                    90            fOPC  0;]Freg;FR3;$23 ; Load opcode
                    90   fOpcod   equ   FR3*$80+0+]Freg+$23
2233: A2 01         90            ldx   #>fOpcod
2235: A0 A3         90            ldy   #<fOpcod
                    90            eom
2237: 20 C4 22      90            jsr   fRegReg    ;  and do it.
                    90            eom
                    90            fin
                    90            eom
                    90            eom
                    91            Fidx  fD;VX;i    ; fptr -> VX(i)
                    91   ]FfIDX   equ   ]FfIDX.fDb ; Include fIDXfD
223A: A2 6F         91            ldx   #<VX
223C: A0 20         91            ldy   #>VX
223E: AD 01 20      91            lda   i
2241: 20 AE 23      91            jsr   fIDXfD
                    91            eom
                    92            Fadd  fD;*;FR3   ; FR3 = VX(i)
                    92            fOP   $22;fD;*;FR3
                    92            do    fD/8       ; If MR op
                    92            fFPTR *          ;  set fptr (if needed)
                    92            if    *=* ; Already set if addr="*".
                    92            else             ; If addr <> "*"
                    92            lda   #<]1       ;  set fptr = &addr
                    92            sta   fptr
                    92            lda   #>]1
                    92            sta   fptr+1
                    92            fin
                    92            eom
                    92            fMR   $22;fD;FR3 ;  do MR &op;&fmt;&FRd
                    92   ]FfMR    equ   ]FfMR.fDb  ; Include fMemReg fD code
                    92            fOPC  $4000;fD-fF;FR3;$22 ; Load opcode
                    92   fOpcod   equ   FR3*$80+$4000+fD-fF+$22
2244: A2 55         92            ldx   #>fOpcod
```

```
2246: A0 A2     92            ldy   #<fOpcod
                92            eom
2248: 20 06 23  92            jsr   fMemReg    ;  and do it.
                92            eom
                92            else            ; If RR op
                92            fRR   ]1;]2;]3   ;  do RR &op;&FRs;&FRd
                92            fin
                92            eom
                92            eom
                93            Fmove FR3;fD;*   ; VX(i) = VX(i)-impi*dx
                93            do    FR3/8      ; If MR op
                93            fOP   0;]1;]2;]3 ;  do MR Fmove.
                93            else
                93            do    fD/8       ; If RM op
                93   ]FfRM    equ   ]FfRM.fDb  ; Include fRegMem fD code
                93            fFPTR *          ;  set fptr (if needed)
                93            if    *=* ; Already set if addr="*".
                93            else            ; If addr <> "*"
                93            lda   #<]1       ;  set fptr = &addr
                93            sta   fptr
                93            lda   #>]1
                93            sta   fptr+1
                93            fin
                93            eom
                93            do    fD-fP
                93            fOPC  $6000;fD-fF;FR3;0 ; Kfac = 0 if not fP
                93   fOpcod   equ   FR3*$80+$6000+fD-fF+0
224B: A2 75     93            ldx   #>fOpcod
224D: A0 80     93            ldy   #<fOpcod
                93            eom
                93            else
                93            fOPC  $6000;]2-fF;]1;]Fkfac ; Use Kfac if fP
                93            fin
224F: 20 73 23  93            jsr   fRegMem    ;  and do it.
                93            else            ; If RR op
                93            fRR   0;]1;]2    ;  do RR Fmove.
                93            fin
                93            fin
                93            eom
                94
                95            Fneg  FR6;FR3;   ; FR3 = -impj
                95            fOP   $1A;FR6;FR3;
                95            do    FR6/8      ; If MR op
                95            fFPTR ]3         ;  set fptr (if needed)
                95            fMR   ]1;]2;]4   ;  do MR &op;&fmt;&FRd
                95            else            ; If RR op
                95            fRR   $1A;FR6;FR3 ;  do RR &op;&FRs;&FRd
                95   ]FfRR    equ   1          ; Include fRegReg code.
                95   ]Freg    equ   FR6*$400   ; Shift FRs
                95            fOPC  0;]Freg;FR3;$1A ; Load opcode
                95   fOpcod   equ   FR3*$80+0+]Freg+$1A
2252: A2 19     95            ldx   #>fOpcod
2254: A0 9A     95            ldy   #<fOpcod
                95            eom
2256: 20 C4 22  95            jsr   fRegReg    ;  and do it.
                95            eom
                95            fin
                95            eom
                95            eom
                96            Fmul  FR0;FR3;   ; FR3 = -impj*dx
                96            fOP   $23;FR0;FR3;
                96            do    FR0/8      ; If MR op
                96            fFPTR ]3         ;  set fptr (if needed)
                96            fMR   ]1;]2;]4   ;  do MR &op;&fmt;&FRd
                96            else            ; If RR op
                96            fRR   $23;FR0;FR3 ;  do RR &op;&FRs;&FRd
                96   ]FfRR    equ   1          ; Include fRegReg code.
```

```
                    96   ]Freg    equ   FR0*$400   ; Shift FRs
                    96            fOPC  0;]Freg;FR3;$23 ; Load opcode
                    96   fOpcod   equ   FR3*$80+0+]Freg+$23
2259: A2 01         96            ldx   #>fOpcod
225B: A0 A3         96            ldy   #<fOpcod
                    96            eom
225D: 20 C4 22      96            jsr   fRegReg    ;  and do it.
                    96            eom
                    96            fin
                    96            eom
                    96            eom
                    97            Fidx  fD;VX;j    ; fptr -> VX(j)
                    97   ]FfIDX   equ   ]FfIDX.fDb ; Include fIDXfD
2260: A2 6F         97            ldx   #<VX
2262: A0 20         97            ldy   #>VX
2264: AD 02 20      97            lda   j
2267: 20 AE 23      97            jsr   fIDXfD
                    97            eom
                    98            Fadd  fD;*;FR3   ; FR3 = VX(j)
                    98            fOP   $22;fD;*;FR3
                    98            do    fD/8       ; If MR op
                    98            fFPTR *          ;  set fptr (if needed)
                    98            if    *=* ; Already set if addr="*".
                    98            else             ; If addr <> "*"
                    98            lda   #<]1       ;  set fptr = &addr
                    98            sta   fptr
                    98            lda   #>]1
                    98            sta   fptr+1
                    98            fin
                    98            eom
                    98            fMR   $22;fD;FR3 ;  do MR &op;&fmt;&FRd
                    98   ]FfMR    equ   ]FfMR.fDb  ; Include fMemReg fD code
                    98            fOPC  $4000;fD-fF;FR3;$22 ; Load opcode
                    98   fOpcod   equ   FR3*$80+$4000+fD-fF+$22
226A: A2 55         98            ldx   #>fOpcod
226C: A0 A2         98            ldy   #<fOpcod
                    98            eom
226E: 20 06 23      98            jsr   fMemReg    ;  and do it.
                    98            eom
                    98            else             ; If RR op
                    98            fRR   ]1;]2;]3   ;  do RR &op;&FRs;&FRd
                    98            fin
                    98            eom
                    98            eom
                    99            Fmove FR3;fD;*   ; VX(j) = VX(j)-impj*dx
                    99            do    FR3/8      ; If MR op
                    99            fOP   0;]1;]2;]3 ;  do MR Fmove.
                    99            else
                    99            do    fD/8       ; If RM op
                    99   ]FfRM    equ   ]FfRM.fDb  ; Include fRegMem fD code
                    99            fFPTR *          ;  set fptr (if needed)
                    99            if    *=* ; Already set if addr="*".
                    99            else             ; If addr <> "*"
                    99            lda   #<]1       ;  set fptr = &addr
                    99            sta   fptr
                    99            lda   #>]1
                    99            sta   fptr+1
                    99            fin
                    99            eom
                    99            do    fD-fP
                    99            fOPC  $6000;fD-fF;FR3;0 ; Kfac = 0 if not fP
                    99   fOpcod   equ   FR3*$80+$6000+fD-fF+0
2271: A2 75         99            ldx   #>fOpcod
2273: A0 80         99            ldy   #<fOpcod
                    99            eom
                    99            else
                    99            fOPC  $6000;]2-fF;]1;]Fkfac ; Use Kfac if fP
```

```
                    99            fin
2275: 20 73 23      99            jsr   fRegMem    ;  and do it.
                    99            else              ; If RR op
                    99            fRR   0;]1;]2     ;  do RR Fmove.
                    99            fin
                    99            fin
                    99            eom
                    100
                    101           Fneg  FR5;FR3;    ; FR3 = -impi
                    101           fOP   $1A;FR5;FR3;
                    101           do    FR5/8       ; If MR op
                    101           fFPTR ]3          ;  set fptr (if needed)
                    101           fMR   ]1;]2;]4    ;  do MR &op;&fmt;&FRd
                    101           else              ; If RR op
                    101           fRR   $1A;FR5;FR3 ;  do RR &op;&FRs;&FRd
                    101   ]FfRR    equ   1           ; Include fRegReg code.
                    101   ]Freg    equ   FR5*$400    ; Shift FRs
                    101           fOPC  0;]Freg;FR3;$1A ; Load opcode
                    101   fOpcod   equ   FR3*$80+0+]Freg+$1A
2278: A2 15         101           ldx   #>fOpcod
227A: A0 9A         101           ldy   #<fOpcod
                    101           eom
227C: 20 C4 22      101           jsr   fRegReg     ;  and do it.
                    101           eom
                    101           fin
                    101           eom
                    101           eom
                    102           Fmul  FR1;FR3;    ; FR3 = -impi*dy
                    102           fOP   $23;FR1;FR3;
                    102           do    FR1/8       ; If MR op
                    102           fFPTR ]3          ;  set fptr (if needed)
                    102           fMR   ]1;]2;]4    ;  do MR &op;&fmt;&FRd
                    102           else              ; If RR op
                    102           fRR   $23;FR1;FR3 ;  do RR &op;&FRs;&FRd
                    102   ]FfRR    equ   1           ; Include fRegReg code.
                    102   ]Freg    equ   FR1*$400    ; Shift FRs
                    102           fOPC  0;]Freg;FR3;$23 ; Load opcode
                    102   fOpcod   equ   FR3*$80+0+]Freg+$23
227F: A2 05         102           ldx   #>fOpcod
2281: A0 A3         102           ldy   #<fOpcod
                    102           eom
2283: 20 C4 22      102           jsr   fRegReg     ;  and do it.
                    102           eom
                    102           fin
                    102           eom
                    102           eom
                    103           Fidx  fD;VY;i     ; fptr -> VY(i)
                    103   ]FfIDX   equ   ]FfIDX.fDb ; Include fIDXfD
2286: A2 97         103           ldx   #<VY
2288: A0 20         103           ldy   #>VY
228A: AD 01 20      103           lda   i
228D: 20 AE 23      103           jsr   fIDXfD
                    103           eom
                    104           Fadd  fD;*;FR3    ; FR3 = VY(i)-impi*dy
                    104           fOP   $22;fD;*;FR3
                    104           do    fD/8        ; If MR op
                    104           fFPTR *           ;  set fptr (if needed)
                    104           if    *=* ; Already set if addr="*".
                    104           else              ; If addr <> "*"
                    104           lda   #<]1        ;  set fptr = &addr
                    104           sta   fptr
                    104           lda   #>]1
                    104           sta   fptr+1
                    104           fin
                    104           eom
                    104           fMR   $22;fD;FR3 ;  do MR &op;&fmt;&FRd
                    104   ]FfMR    equ   ]FfMR.fDb   ; Include fMemReg fD code
```

```
                  104           fOPC  $4000;fD-fF;FR3;$22 ; Load opcode
                  104  fOpcod   equ   FR3*$80+$4000+fD-fF+$22
2290: A2 55       104           ldx   #>fOpcod
2292: A0 A2       104           ldy   #<fOpcod
                  104           eom
2294: 20 06 23    104           jsr   fMemReg    ;  and do it.
                  104           eom
                  104           else            ; If RR op
                  104           fRR   ]1;]2;]3   ;  do RR &op;&FRs;&FRd
                  104           fin
                  104           eom
                  104           eom
                  105           Fmove FR3;fD;*   ; VY(i) = VY(i)-impi*dy
                  105           do    FR3/8      ; If MR op
                  105           fOP   0;]1;]2;]3 ;  do MR Fmove.
                  105           else
                  105           do    fD/8       ; If RM op
                  105  ]FfRM     equ   ]FfRM.fDb  ; Include fRegMem fD code
                  105           fFPTR *          ;  set fptr (if needed)
                  105           if    *=* ; Already set if addr="*".
                  105           else            ; If addr <> "*"
                  105           lda   #<]1       ;  set fptr = &addr
                  105           sta   fptr
                  105           lda   #>]1
                  105           sta   fptr+1
                  105           fin
                  105           eom
                  105           do    fD-fP
                  105           fOPC  $6000;fD-fF;FR3;0 ; Kfac = 0 if not fP
                  105  fOpcod   equ   FR3*$80+$6000+fD-fF+0
2297: A2 75       105           ldx   #>fOpcod
2299: A0 80       105           ldy   #<fOpcod
                  105           eom
                  105           else
                  105           fOPC  $6000;]2-fF;]1;]Fkfac ; Use Kfac if fP
                  105           fin
229B: 20 73 23    105           jsr   fRegMem    ;  and do it.
                  105           else            ; If RR op
                  105           fRR   0;]1;]2    ;  do RR Fmove.
                  105           fin
                  105           fin
                  105           eom
                  106
                  107           Fneg  FR6;FR3;   ; FR3 = -impj
                  107           fOP   $1A;FR6;FR3;
                  107           do    FR6/8      ; If MR op
                  107           fFPTR ]3         ;  set fptr (if needed)
                  107           fMR   ]1;]2;]4   ;  do MR &op;&fmt;&FRd
                  107           else            ; If RR op
                  107           fRR   $1A;FR6;FR3 ;  do RR &op;&FRs;&FRd
                  107  ]FfRR     equ   1          ; Include fRegReg code.
                  107  ]Freg     equ   FR6*$400   ; Shift FRs
                  107           fOPC  0;]Freg;FR3;$1A ; Load opcode
                  107  fOpcod   equ   FR3*$80+0+]Freg+$1A
229E: A2 19       107           ldx   #>fOpcod
22A0: A0 9A       107           ldy   #<fOpcod
                  107           eom
22A2: 20 C4 22    107           jsr   fRegReg    ;  and do it.
                  107           eom
                  107           fin
                  107           eom
                  107           eom
                  108           Fmul  FR1;FR3;   ; FR3 = -impj*dy
                  108           fOP   $23;FR1;FR3;
                  108           do    FR1/8      ; If MR op
                  108           fFPTR ]3         ;  set fptr (if needed)
                  108           fMR   ]1;]2;]4   ;  do MR &op;&fmt;&FRd
```

```
                    108           else             ; If RR op
                    108           fRR   $23;FR1;FR3 ;  do RR &op;&FRs;&FRd
                    108  ]FfRR    equ   1          ; Include fRegReg code.
                    108  ]Freg    equ   FR1*$400   ; Shift FRs
                    108           fOPC  0;]Freg;FR3;$23 ; Load opcode
                    108  fOpcod   equ   FR3*$80+0+]Freg+$23
22A5: A2 05         108           ldx   #>fOpcod
22A7: A0 A3         108           ldy   #<fOpcod
                    108           eom
22A9: 20 C4 22      108           jsr   fRegReg    ;  and do it.
                    108           eom
                    108           fin
                    108           eom
                    108           eom
                    109           Fidx  fD;VY;j    ; fptr -> VY(j)
                    109  ]FfIDX   equ   ]FfIDX.fDb ; Include fIDXfD
22AC: A2 97         109           ldx   #<VY
22AE: A0 20         109           ldy   #>VY
22B0: AD 02 20      109           lda   j
22B3: 20 AE 23      109           jsr   fIDXfD
                    109           eom
                    110           Fadd  fD;*;FR3   ; FR3 = VY(j)-impj*dy
                    110           fOP   $22;fD;*;FR3
                    110           do    fD/8       ; If MR op
                    110           fFPTR *          ;  set fptr (if needed)
                    110           if    *=* ; Already set if addr="*".
                    110           else             ; If addr <> "*"
                    110           lda   #<]1       ;  set fptr = &addr
                    110           sta   fptr
                    110           lda   #>]1
                    110           sta   fptr+1
                    110           fin
                    110           eom
                    110           fMR   $22;fD;FR3 ;  do MR &op;&fmt;&FRd
                    110  ]FfMR    equ   ]FfMR.fDb  ; Include fMemReg fD code
                    110           fOPC  $4000;fD-fF;FR3;$22 ; Load opcode
                    110  fOpcod   equ   FR3*$80+$4000+fD-fF+$22
22B6: A2 55         110           ldx   #>fOpcod
22B8: A0 A2         110           ldy   #<fOpcod
                    110           eom
22BA: 20 06 23      110           jsr   fMemReg    ;  and do it.
                    110           eom
                    110           else             ; If RR op
                    110           fRR   ]1;]2;]3   ;  do RR &op;&FRs;&FRd
                    110           fin
                    110           eom
                    110           eom
                    111           Fmove FR3;fD;*   ; VY(j) = VY(j)-impj*dy
                    111           do    FR3/8      ; If MR op
                    111           fOP   0;]1;]2;]3 ;  do MR Fmove.
                    111           else
                    111           do    fD/8       ; If RM op
                    111  ]FfRM    equ   ]FfRM.fDb  ; Include fRegMem fD code
                    111           fFPTR *          ;  set fptr (if needed)
                    111           if    *=* ; Already set if addr="*".
                    111           else             ; If addr <> "*"
                    111           lda   #<]1       ;  set fptr = &addr
                    111           sta   fptr
                    111           lda   #>]1
                    111           sta   fptr+1
                    111           fin
                    111           eom
                    111           do    fD-fP
                    111           fOPC  $6000;fD-fF;FR3;0 ; Kfac = 0 if not fP
                    111  fOpcod   equ   FR3*$80+$6000+fD-fF+0
22BD: A2 75         111           ldx   #>fOpcod
22BF: A0 80         111           ldy   #<fOpcod
```

```
                  111           eom
                  111           else
                  111           fOPC  $6000;]2-fF;]1;]Fkfac ; Use Kfac if fP
                  111           fin
22C1: 20 73 23    111           jsr   fRegMem    ;  and do it.
                  111           else            ; If RR op
                  111           fRR   0;]1;]2    ;  do RR Fmove.
                  111           fin
                  111           fin
                  111           eom
                  112
                  113           put   FPERUN
```

```
                        >2    ************************************************************
                        >3    *                                                          *
                        >4    *             FPERUN - FPE Runtime routines                *
                        >5    *                                                          *
                        >6    *                   Michael J. Mahon                       *
                        >7    *                     Sep 22, 2010                         *
                        >8    *                  Copyright (C) 2010                      *
                        >9    *                                                          *
                        >10   * Runtime routines to support FPE macros are assembled     *
                        >11   * conditionally to minimize the size of the runtime.       *
                        >12   *                                                          *
                        >13   ************************************************************
                        >14
                        >15            do    ]FfRR
                        >16   * fRegReg implements FPE Register-to-Register ops
                        >17
22C4: 8E C8 C0          >18   fRegReg  stx   FPcmd      ; Issue the op
22C7: 8C C9 C0          >19            sty   FPcmd+1
                        >20            fin
                        >21            do    ]FfRR.]FfMR.]FfRM
                        >22   fWEXIT   Fwait            ; Wait for completion
22CA: AD C0 C0          >22   wait     lda   FPresp     ; Keep waiting until
22CD: 0A                >22            asl
22CE: AD C1 C0          >22            lda   FPresp+1
22D1: B0 F7             >22            bcs   wait       ;  FPresp hi bit off.
                        >22            eom
22D3: 60                >23            rts              ;  and return.
                        >24
                        >25            fin
                        >26            do    ]FfMR
                        >27   * fMemReg implements FPE Memory-to-Register ops
                        >28
                        >29            do    fXb&]FfMR
                        >30   fMRX     lda   (fptr),y   ; Load 10-byte eXtended FP
                        >31            sta   FPopnd     ; 2-byte exponent
                        >32            iny
                        >33            lda   (fptr),y
                        >34            sta   FPopnd+1
                        >35            iny
                        >36            lda   #0         ; Fake 2 zero bytes
                        >37            sta   FPopnd+2
                        >38            sta   FPopnd+3   ; (fMRD puts last 8 bytes)
                        >39            fin
                        >40            do    fDb.fXb.fPb&]FfMR
22D4: B1 06             >41   fMRD     lda   (fptr),y   ; Load 8-byte Double FP
22D6: 8D CC C0          >42            sta   FPopnd
22D9: C8                >43            iny
22DA: B1 06             >44            lda   (fptr),y
22DC: 8D CD C0          >45            sta   FPopnd+1
22DF: C8                >46            iny
22E0: B1 06             >47            lda   (fptr),y
22E2: 8D CE C0          >48            sta   FPopnd+2
22E5: C8                >49            iny
22E6: B1 06             >50            lda   (fptr),y
22E8: 8D CF C0          >51            sta   FPopnd+3
22EB: C8                >52            iny
                        >53            fin
                        >54            do    fSb.fDb.fXb.fPb.fCb&]FfMR
22EC: B1 06             >55   fMRS     lda   (fptr),y   ; Load 4-byte Single FP
22EE: 8D CC C0          >56            sta   FPopnd
22F1: C8                >57            iny
22F2: B1 06             >58            lda   (fptr),y
22F4: 8D CD C0          >59            sta   FPopnd+1
22F7: C8                >60            iny
22F8: B1 06             >61            lda   (fptr),y
22FA: 8D CE C0          >62            sta   FPopnd+2
22FD: C8                >63            iny
```

```
22FE: B1 06    >64            lda   (fptr),y
2300: 8D CF C0 >65            sta   FPopnd+3
2303: 4C CA 22 >66            jmp   fWEXIT
               >67
               >68            fin
2306: 8E C8 C0 >69   fMemReg  stx   FPcmd      ; Issue the op
2309: 8C C9 C0 >70            sty   FPcmd+1
               >71            Fwtdata          ; Wait to xfer data
230C: AD C0 C0 >71   wait     lda   FPresp     ; Check FPE status.
230F: AC C1 C0 >71            ldy   FPresp+1
2312: D0 04    >71            bne   done       ; Ready if FPresp <> $8900
2314: C9 89    >71            cmp   #$89
2316: F0 F4    >71            beq   wait       ;  else keep waiting...
               >71   done     eom
2318: A0 00    >72            ldy   #0         ; Prepare for first byte
231A: 8A       >73            txa              ; Recover storage format
               >74            do    fCb&]FfMR  ; If ctl reg load
               >75            bmi   fMRS       ; Ctl is 4 bytes big-endian
               >76            fin
               >77            do    fDb.fSb.fXb.fPb.fBb.fWb.fLb&]FfMR
231B: 29 1C    >78            and   #$1C       ; Isolate storage format.
               >79            fin
               >80            fVECT MR;D       ; Double
               >80            do    fDb&]FfMR  ; Include if used
231D: C9 14    >80            cmp   #>fD
231F: F0 B3    >80            beq   fMRD
               >80            fin
               >80            eom
               >81            fVECT MR;S       ; Single
               >81            do    fSb&]FfMR  ; Include if used
               >81            cmp   #>f]2
               >81            beq   f]1]2
               >81            fin
               >81            eom
               >82            fVECT MR;X       ; Extended
               >82            do    fXb&]FfMR  ; Include if used
               >82            cmp   #>f]2
               >82            beq   f]1]2
               >82            fin
               >82            eom
               >83            fVECT MR;P       ; BCD (12 bytes)
               >83            do    fPb&]FfMR  ; Include if used
2321: C9 0C    >83            cmp   #>fP
2323: F0 01    >83            beq   fMRP
               >83            fin
               >83            eom
               >84            do    fBb.fWb.fLb&]FfMR
               >85            ldx   #0         ; Prepare for little-endian
               >86            fVECT MR;B       ; Byte
               >87            fin
               >88            do    fWb&]FfMR
               >89            iny              ; y=1
               >90            fVECT MR;W       ; Word int (2 bytes)
               >91            fin
               >92            do    fLb&]FfMR
               >93            ldy   #3         ; y=3
               >94            fVECT MR;L       ; Long int (4 bytes)
               >95            fin
2325: 00       >96            brk              ; *** Unexpected mem fmt ***
               >97
               >98            do    fPb&]FfMR
2326: B1 06    >99   fMRP     lda   (fptr),y   ; Load 12-byte packed BCD FP
2328: 8D CC C0 >100           sta   FPopnd
232B: C8       >101           iny
232C: B1 06    >102           lda   (fptr),y
232E: 8D CD C0 >103           sta   FPopnd+1
2331: C8       >104           iny
```

```
2332: B1 06    >105           lda   (fptr),y
2334: 8D CE C0 >106           sta   FPopnd+2
2337: C8       >107           iny
2338: B1 06    >108           lda   (fptr),y
233A: 8D CF C0 >109           sta   FPopnd+3
233D: C8       >110           iny
233E: 4C D4 22 >111           jmp   fMRD       ; Finish last 8 bytes
               >112
               >113           fin
               >114           do    fLb&]FfMR
               >115  fMRL     lda   (fptr),y   ; Load 4-byte integer
               >116           inx
               >117           sta   FPopnd,x
               >118           dey
               >119           lda   (fptr),y
               >120           inx
               >121           sta   FPopnd,x
               >122           dey
               >123           fin
               >124           do    fWb&]FfMR
               >125  fMRW     lda   (fptr),y   ; Load 2-byte integer
               >126           inx
               >127           sta   FPopnd,x
               >128           dey
               >129           fin
               >130           do    fBb&]FfMR
               >131  fMRB     lda   (fptr),y   ; Load 1-byte integer
               >132           sta   FPopnd,x
               >133           jmp   fWEXIT
               >134           fin
               >135
               >136           fin   ]FfMR
               >137           do    ]FfRM
               >138  * fRegMem implements FPE Register-to-Memory ops
               >139
               >140           do    fXb&]FfRM
               >141  fRMX     lda   FPopnd     ; Store 10-byte eXtended FP
               >142           sta   (fptr),y   ; 2-byte exponent
               >143           iny
               >144           lda   FPopnd+1
               >145           sta   (fptr),y
               >146           iny
               >147           lda   FPopnd+2   ; Discard 2 zero bytes
               >148           lda   FPopnd+3   ; fRMD gets last 8 bytes.
               >149           fin
               >150           do    fDb.fXb.fPb&]FfRM
2341: AD CC C0 >151  fRMD     lda   FPopnd     ; Store 8-byte Double FP
2344: 91 06    >152           sta   (fptr),y
2346: C8       >153           iny
2347: AD CD C0 >154           lda   FPopnd+1
234A: 91 06    >155           sta   (fptr),y
234C: C8       >156           iny
234D: AD CE C0 >157           lda   FPopnd+2
2350: 91 06    >158           sta   (fptr),y
2352: C8       >159           iny
2353: AD CF C0 >160           lda   FPopnd+3
2356: 91 06    >161           sta   (fptr),y
2358: C8       >162           iny
               >163           fin
               >164           do    fSb.fDb.fXb.fPb.fCb&]FfRM
2359: AD CC C0 >165  fRMS     lda   FPopnd     ; Store 4-byte Single FP
235C: 91 06    >166           sta   (fptr),y
235E: C8       >167           iny
235F: AD CD C0 >168           lda   FPopnd+1
2362: 91 06    >169           sta   (fptr),y
2364: C8       >170           iny
2365: AD CE C0 >171           lda   FPopnd+2
```

```
2368: 91 06    >172           sta   (fptr),y
236A: C8       >173           iny
236B: AD CF C0 >174           lda   FPopnd+3
236E: 91 06    >175           sta   (fptr),y
2370: 4C CA 22 >176           jmp   fWEXIT
               >177
               >178           fin
2373: 8E C8 C0 >179  fRegMem  stx   FPcmd      ; Issue the op
2376: 8C C9 C0 >180           sty   FPcmd+1
               >181           Fwtdata          ; Wait to xfer data
2379: AD C0 C0 >181  wait     lda   FPresp     ; Check FPE status.
237C: AC C1 C0 >181           ldy   FPresp+1
237F: D0 04    >181           bne   done       ; Ready if FPresp <> $8900
2381: C9 89    >181           cmp   #$89
2383: F0 F4    >181           beq   wait       ;  else keep waiting...
               >181  done     eom
2385: A0 00    >182           ldy   #0         ; Prepare for first byte
2387: 8A       >183           txa              ; Recover storage format
               >184           do    fCb&]FfRM  ; If ctl reg store
               >185           bmi   fRMS       ; Ctl is 4 bytes big-endian
               >186           fin
               >187           do    fDb.fSb.fXb.fPb.fBb.fWb.fLb&]FfRM
2388: 29 1C    >188           and   #$1C       ; Isolate storage format.
               >189           fin
               >190           fVECT RM;D       ; Double
               >190           do    fDb&]FfRM  ; Include if used
238A: C9 14    >190           cmp   #>fD
238C: F0 B3    >190           beq   fRMD
               >190           fin
               >190           eom
               >191           fVECT RM;S       ; Single
               >191           do    fSb&]FfRM  ; Include if used
               >191           cmp   #>f]2
               >191           beq   f]1]2
               >191           fin
               >191           eom
               >192           fVECT RM;X       ; Extended
               >192           do    fXb&]FfRM  ; Include if used
               >192           cmp   #>f]2
               >192           beq   f]1]2
               >192           fin
               >192           eom
               >193           fVECT RM;P       ; BCD (12 bytes)
               >193           do    fPb&]FfRM  ; Include if used
238E: C9 0C    >193           cmp   #>fP
2390: F0 01    >193           beq   fRMP
               >193           fin
               >193           eom
               >194           do    fBb.fWb.fLb&]FfRM
               >195           ldx   #0         ; Prepare for little-endian
               >196           fVECT RM;B       ; Byte
               >197           fin
               >198           do    fWb&]FfRM
               >199           iny              ; y=1
               >200           fVECT RM;W       ; Word int (2 bytes)
               >201           fin
               >202           do    fLb&]FfRM
               >203           ldy   #3         ; y=3
               >204           fVECT RM;L       ; Long int (4 bytes)
               >205           fin
2392: 00       >206           brk              ; *** Unexpected mem fmt ***
               >207
               >208           do    fPb&]FfRM
2393: AD CC C0 >209  fRMP     lda   FPopnd     ; Store 12-byte packed BCD FP
2396: 91 06    >210           sta   (fptr),y
2398: C8       >211           iny
2399: AD CD C0 >212           lda   FPopnd+1
```

```
239C: 91 06    >213           sta   (fptr),y
239E: C8       >214           iny
239F: AD CE C0 >215           lda   FPopnd+2
23A2: 91 06    >216           sta   (fptr),y
23A4: C8       >217           iny
23A5: AD CF C0 >218           lda   FPopnd+3
23A8: 91 06    >219           sta   (fptr),y
23AA: C8       >220           iny
23AB: 4C 41 23 >221           jmp   fRMD       ; Finish last 8 bytes
               >222
               >223           fin
               >224           do    fLb&]FfRM
               >225  fRML     lda   FPopnd,x   ; Store 4-byte integer
               >226           inx
               >227           sta   (fptr),y
               >228           dey
               >229           lda   FPopnd,x
               >230           inx
               >231           sta   (fptr),y
               >232           dey
               >233           fin
               >234           do    fWb&]FfRM
               >235  fRMW     lda   FPopnd,x   ; Store 2-byte integer
               >236           inx
               >237           sta   (fptr),y
               >238           dey
               >239           fin
               >240           do    fBb&]FfRM
               >241  fRMB     lda   FPopnd,x   ; Store 1-byte integer
               >242           sta   (fptr),y
               >243           jmp   fWEXIT
               >244           fin
               >245
               >246           fin   ]FfRM
               >247           do    ]FfCOND
               >248  * Fcond maps FPE condition codes to 6502 flags
               >249  *
               >250  *  FPE Infinity ==> 6502 Overflow
               >251  *  FPE NaN      ==> 6502 Carry
               >252
               >253  * 6502 Flag definitions (NV_B DIZC)
               >254
               >255  f65N     equ   $80        ; Negative
               >256  f65V     equ   $40        ; Overflow
               >257  f65B     equ   $10        ; Break
               >258  f65D     equ   $08        ; Decimal
               >259  f65I     equ   $04        ; Interrupt enable
               >260  f65Z     equ   $02        ; Zero
               >261  f65C     equ   $01        ; Carry
               >262
               >263  F65flags db    0          ;          0000 FPE condition
               >264           db    f65C       ;          000n      (NZIn)
               >265           db    f65V       ;          00I0
               >266           db    f65V+f65C  ;          00In
               >267           db    f65Z       ;          0Z00
               >268           db    f65Z+f65C  ;          0Z0n
               >269           db    f65Z+f65V  ;          0ZI0
               >270           db    f65Z+f65V+f65C ;      0ZIn
               >271           db    f65N       ;          N000
               >272           db    f65N+f65C  ;          N00n
               >273           db    f65N+f65V  ;          N0I0
               >274           db    f65N+f65V+f65C ;      N0In
               >275           db    f65N+f65Z  ;          NZ00
               >276           db    f65N+f65Z+f65C ;      NZ0n
               >277           db    f65N+f65Z+f65V ;      NZI0
               >278           db    f65N+f65Z+f65V+f65C ; NZIn
               >279
```

```
                >280  fCOND    lda   #FCstat*4+$A0 ; Read FP status reg
                >281           sta   FPcmd
                >282           lda   #0
                >283           sta   FPcmd+1
                >284           Fwtdata
                >285           ldx   FPopnd     ; FP cond byte
                >286           stx   $00        ; **** debug ****
                >287           lda   FPopnd+1   ; Discard
                >288           sta   $01        ; **** debug ****
                >289           lda   FPopnd+2   ;  other 3
                >290           sta   $02        ; **** debug ****
                >291           lda   FPopnd+3   ;   status bytes.
                >292           sta   $03        ; **** debug ****
                >293           php              ; Get 6502 flags
                >294           pla
                >295           and   #f65D+f65I ; Isolate D & I state
                >296           ora   F65flags,x ; OR in condition
                >297           pha              ; Set 6502 flags.
                >298           plp
                >299           rts              ; Return.
                >300           fin
                >301
                >302           fin
                >303           do    ]FfIDX
                >304  ************************************************************
                >305  *                                                          *
                >306  *   Subscript routines:  Set fptr = fLEN * index + addr     *
                >307  *     where: addr = (X,Y)  and  index = (A).               *
                >308  *                                                          *
                >309  *   Called by: Fidx   &fQ;&addr;&index                     *
                >310  *                                                          *
                >311  ************************************************************
                >312
                >313           do    fBb&]FfIDX
                >314  fIDXfB   sta   fptr       ; fB: fptr = index + addr
                >315           lda   #0
                >316           beq   fPaddr     ; (always)
                >317
                >318           fin
                >319           do    fWb&]FfIDX
                >320  fIDXfW   sta   fptr       ; fI: fptr = 2 * index + addr
                >321           lda   #0
                >322           beq   fX2Paddr   ; (always)
                >323
                >324           fin
                >325           do    fLb.fSb&]FfIDX
                >326  fIDXfL   equ   *          ; fL: fptr = 4 * index + addr
                >327  fIDXfS   sta   fptr       ; fS: fptr = 4 * index + addr
                >328           lda   #0
                >329           beq   fX4Paddr   ; (always)
                >330
                >331           fin
23AE: 0A        >332  fIDXfD   asl              ; fD: fptr = 8 * index + addr
23AF: 85 06     >333           sta   fptr
23B1: A9 00     >334           lda   #0
23B3: 2A        >335           rol
23B4: 06 06     >336  fX4Paddr asl   fptr       ; (fptr,A) = (fptr,A) * 2
23B6: 2A        >337           rol
23B7: 06 06     >338  fX2Paddr asl   fptr       ; fptr = (fptr,A) * 2
23B9: 2A        >339           rol
23BA: 85 07     >340  fPaddr   sta   fptr+1     ; fptr = (fptr,A) + addr
23BC: 8A        >341           txa
23BD: 65 06     >342           adc   fptr
23BF: 85 06     >343           sta   fptr
23C1: 98        >344           tya
23C2: 65 07     >345           adc   fptr+1
23C4: 85 07     >346           sta   fptr+1
```

```
23C6: 60       >347           rts
               >348
               >349           fin
               >350           do    fXb&]FfIDX
               >351  fIDXfX   pha              ; fX: fptr = 10 * index + addr
               >352           sta   fptr       ; (fptr,A) = index
               >353           lda   #0
               >354           asl   fptr       ; (fptr,A) = index * 2
               >355           rol
               >356           asl   fptr       ; fptr = index * 4
               >357           rol
               >358           sta   fptr+1
               >359           pla              ; (fptr,A) = index * 5
               >360           adc   fptr
               >361           sta   fptr
               >362           lda   fptr+1
               >363           adc   #0
               >364           bcc   fX2Paddr   ; (always)
               >365
               >366           fin
               >367           do    fPb&]FfIDX
               >368  fIDXfP   pha              ; fP: fptr = 12 * index + addr
               >369           sta   fptr       ; (fptr,A) = index
               >370           lda   #0
               >371           asl   fptr       ; fptr = index * 2
               >372           rol
               >373           sta   fptr+1
               >374           pla              ; (fptr,A) = index * 3
               >375           adc   fptr
               >376           sta   fptr
               >377           lda   fptr+1
               >378           adc   #0
               >379           bcc   fX4Paddr   ; (always)
               >380           fin
               >381           fin
               >382
               >383           do    ]FfPRINT
               >384  ************************************************************
               >385  *                                                          *
               >386  * Print packed BCD value of fBCD in exponential format     *
               >387  * using 24 characters: '-3.14159265358979323e+000'         *
               >388  *                                                          *
               >389  *   Called by: Fprint &FRs                                 *
               >390  *                                                          *
               >391  ************************************************************
               >392
               >393  fCOUT    equ   $FDED      ; COUT monitor routine
               >394
               >395  fBCD     ds    fPl        ; Packed BCD FP value
               >396  fEnd     db    0          ; End index
               >397
               >398  fPRINT   bit   fBCD       ; Mantissa sign
               >399           bpl   :mpos
               >400           lda   #"-"
               >401           bne   :msign     ; (always)
               >402
               >403  :mpos    lda   #"         "
               >404  :msign   jsr   fCOUT      ; Print sign
               >405           ldy   #12        ; Index of last+1 mant byte
               >406           sty   fEnd
               >407           ldy   #3         ; Index of first byte
               >408           jsr   :prsec     ; Start with second digit
               >409           lda   #"e"       ; Print exponent
               >410           jsr   fCOUT
               >411           lda   fBCD       ; Get exponent sign
               >412           asl
               >413           bpl   :epos
```

```
>414           lda   #"-"
>415           bne   :esign     ; (always)
>416
>417  :epos    lda   #"+"
>418  :esign   jsr   fCOUT      ; Print exponent sign
>419           ldy   #2         ; Index of last+1 exp byte
>420           sty   fEnd
>421           ldy   #0         ; Index of first exp byte
>422           beq   :prsec     ; (always) returns to caller.
>423
>424  :prbcd   lda   fBCD,y     ; Get first BCD digit,
>425           lsr              ;  isolate it in
>426           lsr              ;   low nibble,
>427           lsr
>428           lsr
>429           ora   #"0"       ;    cvt to ASCII,
>430           jsr   fCOUT      ;     and print it.
>431  :prsec   lda   fBCD,y     ; Get second BCD digit,
>432           and   #$0F       ;  isolate it,
>433           ora   #"0"       ;   cvt to ASCII,
>434           jsr   fCOUT      ;    and print it.
>435           cpy   #3         ; Insert decimal point?
>436           bne   :nopt      ; -No.
>437           lda   #"."       ; -Yes, do it.
>438           jsr   fCOUT
>439  :nopt    iny              ; Move to next byte
>440           cpy   fEnd       ; Are we done?
>441           bcc   :prbcd     ; -No, keep printing.
>442           rts              ; -Yes, return.
>443           fin   ]FfPRINT
```

```
--End assembly, 967 bytes, Errors: 0
```

```
Symbol table - alphabetical order:
```

```
   BCD     =$2013   ?  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     ?  FCcntl  =$04
?  FCstat  =$02        FPE     =$C0C0      FPEslot =$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   ?  FPrest  =$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        FR1     =$01
   FR2     =$02        FR3     =$03        FR4     =$04        FR5     =$05
   FR6     =$06        FR7     =$07     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?Fetoxm1 =$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        G       =$200B      M       =$20BF
?  N       =$2000      Nmax    =$05        VX      =$206F      VY      =$2097
   X       =$201F      Y       =$2047   V? ]Fdebug =$00     V  ]Fdeflt =$00
```

```
V  ]FfCOND =$00     V  ]FfIDX  =$20     V  ]FfMR   =$28     V  ]FfPRINT=$00
V  ]FfRM   =$28     V  ]FfRR   =$01     V  ]Fkfac  =$11     MV ]Freg   =$0400
M  done    =$2385      dt      =$2003   ?  fB      =$180F      fBb     =$40
?  fBl     =$01        fCb     =$80     ?  fCl     =$04        fD      =$140F
   fDb     =$20        fDl     =$08        fF      =$0F     MD fFPTR   =$8000
   fIDXfD  =$23AE   ?  fL      =$0F        fLb     =$01     ?  fLl     =$04
MD fMR     =$8000      fMRD    =$22D4      fMRP    =$2326   ?  fMRS    =$22EC
   fMemReg =$2306   MD fOP     =$8000   MD fOPC    =$8000   M  fOpcod  =$7580
   fP      =$0C0F   ?  fPaddr  =$23BA      fPb     =$08        fPl     =$0C
   fRMD    =$2341      fRMP    =$2393   ?  fRMS    =$2359   MD fRR     =$8000
   fRegMem =$2373      fRegReg =$22C4   ?  fS      =$040F      fSb     =$02
?  fSl     =$04     MD fVECT   =$8000   ?  fW      =$100F      fWEXIT  =$22CA
   fWb     =$10     ?  fWl     =$02     ?  fX      =$080F   ?  fX2Paddr=$23B7
?  fX4Paddr=$23B4      fXb     =$04     ?  fXl     =$0A        fptr    =$06
   i       =$2001      j       =$2002   ?  loop    =$2138   M  wait    =$2379
```

Symbol table - numerical order:

```
   FR0     =$00        Fpi     =$00     ?  FPrnear =$00     ?  FPrpX   =$00
V  ]Fdeflt =$00     V  ]FfCOND =$00     V  ]FfPRINT=$00     V? ]Fdebug =$00
?  fBl     =$01        FR1     =$01     ?  FPNaN   =$01     ?  FPinexd =$01
   fLb     =$01     V  ]FfRR   =$01     ?  fWl     =$02        FR2     =$02
?  FCstat  =$02     ?  FPinf   =$02     ?  FPinexop=$02        fSb     =$02
   FR3     =$03        FPEslot =$04     ?  fLl     =$04     ?  fSl     =$04
?  fCl     =$04        FR4     =$04     ?  FPzero  =$04     ?  FPdz    =$04
?  FCcntl  =$04        fXb     =$04        FR5     =$05        Nmax    =$05
   fptr    =$06        FR6     =$06        FR7     =$07        fDl     =$08
?  FPneg   =$08     ?  FPunfl  =$08     ?  FPinexA =$08        fPb     =$08
?  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     V  ]FfIDX  =$20     V  ]FfMR   =$28
V  ]FfRM   =$28     ?  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     MV ]Freg   =$0400   ?  fS      =$040F
?  fX      =$080F      fP      =$0C0F   ?  fW      =$100F      fD      =$140F
?  fB      =$180F   ?  N       =$2000      i       =$2001      j       =$2002
   dt      =$2003      G       =$200B      BCD     =$2013      X       =$201F
   Y       =$2047      VX      =$206F      VY      =$2097      M       =$20BF
?  loop    =$2138      fRegReg =$22C4      fWEXIT  =$22CA      fMRD    =$22D4
?  fMRS    =$22EC      fMemReg =$2306      fMRP    =$2326      fRMD    =$2341
?  fRMS    =$2359      fRegMem =$2373   M  wait    =$2379   M  done    =$2385
   fRMP    =$2393      fIDXfD  =$23AE   ?  fX4Paddr=$23B4   ?  fX2Paddr=$23B7
?  fPaddr  =$23BA   M  fOpcod  =$7580   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 Fsqrt   =$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
```