

!A

LLOAD CXFPMATH.L,A\$4000

*** End of Pass 1

LLOAD CXFPMATH.1.L,A\$4000

LLOAD CXFPMATH.2.L,A\$4000

LLOAD CXFPMATH.3.L,A\$4000

LLOAD CXFPMATH.4.L,A\$4000

LLOAD CXFPMATH.5.L,A\$4000

LLOAD CXFPMATH.L,A\$4000

*** End of Pass 2

```

0800      1          ttl "CXFPMATH Routines, CXFPMATH.L"
0800      2          src "CXFPMATH.L"
0800      3      ;
0800      4      ;
0800      5      ; CXFPMATH.L
0800      6      ;
0800      7      ;
0008      8  REALPTRS  epz $08                ; 0x08:0x0D
0018      9  IMAGPTRS  epz $18                ; 0x18:0x1D
001E     10  COUNT      epz $1E
001F     11  OFFSET     epz $1F
006F     12  FRETOP     epz $6F
0073     13  HIMEM      epz $73
0092     14  ARGGUARD   epz $92
009D     15  FACEXP     epz $9D                ; FAC exponent
009E     16  FACMANT    epz $9E                ; FAC mantissa byte 1
00A2     17  FACSIGN    epz $A2
00AA     18  ARGSIGN    epz $AA
00AB     19  XORSIGN    epz $AB
00AC     20  FACGUARD   epz $AC
00B1     21  CHRGET     epz $B1
00B7     22  CHRGOT     epz $B7
0800     23      ;
0800     24          enz
0800     25      ;
0000     26  ZERO       equ $00
007F     27  MSBCLR     equ $7F
0080     28  MSBSET     equ $80
00FF     29  NEGONE     equ $FF
0800     30      ;
0004     31  ARRAYSIZ   equ $04                ; number of 2x2 array elements
0004     32  VARS2      equ $04                ; number of bytes for 2 values
0005     33  IMAGOFF    equ $05                ; imaginary to real FP offset
0005     34  FPLEN      equ $05                ; floating point number length
0005     35  CMDLEN     equ $05                ; length of all input commands
0006     36  ARGLEN     equ $06                ; real and imaginary arguments
0006     37  VARS3      equ $06                ; number of bytes for 3 values
000A     38  CXFPOFF    equ IMAGOFF*2          ; next complex array value
0800     39      ;
008D     40  RETURN     equ $8D
009B     41  ESCAPE     equ $9B
00A0     42  SPACE      equ $A0
00A2     43  QUOTE      equ $A2
00AC     44  COMMA      equ $AC
0800     45      ;
0100     46  PAGESIZE   equ $100
0800     47      ;
03D0     48  DOSWARM    equ $3D0
0800     49      ;
C000     50  KEY        equ $C000
C010     51  CLRKEY     equ $C010
0800     52      ;
C082     53  ROM2WP     equ $C082
0800     54      ;
DFE3     55  PTRGET     equ $DFE3
0800     56      ;
E7A7     57  FSUB       equ $E7A7                ; add ARG to negative FAC
E7BE     58  FADD       equ $E7BE                ; add ARG to FAC
E97F     59  FMULT      equ $E97F                ; multiply ARG and FAC
E9E3     60  LOADARG    equ $E9E3                ; load ARG from memory

```

```

EA66      61  FDIV      equ  $EA66      ; divide FAC by ARG
EAF9      62  LOADFAC  equ  $EAF9      ; load FAC from memory
EB2B      63  COPYFAC  equ  $EB2B      ; copy FAC to memory
EB53      64  MOVFA    equ  $EB53      ; copy ARG to FAC
EB90      65  FSGN     equ  $EB90      ; evaluate FAC for 0<,<=0,>0
EE8D      66  FSQR     equ  $EE8D      ; compute square root of FAC
EE97      67  FEXP     equ  $EE97      ; raise ARG to power of FAC
EFEA      68  FCOS     equ  $EFEA      ; compute cosine of FAC in rad
EFF1      69  FSIN     equ  $EFF1      ; compute sine of FAC in rad
0800      70  ;
F03A      71  FTAN     equ  $F03A      ; calc tangent of FAC in rad
F09E      72  FATAN    equ  $F09E      ; compute arctangent of FAC
0800      73  ;
FD8E      74  CROUT    equ  $FD8E
FD8E      75  PRBYTE   equ  $FD8E
FDDA      76  PRHEX    equ  $FDE3
FDE3      77  COUT     equ  $FDED
0800      78  ;
0800      79  ;
B000      80          org  $B000
B000      81          obj  $900
B000      82          usr
B000      83  ;
B000      84  ;
B000      85  ; Initialization Table INITBL and Jump Table JMPTBL for all
B000      86  ; Input Commands.
B000      87  ;
B000 4C 00 B1 88  INITBL    jmp  LOADER
B003 4C 37 B1 89          jmp  LIBENT
B006 4C 7A B1 90          jmp  LINKER
B009      91  ;
B009 4C 4D B2 92  JMPTBL    jmp  LGNOT
B00C 4C 5D B2 93          jmp  LGAND
B00F 4C 61 B2 94          jmp  LGORA
B012 4C 65 B2 95          jmp  LGEOR
B015      96  ;
B015 4C 7D B2 97          jmp  CXADD
B018 4C A7 B2 98          jmp  CXSUB
B01B 4C D1 B2 99          jmp  CXMUL
B01E 4C 33 B3 100         jmp  CXDIV
B021      101 ;
B021 4C F0 B3 102         jmp  CX2PL
B024 4C 6D B4 103         jmp  PL2CX
B027      104 ;
B027 4C 9D B4 105         jmp  AYADD
B02A 4C B2 B4 106         jmp  AYSUB
B02D 4C C7 B4 107         jmp  AYMUL
B030 4C 28 B5 108         jmp  AYINV
B033      109 ;
B033 4C 0D B6 110         jmp  CSMUL
B036      111 ;
B036      112 ;
B036      113         dfs  $40-*)&NEGONE,ZERO
B040      114 ;
B040      115 ;
B040      116 ; Hash Value Table HASHTBL for all Input Commands.
B040      117 ;
B040 36      118 HASHTBL    byt  $36      ; LGNOT
B041 EF      119          byt  $EF      ; LGAND
B042 2D      120          byt  $2D      ; LGORA
B043 0F      121          byt  $0F      ; LGEOR

```

```

B044      122 ;
B044 D3    123      byt $D3      ; CXADD
B045 3C    124      byt $3C      ; CXSUB
B046 2E    125      byt $2E      ; CXMUL
B047 FB    126      byt $FB      ; CXDIV
B048      127 ;
B048 B7    128      byt $B7      ; CX2PL
B049 19    129      byt $19      ; PL2CX
B04A      130 ;
B04A BB    131      byt $BB      ; AYADD
B04B 24    132      byt $24      ; AYSUB
B04C 15    133      byt $15      ; AIMUL
B04D 01    134      byt $01      ; AYINV
B04E      135 ;
B04E 05    136      byt $05      ; CSMUL
B04F      137 ;
000F      138 HASHSIZ equ *-HASHTBL
B04F      139 ;
B04F      140 ;
B04F      141 ; Pointer Length Table PTRLNTBL for all Input Commands.
B04F      142 ;
B04F 04    143 PTRLNTBL byt VARS2      ; LGNOT
B050 06    144      byt VARS3      ; LGAND
B051 06    145      byt VARS3      ; LGORA
B052 06    146      byt VARS3      ; LGEOR
B053      147 ;
B053 06    148      byt VARS3      ; CXADD
B054 06    149      byt VARS3      ; CXSUB
B055 06    150      byt VARS3      ; CXMUL
B056 06    151      byt VARS3      ; CXDIV
B057      152 ;
B057 04    153      byt VARS2      ; CX2PL
B058 04    154      byt VARS2      ; PL2CX
B059      155 ;
B059 06    156      byt VARS3      ; AYADD
B05A 06    157      byt VARS3      ; AYSUB
B05B 06    158      byt VARS3      ; AIMUL
B05C 06    159      byt VARS3      ; AYINV
B05D      160 ;
B05D 06    161      byt VARS3      ; CSMUL
B05E      162 ;
B05E      163 ;
B05E      164      dfs $70-*)&NEGONE,ZERO
B070      165 ;
B070      166 ;
B070      167 ; Working variables.
B070      168 ;
B070      169 FIRSTIME dfs 1,ZERO
B071      170 HASHVAL  dfs 1,ZERO
B072      171 CMDNDX   dfs 1,ZERO
B073      172 PTRNUM    dfs 1,ZERO
B074      173 PTRNDX    dfs 1,ZERO
B075      174 PTRLEN    dfs 1,ZERO
B076      175 SGNFLG1   dfs 1,ZERO
B077      176 SGNFLG2   dfs 1,ZERO
B078      177 ;
B078      178 ;
B078      179 ; Real and Imaginary coefficient addresses.
B078      180 ;
B078      181 FPREAL:
B078      182 REAL1L    dfs 1,ZERO

```

```

B079      183  REAL1H    dfs 1,ZERO
B07A      184  REAL2L    dfs 1,ZERO
B07B      185  REAL2H    dfs 1,ZERO
B07C      186  REAL3L    dfs 1,ZERO
B07D      187  REAL3H    dfs 1,ZERO
B07E      188  ;
B07E      189  FPIMAG:
B07E      190  IMAG1L    dfs 1,ZERO
B07F      191  IMAG1H    dfs 1,ZERO
B080      192  IMAG2L    dfs 1,ZERO
B081      193  IMAG2H    dfs 1,ZERO
B082      194  IMAG3L    dfs 1,ZERO
B083      195  IMAG3H    dfs 1,ZERO
B084      196  ;
B084      197  ;
B084      198  ; Temporary locations for intermediate results.
B084      199  ;
B084      200  TEMP1      dfs FPLEN,ZERO
B089      201  TEMP2      dfs FPLEN,ZERO
B08E      202  ;
B08E      203  TEMPR1     dfs FPLEN,ZERO
B093      204  TEMPR2     dfs FPLEN,ZERO
B098      205  TEMPI1     dfs FPLEN,ZERO
B09D      206  TEMPI2     dfs FPLEN,ZERO
B0A2      207  ;
B0A2      208  ;
B0A2      209  ; Offsets that are used by AYADD and AYSUB in BLDARGS1.
B0A2      210  ;
B0A2 00 14 0A 211  ADDSUB    byt ZERO,CXFPOFF*2,CXFPOFF,CXFPOFF*3 ; X and Y
B0A5 1E
B0A6      212  ;
B0A6      213  ;
B0A6      214  ; Offsets and temporary memory that is used by AYMUL and
B0A6      215  ; BLDARGS2.
B0A6      216  ;
B0A6 00 14 00 217  MULOFF1   byt ZERO,CXFPOFF*2,ZERO,CXFPOFF*2 ; X
B0A9 14
B0AA 0A 1E 0A 218           byt CXFPOFF,CXFPOFF*3,CXFPOFF,CXFPOFF*3
B0AD 1E
B0AE 00 0A 14 219  MULOFF2   byt ZERO,CXFPOFF,CXFPOFF*2,CXFPOFF*3 ; Y
B0B1 1E
B0B2 00 0A 14 220           byt ZERO,CXFPOFF,CXFPOFF*2,CXFPOFF*3
B0B5 1E
B0B6      221  ;
B0B6 8E 93 8E 222  MULLSB1   byt TEMPR1,TEMPR2,TEMPR1,TEMPR2 ; real values
B0B9 93
B0BA 8E 93 8E 223           byt TEMPR1,TEMPR2,TEMPR1,TEMPR2
B0BD 93
B0BE 98 9D 98 224  MULLSB2   byt TEMPI1,TEMPI2,TEMPI1,TEMPI2 ; imaginary values
B0C1 9D
B0C2 98 9D 98 225           byt TEMPI1,TEMPI2,TEMPI1,TEMPI2
B0C5 9D
B0C6      226  ;
B0C6 00 14 0A 227  MULOUT    byt ZERO,CXFPOFF*2,CXFPOFF,CXFPOFF*3 ; Z
B0C9 1E
B0CA      228  ;
B0CA      229  ;
B0CA      230  ; Offsets and temporary memory that is used by AYINV and
B0CA      231  ; BLDARGS3 for computing a determinant.
B0CA      232  ;
B0CA 00 0A 233  DETOFF1    byt ZERO,CXFPOFF ; a and c

```

```

B0CC 1E 14      234 DETOFF2   byt CXFPOFF*3,CXFPOFF*2 ; d and b
B0CE           235 ;
B0CE 8E 93      236 DETLSB1   byt TEMPR1,TEMPR2      ; real values
B0D0 98 9D      237 DETLSB2   byt TEMPI1,TEMPI2      ; imaginary values
B0D2           238 ;
B0D2           239 ;
B0D2           240 ; Offsets and temporary memory that is used by AYINV and
B0D2           241 ; BLDARGS4 for computing an inverse matrix.
B0D2           242 ;
B0D2 1E 14 0A   243 INVOFF1   byt CXFPOFF*3,CXFPOFF*2,CXFPOFF,ZERO ; X
B0D5 00
B0D6 00 14 0A   244 INVOFF2   byt ZERO,CXFPOFF*2,CXFPOFF,CXFPOFF*3 ; Z
B0D9 1E
B0DA           245 ;
B0DA 00 E2 E2   246 INVLSB    byt ZERO,MINUSONE,MINUSONE,ZERO ; minus values
B0DD 00
B0DE           247 ;
B0DE           248 ;
B0DE           249 ; Offsets that are used by CSMUL and BLDARGS5 for
B0DE           250 ; multiplying a complex array by a complex scalar.
B0DE           251 ;
B0DE 00 0A 14   252 CSOFF     byt ZERO,CXFPOFF,CXFPOFF*2,CXFPOFF*3 ; X and Z
B0E1 1E
B0E2           253 ;
B0E2           254 ;
B0E2           255 ; Floating point values that are needed for computations.
B0E2           256 ;
B0E2 81 80 00   257 MINUSONE hex 8180000000
B0E5 00 00
B0E7 E0 7F FF   258 HIGHVAL   hex E07FFFFFFF
B0EA FF FF
B0EC 81 49 0F   259 PIDIV2    hex 81490FDAA2
B0EF DA A2
B0F1 82 49 0F   260 PI         hex 82490FDAA2
B0F4 DA A2
B0F6           261 ;
B0F6           262 ;
B0F6           263 ;         dfs $FE-*)&NEGONE,ZERO
B0FE           264 ;
B0FE           265 ;
B0FE 45 06      266 VERSION   byt $45,$06          ; Version #4.5.06
B100           267 ;
B100           268 ;
B100           269 ; Parse the Applesoft command line for the CXFPMATH
B100           270 ; routine name and obtain the memory addresses for two
B100           271 ; input complex variables and for one output complex
B100           272 ; variable. Enter the selected CXFPMATH routine and
B100           273 ; process the data. FIRSTIME initializes FRETOP in order
B100           274 ; to protect CXMATH.
B100           275 ;
B100 AD 70 B0   276 LOADER    lda FIRSTIME
B103 D0 09      277             bne >1
B105           278 ;
B105 CE 70 B0   279             dec FIRSTIME
B108           280 ;
B108 85 6F      281             sta FRETOP
B10A           282 ;
B10A A9 B0      283             lda /INITBL
B10C 85 70      284             sta FRETOP+1
B10E           285 ;
B10E 20 37 B1   286 ^1          jsr LIBENT

```

```

B111      287 ;
B111 AE 72 B0 288      ldx CMDNDX
B114      289 ;
B114 BD 4F B0 290      lda PTRLNTBL,X
B117 8D 75 B0 291      sta PTRLEN
B11A      292 ;
B11A 20 7A B1 293      jsr LINKER
B11D      294 ;
B11D AD 72 B0 295      lda CMDNDX
B120 0A      296      asl
B121      297 ;
B121 6D 72 B0 298      adc CMDNDX
B124 AA      299      tax
B125      300 ;
B125 BD 0A B0 301      lda JMPTBL+1,X
B128 8D 35 B1 302      sta TEMPMOD+1
B12B      303 ;
B12B BD 0B B0 304      lda JMPTBL+2,X
B12E 8D 36 B1 305      sta TEMPMOD+2
B131      306 ;
B131 2C 82 C0 307      bit ROM2WP
B134      308 ;
B134 4C 00 00 309      TEMPMOD jmp *-*
B137      310 ;
B137      311 ;
B137      312 ; Parse the Applesoft command line for the CXFPMATH
B137      313 ; routine name and create a hash value from that name.
B137      314 ; Match the hash value in HASHTBL and save its index value.
B137      315 ;
B137      316 ; Terminate processing if there are no variables, if the
B137      317 ; routine name length is wrong, or if the hash value is not
B137      318 ; found in HASHTBL.
B137      319 ;
B137 A2 00      320      LIBENT ldx #ZERO
B139 8E 71 B0 321      stx HASHVAL
B13C      322 ;
B13C 20 B1 00 323      ^1 jsr CHRGET
B13F F0 2E      324      beq >5
B141      325 ;
B141 C9 2C      326      cmp #COMMA&MSBCLR
B143 F0 13      327      beq >2
B145      328 ;
B145 C9 22      329      cmp #QUOTE&MSBCLR
B147 F0 F3      330      beq <1
B149      331 ;
B149 18      332      clc
B14A      333 ;
B14A E9 30      334      sbc #"0"&MSBCLR
B14C      335 ;
B14C 0E 71 B0 336      asl HASHVAL
B14F 6D 71 B0 337      adc HASHVAL
B152 8D 71 B0 338      sta HASHVAL
B155      339 ;
B155 E8      340      inx
B156 D0 E4      341      bne <1 ; always taken
B158      342 ;
B158 E0 05      343      ^2 cpx #CMDLEN
B15A D0 16      344      bne >6
B15C      345 ;
B15C A2 0E      346      ldx #HASHSIZ-1
B15E      347 ;

```

```

B15E AD 71 B0      348      lda HASHVAL
B161              349      ;
B161 DD 40 B0      350      ^3      cmp HASHTBL,X
B164 F0 05         351      beq >4
B166              352      ;
B166 CA           353      dex
B167 10 F8         354      bpl <3
B169              355      ;
B169 30 0A         356      bmi >7
B16B              357      ;
B16B 8E 72 B0      358      ^4      stx CMDNDX
B16E              359      ;
B16E 60            360      rts
B16F              361      ;
B16F A0 00         362      ^5      ldy #ERR1-ERRS
B171              363      ;
B171 2C 00 00      364      bit *-*
B174              365      dfs !-2
B172              366      ;
B172 A0 26         367      ^6      ldy #ERR2-ERRS
B174              368      ;
B174 2C 00 00      369      bit *-*
B177              370      dfs !-2
B175              371      ;
B175 A0 3F         372      ^7      ldy #ERR3-ERRS
B177              373      ;
B177 4C C5 B1      374      jmp PRNTERR
B17A              375      ;
B17A              376      ;
B17A              377      ; Parse the Applesoft command line after the CXFPMATH
B17A              378      ; routine name to obtain the memory addresses for two input
B17A              379      ; complex variables and for one output complex variable.
B17A              380      ;
B17A              381      ; Terminate processing if the number of complex variables
B17A              382      ; is wrong.
B17A              383      ;
B17A A2 00         384      LINKER   ldx #ZERO
B17C              385      ;
B17C 8E 74 B0      386      ^1      stx PTRNDX
B17F              387      ;
B17F 20 B7 00      388      jsr CHRGOT
B182 F0 36         389      beq >4
B184              390      ;
B184 20 B1 00      391      jsr CHRGET
B187 F0 31         392      beq >4
B189              393      ;
B189 C9 2C         394      cmp #COMMA&MSBCLR
B18B F0 29         395      beq >3
B18D              396      ;
B18D 20 E3 DF      397      jsr PTRGET
B190              398      ;
B190 AE 74 B0      399      ldx PTRNDX
B193 EC 75 B0      400      cpx PTRLEN
B196 B0 1E         401      bcs >3
B198              402      ;
B198 95 08         403      sta REALPTRS+0,X
B19A 94 09         404      sty REALPTRS+1,X
B19C              405      ;
B19C 9D 78 B0      406      sta FPREAL+0,X
B19F 48            407      pha
B1A0              408      ;

```



```

B1A0 98          409          tya
B1A1 9D 79 B0    410          sta FPREAL+1,X
B1A4             411          ;
B1A4 18          412          clc
B1A5             413          ;
B1A5 68          414          pla
B1A6 69 05       415          adc #IMAGOFF
B1A8 90 01       416          bcc >2
B1AA             417          ;
B1AA C8          418          iny
B1AB             419          ;
B1AB 95 18       420          ^2      sta IMAGPTRS+0,X
B1AD 94 19       421          sty IMAGPTRS+1,X
B1AF             422          ;
B1AF 9D 7E B0    423          sta FPIMAG+0,X
B1B2             424          ;
B1B2 98          425          tya
B1B3 9D 7F B0    426          sta FPIMAG+1,X
B1B6             427          ;
B1B6 E8          428          ^3      inx
B1B7 E8          429          inx
B1B8             430          ;
B1B8 D0 C2       431          bne <1          ; always taken
B1BA             432          ;
B1BA AE 74 B0    433          ^4      ldx PTRNDX
B1BD EC 75 B0    434          cpx PTRLEN
B1C0 D0 01       435          bne >5
B1C2             436          ;
B1C2 60          437          rts
B1C3             438          ;
B1C3 A0 5D       439          ^5      ldy #ERR4-ERRS          ; fall into PRNTERR
B1C5             440          ;
B1C5             441          ;
B1C5             442          ; Print the Error Message in ERRS given by Y-reg.
B1C5             443          ;
B1C5 B9 D3 B1    444          PRNTERR lda ERRS,Y
B1C8 F0 06       445          beq >1
B1CA             446          ;
B1CA 20 ED FD    447          jsr COUT
B1CD             448          ;
B1CD C8          449          iny
B1CE D0 F5       450          bne PRNTERR
B1D0             451          ;
B1D0 4C D0 03    452          ^1      jmp DOSWARM
B1D3             453          ;
B1D3             454          ;
B1D3             455          ; Error messages.
B1D3             456          ;
B1D3             457          ERRS:
B1D3             458          ;
B1D3 8D          459          ERR1      byt RETURN
B1D4 CE EF A0    460          asc "No variables included with command."
B1D7 F6 E1 F2
B1DA E9 E1 E2
B1DD EC E5 F3
B1E0 A0 E9 EE
B1E3 E3 EC F5
B1E6 E4 E5 E4
B1E9 A0 F7 E9
B1EC F4 E8 A0
B1EF E3 EF ED

```

```
B1F2 ED E1 EE
B1F5 E4 AE
B1F7 8D 00      461      byt RETURN,ZERO
B1F9           462      ;
B1F9 8D         463      ERR2      byt RETURN
B1FA C3 EF ED   464      asc "Command size is wrong."
B1FD ED E1 EE
B200 E4 A0 F3
B203 E9 FA E5
B206 A0 E9 F3
B209 A0 F7 F2
B20C EF EE E7
B20F AE
B210 8D 00      465      byt RETURN,ZERO
B212           466      ;
B212 8D         467      ERR3      byt RETURN
B213 C3 EF ED   468      asc "Command not found in table."
B216 ED E1 EE
B219 E4 A0 EE
B21C EF F4 A0
B21F E6 EF F5
B222 EE E4 A0
B225 E9 EE A0
B228 F4 E1 E2
B22B EC E5 AE
B22E 8D 00      469      byt RETURN,ZERO
B230           470      ;
B230 8D         471      ERR4      byt RETURN
B231 D7 F2 EF   472      asc "Wrong number of arguments."
B234 EE E7 A0
B237 EE F5 ED
B23A E2 E5 F2
B23D A0 EF E6
B240 A0 E1 F2
B243 E7 F5 ED
B246 E5 EE F4
B249 F3 AE
B24B 8D 00      473      byt RETURN,ZERO
B24D           474      ;
B24D           475      ;
B24D           476      icl "CXFPMATH.1.L"
```

```
LLOAD CXFPMATH.1.L,A$4000
```

```

B24D          1          ttl "CXFPMATH.1 Routines, CXFPMATH.1.L"
B24D          2          ;
B24D          3          ;
B24D          4          ; CXFPMATH.1.L
B24D          5          ;
B24D          6          ;
B24D          7          ; Logical operation routines.  These routines operate on
B24D          8          ; integers only and they are written as:
B24D          9          ;
B24D         10          ; CALL "LGNOT",X%,Z%                where X% NOT form Z%
B24D         11          ;
B24D         12          ; CALL "LGAND",X%,Y%,Z%            where X% AND Y% form Z%
B24D         13          ; CALL "LGORA",X%,Y%,Z%            where X% OR  Y% form Z%
B24D         14          ; CALL "LGEOR",X%,Y%,Z%            where X% EOR Y% form Z%
B24D         15          ;
B24D         16          ;
B24D         17          ; Calculate the NOT logical operator.
B24D         18          ;
B24D A0 00     19  LGNOT    ldy #ZERO
B24F          20          ;
B24F B1 08     21          lda (REALPTRS+0),Y    ; get MSB of X
B251 49 FF     22          eor #NEGONE           ; operate on MSB of X
B253 91 0A     23          sta (REALPTRS+2),Y    ; save result to MSB of Z
B255          24          ;
B255 C8        25          iny
B256          26          ;
B256 B1 08     27          lda (REALPTRS+0),Y    ; get LSB of X
B258 49 FF     28          eor #NEGONE           ; operate on LSB of X
B25A 91 0A     29          sta (REALPTRS+2),Y    ; save result to LSB of Z
B25C          30          ;
B25C 60        31          rts
B25D          32          ;
B25D          33          ;
B25D          34          ; Get the logical operator AND and branch to LGCALC.
B25D          35          ;
B25D A9 00     36  LGAND    lda #ZERO
B25F          37          dfs !-1
B25E          38          ;
B25E 31 08     39          and (REALPTRS),Y
B260          40          dfs !-1
B25F          41          ;
B25F D0 06     42          bne LGCALC                ; always taken
B261          43          ;
B261          44          ;
B261          45          ; Get the logical operator OR  and branch to LGCALC.
B261          46          ;
B261 A9 00     47  LGORA    lda #ZERO
B263          48          dfs !-1
B262          49          ;
B262 11 08     50          ora (REALPTRS),Y
B264          51          dfs !-1
B263          52          ;
B263 D0 02     53          bne LGCALC                ; always taken
B265          54          ;
B265          55          ;
B265          56          ; Get the logical operator EOR and fall into LGCALC.
B265          57          ;
B265 A9 00     58  LGEOR    lda #ZERO
B267          59          dfs !-1
B266          60          ;

```

```
B266 51 08      61      eor (REALPTRS),Y
B268           62      dfs !-1
B267           63      ;
B267           64      ;
B267 8D 71 B2    65  LGCALC   sta LGMOD1
B26A 8D 78 B2    66      sta LGMOD2
B26D           67      ;
B26D A0 00      68      ldy #ZERO
B26F           69      ;
B26F           70      ;
B26F           71      ; The values in Integer Arrays are in MSB,LSB order.
B26F           72      ;
B26F B1 08      73      lda (REALPTRS+0),Y ; get MSB of X
B271 31 0A      74  LGMOD1   and (REALPTRS+2),Y ; operate on MSB of Y
B273 91 0C      75      sta (REALPTRS+4),Y ; save result to MSB of Z
B275           76      ;
B275 C8         77      iny
B276           78      ;
B276 B1 08      79      lda (REALPTRS+0),Y ; get LSB of X
B278 31 0A      80  LGMOD2   and (REALPTRS+2),Y ; operate on LSB of Y
B27A 91 0C      81      sta (REALPTRS+4),Y ; save result to LSB of Z
B27C           82      ;
B27C 60         83      rts
B27D           84      ;
B27D           85      ;
B27D           86      icl "CXFPMATH.2.L"
```

LLOAD CXFPMATH.2.L,A\$4000

```

B27D      1          ttl "CXFPMATH.2 Routines, CXFPMATH.2.L"
B27D      2      ;
B27D      3      ;
B27D      4      ; CXFPMATH.2.L
B27D      5      ;
B27D      6      ;
B27D      7      ; Complex CXADD routine.  This routine adds two complex
B27D      8      ; floating point numbers and is written as:
B27D      9      ;
B27D     10      ; CALL "CXADD",X(0),Y(0),Z(0)              where X + Y form Z
B27D     11      ;
B27D     12      ; The real coefficients are added separately from the
B27D     13      ; imaginary coefficients.  The imaginary coefficients
B27D     14      ; follow the real coefficients in memory.  All floating
B27D     15      ; point numbers are five bytes in size.
B27D     16      ;
B27D     17      ; Put Y into FAC, add in X, and save to Z.
B27D     18      ;
B27D A5 0A     19 CXADD      lda REALPTRS+2
B27F A4 0B     20          ldy REALPTRS+3
B281      21      ;
B281 20 F9 EA   22          jsr LOADFAC
B284      23      ;
B284 A5 08     24          lda REALPTRS+0
B286 A4 09     25          ldy REALPTRS+1
B288      26      ;
B288 20 BE E7   27          jsr FADD
B28B      28      ;
B28B A6 0C     29          ldx REALPTRS+4
B28D A4 0D     30          ldy REALPTRS+5
B28F      31      ;
B28F 20 2B EB   32          jsr COPYFAC
B292      33      ;
B292      34      ;
B292      35      ; Add the imaginary coefficients.
B292      36      ;
B292 A5 1A     37          lda IMAGPTRS+2
B294 A4 1B     38          ldy IMAGPTRS+3
B296      39      ;
B296 20 F9 EA   40          jsr LOADFAC
B299      41      ;
B299 A5 18     42          lda IMAGPTRS+0
B29B A4 19     43          ldy IMAGPTRS+1
B29D      44      ;
B29D 20 BE E7   45          jsr FADD
B2A0      46      ;
B2A0 A6 1C     47          ldx IMAGPTRS+4
B2A2 A4 1D     48          ldy IMAGPTRS+5
B2A4      49      ;
B2A4 4C 2B EB   50          jmp COPYFAC
B2A7      51      ;
B2A7      52      ;
B2A7      53      ; Complex CXSUB routine.  This routine subtracts two
B2A7      54      ; complex floating point numbers and is written as:
B2A7      55      ;
B2A7      56      ; CALL "CXSUB",X(0),Y(0),Z(0)              where X - Y form Z
B2A7      57      ;
B2A7      58      ; The real coefficients are subtracted separately from the
B2A7      59      ; imaginary coefficients.  The imaginary coefficients
B2A7      60      ; follow the real coefficients in memory.  All floating

```

```

B2A7      61 ; point numbers are five bytes in size.
B2A7      62 ;
B2A7      63 ; Put Y into FAC, negate FAC, add in X, and save to Z.
B2A7      64 ;
B2A7 A5 0A 65 CXSUB      lda REALPTRS+2
B2A9 A4 0B 66           ldy REALPTRS+3
B2AB      67 ;
B2AB 20 F9 EA 68           jsr LOADFAC
B2AE      69 ;
B2AE A5 08 70           lda REALPTRS+0
B2B0 A4 09 71           ldy REALPTRS+1
B2B2      72 ;
B2B2 20 A7 E7 73           jsr FSUB
B2B5      74 ;
B2B5 A6 0C 75           ldx REALPTRS+4
B2B7 A4 0D 76           ldy REALPTRS+5
B2B9      77 ;
B2B9 20 2B EB 78           jsr COPYFAC
B2BC      79 ;
B2BC      80 ;
B2BC      81 ; Subtract the imaginary coefficients.
B2BC      82 ;
B2BC A5 1A 83           lda IMAGPTRS+2
B2BE A4 1B 84           ldy IMAGPTRS+3
B2C0      85 ;
B2C0 20 F9 EA 86           jsr LOADFAC
B2C3      87 ;
B2C3 A5 18 88           lda IMAGPTRS+0
B2C5 A4 19 89           ldy IMAGPTRS+1
B2C7      90 ;
B2C7 20 A7 E7 91           jsr FSUB
B2CA      92 ;
B2CA A6 1C 93           ldx IMAGPTRS+4
B2CC A4 1D 94           ldy IMAGPTRS+5
B2CE      95 ;
B2CE 4C 2B EB 96           jmp COPYFAC
B2D1      97 ;
B2D1      98 ;
B2D1      99 ; Complex CXMUL routine. This routine multiplies two
B2D1     100 ; complex floating point numbers and is written as:
B2D1     101 ;
B2D1     102 ; CALL "CXMUL",X(0),Y(0),Z(0) where X * Y form Z
B2D1     103 ;
B2D1     104 ; The real and imaginary coefficients are combined and
B2D1     105 ; processed according to the following procedure:
B2D1     106 ;
B2D1     107 ; (a + bi) * (c + di) -> (ac - bd) + (ad + bc)i
B2D1     108 ;
B2D1     109 ; Put Y into FAC, multiply by X, and save to Z.
B2D1     110 ;
B2D1 A5 0A 111 CXMUL      lda REALPTRS+2
B2D3 A4 0B 112           ldy REALPTRS+3
B2D5     113 ;
B2D5 20 F9 EA 114           jsr LOADFAC
B2D8     115 ;
B2D8 A5 08 116           lda REALPTRS+0
B2DA A4 09 117           ldy REALPTRS+1
B2DC     118 ;
B2DC 20 7F E9 119           jsr FMULT
B2DF     120 ;
B2DF A2 84 121           ldx #TEMP1

```

```

B2E1 A0 B0      122          ldy /TEMP1
B2E3              123      ;
B2E3 20 2B EB    124          jsr COPYFAC
B2E6              125      ;
B2E6              126      ;
B2E6              127      ; Multiply imaginary coefficients b and d.
B2E6              128      ;
B2E6 A5 1A      129          lda IMAGPTRS+2
B2E8 A4 1B      130          ldy IMAGPTRS+3
B2EA              131      ;
B2EA 20 F9 EA    132          jsr LOADFAC
B2ED              133      ;
B2ED A5 18      134          lda IMAGPTRS+0
B2EF A4 19      135          ldy IMAGPTRS+1
B2F1              136      ;
B2F1 20 7F E9    137          jsr FMULT
B2F4              138      ;
B2F4              139      ;
B2F4              140      ; Combine real coefficients ac - bd.
B2F4              141      ;
B2F4 A9 84      142          lda #TEMP1
B2F6 A0 B0      143          ldy /TEMP1
B2F8              144      ;
B2F8 20 A7 E7    145          jsr FSUB
B2FB              146      ;
B2FB A6 0C      147          ldx REALPTRS+4
B2FD A4 0D      148          ldy REALPTRS+5
B2FF              149      ;
B2FF 20 2B EB    150          jsr COPYFAC
B302              151      ;
B302              152      ;
B302              153      ; Multiply coefficients b and c.
B302              154      ;
B302 A5 0A      155          lda REALPTRS+2
B304 A4 0B      156          ldy REALPTRS+3
B306              157      ;
B306 20 F9 EA    158          jsr LOADFAC
B309              159      ;
B309 A5 18      160          lda IMAGPTRS+0
B30B A4 19      161          ldy IMAGPTRS+1
B30D              162      ;
B30D 20 7F E9    163          jsr FMULT
B310              164      ;
B310 A2 84      165          ldx #TEMP1
B312 A0 B0      166          ldy /TEMP1
B314              167      ;
B314 20 2B EB    168          jsr COPYFAC
B317              169      ;
B317              170      ;
B317              171      ; Multiply coefficients a and d.
B317              172      ;
B317 A5 1A      173          lda IMAGPTRS+2
B319 A4 1B      174          ldy IMAGPTRS+3
B31B              175      ;
B31B 20 F9 EA    176          jsr LOADFAC
B31E              177      ;
B31E A5 08      178          lda REALPTRS+0
B320 A4 09      179          ldy REALPTRS+1
B322              180      ;
B322 20 7F E9    181          jsr FMULT
B325              182      ;

```

```

B325      183 ;
B325      184 ; Combine imaginary coefficients ad + bc.
B325      185 ;
B325 A9 84   186      lda #TEMP1
B327 A0 B0   187      ldy /TEMP1
B329      188 ;
B329 20 BE E7 189      jsr FADD
B32C      190 ;
B32C A6 1C   191      ldx IMAGPTRS+4
B32E A4 1D   192      ldy IMAGPTRS+5
B330      193 ;
B330 4C 2B EB 194      jmp COPYFAC
B333      195 ;
B333      196 ;
B333      197 ; Complex CXDIV routine. This routine divides two complex
B333      198 ; floating point numbers and is written as:
B333      199 ;
B333      200 ; CALL "CXDIV",X(0),Y(0),Z(0) where X / Y form Z
B333      201 ;
B333      202 ; The real and imaginary coefficients are combined and
B333      203 ; processed according to the following procedure:
B333      204 ;
B333      205 ; (a + bi) / (c + di) =
B333      206 ; (ac + bd) / (c*c + d*d) + (bc - ad)i / (c*c + d*d)
B333      207 ;
B333      208 ; Calculate the denominator (c*c + d*d).
B333      209 ;
B333 A5 1A   210 CXDIV   lda IMAGPTRS+2
B335 A4 1B   211      ldy IMAGPTRS+3
B337      212 ;
B337 20 F9 EA 213      jsr LOADFAC
B33A      214 ;
B33A A5 1A   215      lda IMAGPTRS+2
B33C A4 1B   216      ldy IMAGPTRS+3
B33E      217 ;
B33E 20 7F E9 218      jsr FMULT
B341      219 ;
B341 A2 84   220      ldx #TEMP1
B343 A0 B0   221      ldy /TEMP1
B345      222 ;
B345 20 2B EB 223      jsr COPYFAC
B348      224 ;
B348 A5 0A   225      lda REALPTRS+2
B34A A4 0B   226      ldy REALPTRS+3
B34C      227 ;
B34C 20 F9 EA 228      jsr LOADFAC
B34F      229 ;
B34F A5 0A   230      lda REALPTRS+2
B351 A4 0B   231      ldy REALPTRS+3
B353      232 ;
B353 20 7F E9 233      jsr FMULT
B356      234 ;
B356 A9 84   235      lda #TEMP1
B358 A0 B0   236      ldy /TEMP1
B35A      237 ;
B35A 20 BE E7 238      jsr FADD
B35D      239 ;
B35D A2 84   240      ldx #TEMP1
B35F A0 B0   241      ldy /TEMP1
B361      242 ;
B361 20 2B EB 243      jsr COPYFAC

```



```

B364      244 ;
B364      245 ;
B364      246 ; Combine coefficients for real numerator (ac + bd) and
B364      247 ; divide by the denominator that is saved in TEMP1.
B364      248 ;
B364      249 ; Multiply real coefficients a and c.
B364      250 ;
B364 A5 0A      251      lda REALPTRS+2
B366 A4 0B      252      ldy REALPTRS+3
B368      253 ;
B368 20 F9 EA    254      jsr LOADFAC
B36B      255 ;
B36B A5 08      256      lda REALPTRS+0
B36D A4 09      257      ldy REALPTRS+1
B36F      258 ;
B36F 20 7F E9    259      jsr FMULT
B372      260 ;
B372 A2 89      261      ldx #TEMP2
B374 A0 B0      262      ldy /TEMP2
B376      263 ;
B376 20 2B EB    264      jsr COPYFAC
B379      265 ;
B379      266 ;
B379      267 ; Multiply imaginary coefficients b and d.
B379      268 ;
B379 A5 1A      269      lda IMAGPTRS+2
B37B A4 1B      270      ldy IMAGPTRS+3
B37D      271 ;
B37D 20 F9 EA    272      jsr LOADFAC
B380      273 ;
B380 A5 18      274      lda IMAGPTRS+0
B382 A4 19      275      ldy IMAGPTRS+1
B384      276 ;
B384 20 7F E9    277      jsr FMULT
B387      278 ;
B387      279 ;
B387      280 ; Combine numerator coefficients (ac + bd) and save.
B387      281 ;
B387 A9 89      282      lda #TEMP2
B389 A0 B0      283      ldy /TEMP2
B38B      284 ;
B38B 20 BE E7    285      jsr FADD
B38E      286 ;
B38E A2 89      287      ldx #TEMP2
B390 A0 B0      288      ldy /TEMP2
B392      289 ;
B392 20 2B EB    290      jsr COPYFAC
B395      291 ;
B395      292 ;
B395      293 ; Recall the denominator and divide it into (ac + bc).
B395      294 ;
B395 A9 84      295      lda #TEMP1
B397 A0 B0      296      ldy /TEMP1
B399      297 ;
B399 20 F9 EA    298      jsr LOADFAC
B39C      299 ;
B39C A9 89      300      lda #TEMP2
B39E A0 B0      301      ldy /TEMP2
B3A0      302 ;
B3A0 20 66 EA    303      jsr FDIV
B3A3      304 ;

```

```

B3A3 A6 0C      305      ldx REALPTRS+4
B3A5 A4 0D      306      ldy REALPTRS+5
B3A7           307      ;
B3A7 20 2B EB   308      jsr COPYFAC
B3AA           309      ;
B3AA           310      ;
B3AA           311      ; Combine coefficients for imaginary numerator (bc - ad)
B3AA           312      ; and divide by the denominator that is saved in TEMP1.
B3AA           313      ;
B3AA           314      ; Multiply the coefficients b and c.
B3AA           315      ;
B3AA A5 0A      316      lda REALPTRS+2
B3AC A4 0B      317      ldy REALPTRS+3
B3AE           318      ;
B3AE 20 F9 EA   319      jsr LOADFAC
B3B1           320      ;
B3B1 A5 18      321      lda IMAGPTRS+0
B3B3 A4 19      322      ldy IMAGPTRS+1
B3B5           323      ;
B3B5 20 7F E9   324      jsr FMULT
B3B8           325      ;
B3B8 A2 89      326      ldx #TEMP2
B3BA A0 B0      327      ldy /TEMP2
B3BC           328      ;
B3BC 20 2B EB   329      jsr COPYFAC
B3BF           330      ;
B3BF           331      ;
B3BF           332      ; Multiply the coefficients a and d.
B3BF           333      ;
B3BF A5 1A      334      lda IMAGPTRS+2
B3C1 A4 1B      335      ldy IMAGPTRS+3
B3C3           336      ;
B3C3 20 F9 EA   337      jsr LOADFAC
B3C6           338      ;
B3C6 A5 08      339      lda REALPTRS+0
B3C8 A4 09      340      ldy REALPTRS+1
B3CA           341      ;
B3CA 20 7F E9   342      jsr FMULT
B3CD           343      ;
B3CD           344      ;
B3CD           345      ; Combine numerator coefficients (bc - ad) and save.
B3CD           346      ;
B3CD A9 89      347      lda #TEMP2
B3CF A0 B0      348      ldy /TEMP2
B3D1           349      ;
B3D1 20 A7 E7   350      jsr FSUB
B3D4           351      ;
B3D4 A2 89      352      ldx #TEMP2
B3D6 A0 B0      353      ldy /TEMP2
B3D8           354      ;
B3D8 20 2B EB   355      jsr COPYFAC
B3DB           356      ;
B3DB           357      ;
B3DB           358      ; Recall the denominator and divide it into (bc - ad).
B3DB           359      ;
B3DB A9 84      360      lda #TEMP1
B3DD A0 B0      361      ldy /TEMP1
B3DF           362      ;
B3DF 20 F9 EA   363      jsr LOADFAC
B3E2           364      ;
B3E2 A9 89      365      lda #TEMP2

```

```
B3E4 A0 B0      366      ldy /TEMP2
B3E6            367      ;
B3E6 20 66 EA   368      jsr FDIV
B3E9            369      ;
B3E9 A6 1C      370      ldx IMAGPTRS+4
B3EB A4 1D      371      ldy IMAGPTRS+5
B3ED            372      ;
B3ED 4C 2B EB   373      jmp COPYFAC
B3F0            374      ;
B3F0            375      ;
B3F0            376      icl "CXFPMATH.3.L"
```

```
LLOAD CXFPMATH.3.L,A$4000
```

```

B3F0      1          ttl "CXFPMATH.3 Routines, CXFPMATH.3.L"
B3F0      2      ;
B3F0      3      ;
B3F0      4      ; CXFPMATH.3.L
B3F0      5      ;
B3F0      6      ;
B3F0      7      ; Complex CX2PL routine.  This routine converts a complex
B3F0      8      ; number into its polar coordinates and is written as:
B3F0      9      ;
B3F0     10      ; CALL "CX2PL",X(0),Z(0)                      where X CX2PL forms Z
B3F0     11      ;
B3F0     12      ; The real and imaginary coefficients are processed
B3F0     13      ; according to the following procedure:
B3F0     14      ;
B3F0     15      ; magnitude = root(real*real + imag*imag)
B3F0     16      ;
B3F0     17      ; phase = atn(imag / real), when real > 0
B3F0     18      ;         = (pi/2) * sgn(imag), when real = 0
B3F0     19      ;         = atn(imag / real) + pi, when real < 0
B3F0     20      ;
B3F0     21      ; Compute the sign flag of the real number as 0, <0, or >0.
B3F0     22      ;
B3F0 A0 00     23 CX2PL      ldy #ZERO
B3F2      24      ;
B3F2 B1 08     25          lda (REALPTRS),Y
B3F4 29 7F     26          and #MSBCLR
B3F6      27      ;
B3F6 C8       28          iny
B3F7      29      ;
B3F7 11 08     30          ora (REALPTRS),Y
B3F9 8D 76 B0  31          sta SGNFLG1
B3FC      32      ;
B3FC      33      ;
B3FC      34      ; Calculate the magnitude of the complex number.
B3FC      35      ;
B3FC A5 18     36          lda IMAGPTRS+0
B3FE A4 19     37          ldy IMAGPTRS+1
B400      38      ;
B400 20 F9 EA  39          jsr LOADFAC
B403      40      ;
B403 A5 18     41          lda IMAGPTRS+0
B405 A4 19     42          ldy IMAGPTRS+1
B407      43      ;
B407 20 7F E9  44          jsr FMULT
B40A      45      ;
B40A A2 84     46          ldx #TEMP1
B40C A0 B0     47          ldy /TEMP1
B40E      48      ;
B40E 20 2B EB  49          jsr COPYFAC
B411      50      ;
B411 A5 08     51          lda REALPTRS+0
B413 A4 09     52          ldy REALPTRS+1
B415      53      ;
B415 20 F9 EA  54          jsr LOADFAC
B418      55      ;
B418 A5 08     56          lda REALPTRS+0
B41A A4 09     57          ldy REALPTRS+1
B41C      58      ;
B41C 20 7F E9  59          jsr FMULT
B41F      60      ;

```

```

B41F A9 84      61      lda #TEMP1
B421 A0 B0      62      ldy /TEMP1
B423           63      ;
B423 20 BE E7   64      jsr FADD
B426 20 8D EE   65      jsr FSQR
B429           66      ;
B429 A6 0A      67      ldx REALPTRS+2
B42B A4 0B      68      ldy REALPTRS+3
B42D           69      ;
B42D 20 2B EB   70      jsr COPYFAC
B430           71      ;
B430           72      ;
B430           73      ; Calculate the phase of the complex number when the
B430           74      ; magnitude of the complex number is zero.
B430           75      ;
B430 AD 76 B0   76      lda SGNFLG1
B433 D0 14      77      bne >1
B435           78      ;
B435 A5 18      79      lda IMAGPTRS+0
B437 A4 19      80      ldy IMAGPTRS+1
B439           81      ;
B439 20 F9 EA   82      jsr LOADFAC
B43C 20 90 EB   83      jsr FSGN
B43F           84      ;
B43F A9 EC      85      lda #PIDIV2
B441 A0 B0      86      ldy /PIDIV2
B443           87      ;
B443 20 7F E9   88      jsr FMULT
B446           89      ;
B446 4C 66 B4   90      jmp C2PEXIT
B449           91      ;
B449           92      ;
B449           93      ; Calculate the phase of the complex number when the
B449           94      ; magnitude of the complex number is not zero.
B449           95      ;
B449 A5 08      96      ^1   lda REALPTRS+0
B44B A4 09      97      ldy REALPTRS+1
B44D           98      ;
B44D 20 F9 EA   99      jsr LOADFAC
B450          100      ;
B450 A5 18     101      lda IMAGPTRS+0
B452 A4 19     102      ldy IMAGPTRS+1
B454          103      ;
B454 20 66 EA  104      jsr FDIV
B457 20 9E F0  105      jsr FATAN
B45A          106      ;
B45A          107      ;
B45A          108      ; Add the value PI if SGNFLG1 < 0.
B45A          109      ;
B45A 2C 76 B0  110      bit SGNFLG1
B45D 10 07     111      bpl C2PEXIT
B45F          112      ;
B45F A9 F1     113      lda #PI
B461 A0 B0     114      ldy /PI
B463          115      ;
B463 20 BE E7  116      jsr FADD
B466          117      ;
B466          118      ;
B466          119      ; Save FAC to the imaginary coefficient.
B466          120      ;
B466 A6 1A     121      C2PEXIT ldx IMAGPTRS+2

```

```

B468 A4 1B      122          ldy IMAGPTRS+3
B46A           123          ;
B46A 4C 2B EB   124          jmp COPYFAC
B46D           125          ;
B46D           126          ;
B46D           127          ; Complex PL2CX routine. This routine converts polar
B46D           128          ; coordinates into a complex number and is written as:
B46D           129          ;
B46D           130          ; CALL "PL2CX",X(0),Z(0)                      where X PL2CX forms Z
B46D           131          ;
B46D           132          ; The magnitude at REALPTRS and phase at IMAGPTRS values
B46D           133          ; are processed according to the following procedure:
B46D           134          ;
B46D           135          ; real number = magnitude * cos(phase)
B46D           136          ; imag number = magnitude * sin(phase)
B46D           137          ;
B46D           138          ; Compute the real coefficient.
B46D           139          ;
B46D A5 18      140 PL2CX     lda IMAGPTRS+0
B46F A4 19      141          ldy IMAGPTRS+1
B471           142          ;
B471 20 F9 EA   143          jsr LOADFAC
B474           144          ;
B474 20 EA EF   145          jsr FCOS
B477           146          ;
B477 A5 08      147          lda REALPTRS+0
B479 A4 09      148          ldy REALPTRS+1
B47B           149          ;
B47B 20 7F E9   150          jsr FMULT
B47E           151          ;
B47E A6 0A      152          ldx REALPTRS+2
B480 A4 0B      153          ldy REALPTRS+3
B482           154          ;
B482 20 2B EB   155          jsr COPYFAC
B485           156          ;
B485           157          ;
B485           158          ; Compute the imaginary coefficient.
B485           159          ;
B485 A5 18      160          lda IMAGPTRS+0
B487 A4 19      161          ldy IMAGPTRS+1
B489           162          ;
B489 20 F9 EA   163          jsr LOADFAC
B48C           164          ;
B48C 20 F1 EF   165          jsr FSIN
B48F           166          ;
B48F A5 08      167          lda REALPTRS+0
B491 A4 09      168          ldy REALPTRS+1
B493           169          ;
B493 20 7F E9   170          jsr FMULT
B496           171          ;
B496 A6 1A      172          ldx IMAGPTRS+2
B498 A4 1B      173          ldy IMAGPTRS+3
B49A           174          ;
B49A 4C 2B EB   175          jmp COPYFAC
B49D           176          ;
B49D           177          ;
B49D           178          icl "CXFPMATH.4.L"

```

LLOAD CXFPMATH.4.L,A\$4000

```

B49D      1          ttl "CXFPMATH.4 Routines, CXFPMATH.4.L"
B49D      2      ;
B49D      3      ;
B49D      4      ; CXFPMATH.4.L
B49D      5      ;
B49D      6      ;
B49D      7      ; Complex AYADD routine. This routine adds two complex
B49D      8      ; 2x2 floating point arrays and is written as:
B49D      9      ;
B49D     10      ; CALL "AYADD",X(0,0,0),Y(0,0,0),Z(0,0,0)          X + Y form Z
B49D     11      ;
B49D     12      ; The real coefficients are added separately from the
B49D     13      ; imaginary coefficients. The four imaginary coefficients
B49D     14      ; follow the four real coefficients for each array in
B49D     15      ; memory. All complex floating point array numbers are
B49D     16      ; five bytes in size.
B49D     17      ;
B49D     18      ; Put Y into FAC, add in X using CXADD, and save to Z.
B49D     19      ;
B49D A9 00     20 AYADD      lda #ZERO                ; initialize offset counter
B49F 85 1E     21          sta COUNT
B4A1 F0 03     22          beq >2                    ; always taken
B4A3          23      ;
B4A3 20 22 B6  24      ^1      jsr BLDARGS1
B4A6          25      ;
B4A6 20 7D B2  26      ^2      jsr CXADD
B4A9          27      ;
B4A9 E6 1E     28          inc COUNT
B4AB          29      ;
B4AB A5 1E     30          lda COUNT
B4AD C9 04     31          cmp #ARRAYSIZ
B4AF D0 F2     32          bne <1
B4B1          33      ;
B4B1 60        34          rts
B4B2          35      ;
B4B2          36      ;
B4B2          37      ; Complex AYSUB routine. This routine subtracts two
B4B2          38      ; complex 2x2 floating point arrays and is written as:
B4B2          39      ;
B4B2          40      ; CALL "AYSUB",X(0,0,0),Y(0,0,0),Z(0,0,0)      X - Y form Z
B4B2          41      ;
B4B2          42      ; The real coefficients are subtracted separately from the
B4B2          43      ; imaginary coefficients. The four imaginary coefficients
B4B2          44      ; follow the four real coefficients for each array in
B4B2          45      ; memory. All complex floating point array numbers are
B4B2          46      ; five bytes in size.
B4B2          47      ;
B4B2          48      ; Put Y into FAC, negate FAC, add in X using CXSUB, and
B4B2          49      ; save to Z.
B4B2          50      ;
B4B2 A9 00     51 AYSUB      lda #ZERO                ; initialize offset counter
B4B4 85 1E     52          sta COUNT
B4B6 F0 03     53          beq >2                    ; always taken
B4B8          54      ;
B4B8 20 22 B6  55      ^1      jsr BLDARGS1
B4BB          56      ;
B4BB 20 A7 B2  57      ^2      jsr CXSUB
B4BE          58      ;
B4BE E6 1E     59          inc COUNT
B4C0          60      ;

```

```

B4C0 A5 1E      61          lda COUNT
B4C2 C9 04      62          cmp #ARRAYSIZ
B4C4 D0 F2      63          bne <1
B4C6            64          ;
B4C6 60         65          rts
B4C7            66          ;
B4C7            67          ;
B4C7            68          ; Complex AYMUL routine. This routine multiplies two
B4C7            69          ; complex 2x2 floating point arrays and is written as:
B4C7            70          ;
B4C7            71          ; CALL "AYMUL",X(0,0,0),Y(0,0,0),Z(0,0,0)      X * Y form Z
B4C7            72          ;
B4C7            73          ; The real and imaginary coefficients are multiplied by
B4C7            74          ; processing the dot product of the rows of array X and
B4C7            75          ; the columns of array Y.
B4C7            76          ;
B4C7            77          ; Put Y into FAC, multiply by X using CXMUL, and save to Z.
B4C7            78          ;
B4C7 A9 00      79 AYMUL    lda #ZERO          ; initialize offset counter
B4C9 85 1E      80          sta COUNT
B4CB            81          ;
B4CB 20 52 B6    82 ^1      jsr BLDARGS2
B4CE 20 D1 B2    83          jsr CXMUL
B4D1            84          ;
B4D1 E6 1E      85          inc COUNT
B4D3            86          ;
B4D3 20 52 B6    87          jsr BLDARGS2
B4D6 20 D1 B2    88          jsr CXMUL
B4D9            89          ;
B4D9            90          ;
B4D9            91          ; Add TEMPR1 and TEMPR2.
B4D9            92          ;
B4D9 A9 93      93          lda #TEMPR2
B4DB A0 B0      94          ldy /TEMPR2
B4DD            95          ;
B4DD 20 F9 EA    96          jsr LOADFAC
B4E0            97          ;
B4E0 A9 8E      98          lda #TEMPR1
B4E2 A0 B0      99          ldy /TEMPR1
B4E4            100         ;
B4E4 20 BE E7    101         jsr FADD
B4E7            102         ;
B4E7            103         ;
B4E7            104         ; Save the computed value.
B4E7            105         ;
B4E7 A5 1E      106         lda COUNT
B4E9 4A         107         lsr
B4EA AA         108         tax
B4EB            109         ;
B4EB 18         110         clc
B4EC            111         ;
B4EC AD 7C B0    112         lda REAL3L
B4EF AC 7D B0    113         ldy REAL3H
B4F2            114         ;
B4F2 7D C6 B0    115         adc MULOUT,X
B4F5 90 01      116         bcc >2
B4F7            117         ;
B4F7 C8         118         iny
B4F8            119         ;
B4F8 AA         120 ^2      tax
B4F9            121         ;

```



```

B4F9 20 2B EB 122      jsr COPYFAC
B4FC          123      ;
B4FC          124      ;
B4FC          125      ; Add TEMPI1 and TEMPI2.
B4FC          126      ;
B4FC A9 9D      127      lda #TEMPI2
B4FE A0 B0      128      ldy /TEMPI2
B500          129      ;
B500 20 F9 EA   130      jsr LOADFAC
B503          131      ;
B503 A9 98      132      lda #TEMPI1
B505 A0 B0      133      ldy /TEMPI1
B507          134      ;
B507 20 BE E7   135      jsr FADD
B50A          136      ;
B50A          137      ;
B50A          138      ; Save the computed value.
B50A          139      ;
B50A A5 1E      140      lda COUNT
B50C 4A          141      lsr
B50D AA          142      tax
B50E          143      ;
B50E 18          144      clc
B50F          145      ;
B50F AD 82 B0   146      lda IMAG3L
B512 AC 83 B0   147      ldy IMAG3H
B515          148      ;
B515 7D C6 B0   149      adc MULOUT,X
B518 90 01      150      bcc >3
B51A          151      ;
B51A C8          152      iny
B51B          153      ;
B51B AA          154      ^3 tax
B51C          155      ;
B51C 20 2B EB   156      jsr COPYFAC
B51F          157      ;
B51F          158      ;
B51F          159      ; Multiply the next row and column.
B51F          160      ;
B51F E6 1E      161      inc COUNT
B521          162      ;
B521 A5 1E      163      lda COUNT
B523 C9 08      164      cmp #ARRAYSIZ*2
B525 D0 A4      165      bne <1
B527          166      ;
B527 60          167      rts
B528          168      ;
B528          169      ;
B528          170      ; Complex AYINV routine. This routine computes the inverse
B528          171      ; of a complex 2x2 floating point array and is written as:
B528          172      ;
B528          173      ; CALL "AYINV",X(0,0,0),Y(0),Z(0,0,0)          X forms Y and Z
B528          174      ;
B528          175      ; The determinant of X is computed and saved to Y. If the
B528          176      ; determinant is not zero, each element of X is divided
B528          177      ; by Y to form Z. Otherwise, all elements of Z are set to
B528          178      ; 7.922816252E+28.
B528          179      ;
B528          180      ;
B528          181      ; Let  $A = \begin{vmatrix} a & b \\ c & d \end{vmatrix}$ , then  $\det A = ad - bc$ , and  $A^{-1} * A = I$ 
B528          182      ;

```

```

B528      183 ;
B528      184 ;   -1      1      | d -b |
B528      185 ; A   = ----  | and I = | 1  0 |
B528      186 ;      detA  | -c  a |      | 0  1 |
B528      187 ;
B528      188 ;
B528      189 ; Compute the determinant detA = ad - bc.
B528      190 ;
B528 A9 00 191 AYINV      lda #ZERO                ; initialize offset counter
B52A 85 1E 192              sta COUNT
B52C      193 ;
B52C 20 AB B6 194              jsr BLDARGS3
B52F 20 D1 B2 195              jsr CXMUL
B532      196 ;
B532 E6 1E 197              inc COUNT
B534      198 ;
B534 20 AB B6 199              jsr BLDARGS3
B537 20 D1 B2 200              jsr CXMUL
B53A      201 ;
B53A      202 ;
B53A      203 ; Subtract TEMP2 from TEMP1, compute SGNFLG1, and save
B53A      204 ; the real coefficient to the determinant Y.
B53A      205 ;
B53A A9 93 206              lda #TEMPR2
B53C A0 B0 207              ldy /TEMPR2
B53E      208 ;
B53E 20 F9 EA 209              jsr LOADFAC
B541      210 ;
B541 A9 8E 211              lda #TEMPR1
B543 A0 B0 212              ldy /TEMPR1
B545      213 ;
B545 20 A7 E7 214              jsr FSUB
B548      215 ;
B548 A5 9D 216              lda FACEXP
B54A 29 7F 217              and #MSBCLR
B54C      218 ;
B54C 05 9E 219              ora FACMANT
B54E 8D 76 B0 220              sta SGNFLG1
B551      221 ;
B551 AE 7A B0 222              ldx FPREAL+2
B554 AC 7B B0 223              ldy FPREAL+3
B557      224 ;
B557 86 0A 225              stx REALPTRS+2
B559 84 0B 226              sty REALPTRS+3
B55B      227 ;
B55B 20 2B EB 228              jsr COPYFAC
B55E      229 ;
B55E      230 ;
B55E      231 ; Subtract TEMP2 from TEMP1, compute SGNFLG2, and save
B55E      232 ; the imaginary coefficient to the determinant Y.
B55E      233 ;
B55E A9 9D 234              lda #TEMPI2
B560 A0 B0 235              ldy /TEMPI2
B562      236 ;
B562 20 F9 EA 237              jsr LOADFAC
B565      238 ;
B565 A9 98 239              lda #TEMPI1
B567 A0 B0 240              ldy /TEMPI1
B569      241 ;
B569 20 A7 E7 242              jsr FSUB
B56C      243 ;

```

```

B56C A5 9D      244      lda FACEXP
B56E 29 7F      245      and #MSBCLR
B570            246      ;
B570 05 9E      247      ora FACMANT
B572 8D 77 B0    248      sta SGNFLG2
B575            249      ;
B575 AE 80 B0    250      ldx FPIMAG+2
B578 AC 81 B0    251      ldy FPIMAG+3
B57B            252      ;
B57B 86 1A      253      stx IMAGPTRS+2
B57D 84 1B      254      sty IMAGPTRS+3
B57F            255      ;
B57F 20 2B EB    256      jsr COPYFAC
B582            257      ;
B582            258      ;
B582            259      ; Determine if either the real or imaginary coefficients
B582            260      ; are zero, and if so, set the inverse matrix coefficients
B582            261      ; to HIGHVAL or 7.922816252E+28.
B582            262      ;
B582 AD 76 B0    263      lda SGNFLG1
B585 F0 05      264      beq INVZERO
B587            265      ;
B587 AD 77 B0    266      lda SGNFLG2
B58A D0 3C      267      bne NONZERO
B58C            268      ;
B58C            269      ;
B58C            270      ; One or both of the sign flags were found to be zero. Set
B58C            271      ; all coefficients of Z to HIGHVAL.
B58C            272      ;
B58C AD 7C B0    273      INVZERO lda REAL3L
B58F AC 7D B0    274      ldy REAL3H
B592            275      ;
B592 85 08      276      sta REALPTRS
B594 84 09      277      sty REALPTRS+1
B596            278      ;
B596 AD 82 B0    279      lda IMAG3L
B599 AC 83 B0    280      ldy IMAG3H
B59C            281      ;
B59C 85 18      282      sta IMAGPTRS
B59E 84 19      283      sty IMAGPTRS+1
B5A0            284      ;
B5A0 A2 04      285      ldx #ARRAYSIZ
B5A2 D0 14      286      bne >4                      ; always taken
B5A4            287      ;
B5A4 A5 08      288      ^1      lda REALPTRS
B5A6 69 05      289      adc #IMAGOFF
B5A8 90 02      290      bcc >2
B5AA            291      ;
B5AA E6 09      292      inc REALPTRS+1
B5AC            293      ;
B5AC 85 08      294      ^2      sta REALPTRS
B5AE            295      ;
B5AE A5 18      296      lda IMAGPTRS
B5B0 69 05      297      adc #IMAGOFF
B5B2 90 02      298      bcc >3
B5B4            299      ;
B5B4 E6 19      300      inc IMAGPTRS+1
B5B6            301      ;
B5B6 85 18      302      ^3      sta IMAGPTRS
B5B8            303      ;
B5B8 A0 04      304      ^4      ldy #FPLEN-1

```

```

B5BA          305 ;
B5BA B9 E7 B0 306 ^5      lda HIGHVAL,Y
B5BD          307 ;
B5BD 91 08    308      sta (REALPTRS),Y
B5BF 91 18    309      sta (IMAGPTRS),Y
B5C1          310 ;
B5C1 88       311      dey
B5C2 10 F6    312      bpl <5
B5C4          313 ;
B5C4 CA       314      dex
B5C5 D0 DD    315      bne <1
B5C7          316 ;
B5C7 60       317      rts
B5C8          318 ;
B5C8          319 ;
B5C8          320 ; Both sign flags were found to be not zero. Compute the
B5C8          321 ; inverse matrix coefficients of matrix X and save them
B5C8          322 ; to matrix Z. The determinant coefficients are still at
B5C8          323 ; REALPTRS2/3 and IMAGPTRS2/3.
B5C8          324 ;
B5C8 A9 00    325 NONZERO  lda #ZERO                ; initialize offset counter
B5CA 85 1E    326      sta COUNT
B5CC          327 ;
B5CC          328 ;
B5CC          329 ; Compute real inverse.
B5CC          330 ;
B5CC          331 ;
B5CC 20 04 B7 331 ^1      jsr BLDARGS4
B5CF 20 33 B3 332      jsr CXDIV
B5D2          333 ;
B5D2 A6 1E    334      ldx COUNT
B5D4          335 ;
B5D4 BD DA B0 336      lda INVLSB,X
B5D7 F0 2B    337      beq >2
B5D9          338 ;
B5D9          339 ;
B5D9          340 ; Multiply real coefficient by -1.
B5D9          341 ;
B5D9 A0 B0    342      ldy /INVLSB
B5DB          343 ;
B5DB 20 F9 EA 344      jsr LOADFAC
B5DE          345 ;
B5DE A5 0C    346      lda REALPTRS+4
B5E0 A4 0D    347      ldy REALPTRS+5
B5E2          348 ;
B5E2 20 7F E9 349      jsr FMULT
B5E5          350 ;
B5E5 A6 0C    351      ldx REALPTRS+4
B5E7 A4 0D    352      ldy REALPTRS+5
B5E9          353 ;
B5E9 20 2B EB 354      jsr COPYFAC
B5EC          355 ;
B5EC          356 ;
B5EC          357 ; Multiply imaginary coefficient by -1.
B5EC          358 ;
B5EC A6 1E    359      ldx COUNT
B5EE          360 ;
B5EE BD DA B0 361      lda INVLSB,X
B5F1 A0 B0    362      ldy /INVLSB
B5F3          363 ;
B5F3 20 F9 EA 364      jsr LOADFAC
B5F6          365 ;

```

```

B5F6 A5 1C      366      lda IMAGPTRS+4
B5F8 A4 1D      367      ldy IMAGPTRS+5
B5FA           368      ;
B5FA 20 7F E9   369      jsr FMULT
B5FD           370      ;
B5FD A6 1C      371      ldx IMAGPTRS+4
B5FF A4 1D      372      ldy IMAGPTRS+5
B601           373      ;
B601 20 2B EB   374      jsr COPYFAC
B604           375      ;
B604 E6 1E      376      ^2      inc COUNT
B606           377      ;
B606 A5 1E      378      lda COUNT
B608 C9 04      379      cmp #ARRAYSIZ
B60A D0 C0      380      bne <1
B60C           381      ;
B60C 60         382      rts
B60D           383      ;
B60D           384      ;
B60D           385      ; Complex CSMUL routine. This routine multiplies the
B60D           386      ; complex 2x2 floating point array with a complex floating
B60D           387      ; point scalar variable and is written as:
B60D           388      ;
B60D           389      ; CALL "CSMUL", X(0,0,0),Y(0),Z(0,0,0)          X * Y form Z
B60D           390      ;
B60D           391      ; The real and imaginary coefficients of each array element
B60D           392      ; are multiplied by the real and imaginary coefficients of
B60D           393      ; the scalar variable.
B60D           394      ;
B60D           395      ; Put Y into FAC, multiply by X using CXMUL, and save to Z.
B60D           396      ;
B60D A9 00      397      CSMUL      lda #ZERO          ; initialize offset counter
B60F 85 1E      398      sta COUNT
B611 F0 03      399      beq >2          ; always taken
B613           400      ;
B613 20 4B B7   401      ^1      jsr BLDARGS5
B616           402      ;
B616 20 D1 B2   403      ^2      jsr CXMUL
B619           404      ;
B619 E6 1E      405      inc COUNT
B61B           406      ;
B61B A5 1E      407      lda COUNT
B61D C9 04      408      cmp #ARRAYSIZ
B61F D0 F2      409      bne <1
B621           410      ;
B621 60         411      rts
B622           412      ;
B622           413      ;
B622           414      icl "CXFPMATH.5.L"

```

LLOAD CXFPMATH.5.L,A\$4000

```

B622          1          ttl "CXFPMATH.5 Routines, CXFPMATH.5.L"
B622          2          ;
B622          3          ;
B622          4          ; CXFPMATH.5.L
B622          5          ;
B622          6          ;
B622          7          ; Point to the next working real, imaginary, and output
B622          8          ; arguments.  BLDARDS1 is used by AYADD and AYSUB.
B622          9          ;
B622 A6 1E     10 BLDARGS1 ldx COUNT
B624          11          ;
B624 BD A2 B0  12          lda ADDSUB,X
B627 85 1F     13          sta OFFSET
B629          14          ;
B629 A2 00     15          ldx #ZERO
B62B          16          ;
B62B 18        17 ^1      clc
B62C          18          ;
B62C BD 78 B0  19          lda FPREAL+0,X
B62F BC 79 B0  20          ldy FPREAL+1,X
B632          21          ;
B632 65 1F     22          adc OFFSET
B634 90 01     23          bcc >2
B636          24          ;
B636 C8        25          iny
B637          26          ;
B637 95 08     27 ^2      sta REALPTRS+0,X
B639 94 09     28          sty REALPTRS+1,X
B63B          29          ;
B63B 18        30          clc
B63C          31          ;
B63C BD 7E B0  32          lda FPIMAG+0,X
B63F BC 7F B0  33          ldy FPIMAG+1,X
B642          34          ;
B642 65 1F     35          adc OFFSET
B644 90 01     36          bcc >3
B646          37          ;
B646 C8        38          iny
B647          39          ;
B647 95 18     40 ^3      sta IMAGPTRS+0,X
B649 94 19     41          sty IMAGPTRS+1,X
B64B          42          ;
B64B E8        43          inx
B64C E8        44          inx
B64D          45          ;
B64D E0 06     46          cpx #6
B64F 90 DA     47          bcc <1
B651          48          ;
B651 60        49          rts
B652          50          ;
B652          51          ;
B652          52          ; Point to the next working real, imaginary, and output
B652          53          ; arguments.  BLDARDS2 is used by AYMUL.
B652          54          ;
B652 A6 1E     55 BLDARGS2 ldx COUNT
B654          56          ;
B654          57          ;
B654          58          ; Setup real coefficients.
B654          59          ;
B654 18        60          clc

```

```

B655          61 ;
B655 AD 78 B0 62      lda FPREAL+0
B658 AC 79 B0 63      ldy FPREAL+1
B65B          64 ;
B65B 7D A6 B0 65      adc MULOFF1,X
B65E 90 01    66      bcc >1
B660          67 ;
B660 C8       68      iny
B661          69 ;
B661 85 08    70 ^1    sta REALPTRS+0
B663 84 09    71      sty REALPTRS+1
B665          72 ;
B665 18       73      clc
B666          74 ;
B666 AD 7A B0 75      lda FPREAL+2
B669 AC 7B B0 76      ldy FPREAL+3
B66C          77 ;
B66C 7D AE B0 78      adc MULOFF2,X
B66F 90 01    79      bcc >2
B671          80 ;
B671 C8       81      iny
B672          82 ;
B672 85 0A    83 ^2    sta REALPTRS+2
B674 84 0B    84      sty REALPTRS+3
B676          85 ;
B676 BD B6 B0 86      lda MULLSB1,X
B679 A0 B0    87      ldy /MULLSB1
B67B          88 ;
B67B 85 0C    89      sta REALPTRS+4
B67D 84 0D    90      sty REALPTRS+5
B67F          91 ;
B67F          92 ;
B67F          93 ; Setup up imaginary coefficients.
B67F          94 ;
B67F 18       95      clc
B680          96 ;
B680 AD 7E B0 97      lda FPIMAG+0
B683 AC 7F B0 98      ldy FPIMAG+1
B686          99 ;
B686 7D A6 B0 100     adc MULOFF1,X
B689 90 01    101     bcc >3
B68B          102 ;
B68B C8       103     iny
B68C          104 ;
B68C 85 18    105 ^3    sta IMAGPTRS+0
B68E 84 19    106     sty IMAGPTRS+1
B690          107 ;
B690 18       108     clc
B691          109 ;
B691 AD 80 B0 110     lda FPIMAG+2
B694 AC 81 B0 111     ldy FPIMAG+3
B697          112 ;
B697 7D AE B0 113     adc MULOFF2,X
B69A 90 01    114     bcc >4
B69C          115 ;
B69C C8       116     iny
B69D          117 ;
B69D 85 1A    118 ^4    sta IMAGPTRS+2
B69F 84 1B    119     sty IMAGPTRS+3
B6A1          120 ;
B6A1 BD BE B0 121     lda MULLSB2,X

```

```

B6A4 A0 B0      122      ldy /MULLSB2
B6A6            123      ;
B6A6 85 1C      124      sta IMAGPTRS+4
B6A8 84 1D      125      sty IMAGPTRS+5
B6AA            126      ;
B6AA 60          127      rts
B6AB            128      ;
B6AB            129      ;
B6AB            130      ; Point to the next working real, imaginary, and output
B6AB            131      ; arguments. BLDARDS3 is used by AYINV to compute a
B6AB            132      ; determinant.
B6AB            133      ;
B6AB A6 1E      134      BLDARGS3 ldx COUNT
B6AD            135      ;
B6AD            136      ;
B6AD            137      ; Setup real coefficients.
B6AD            138      ;
B6AD 18          139      clc
B6AE            140      ;
B6AE AD 78 B0    141      lda FPREAL+0
B6B1 AC 79 B0    142      ldy FPREAL+1
B6B4            143      ;
B6B4 7D CA B0    144      adc DETOFF1,X
B6B7 90 01       145      bcc >1
B6B9            146      ;
B6B9 C8          147      iny
B6BA            148      ;
B6BA 85 08       149      ^1 sta REALPTRS+0
B6BC 84 09       150      sty REALPTRS+1
B6BE            151      ;
B6BE 18          152      clc
B6BF            153      ;
B6BF AD 78 B0    154      lda FPREAL+0
B6C2 AC 79 B0    155      ldy FPREAL+1
B6C5            156      ;
B6C5 7D CC B0    157      adc DETOFF2,X
B6C8 90 01       158      bcc >2
B6CA            159      ;
B6CA C8          160      iny
B6CB            161      ;
B6CB 85 0A       162      ^2 sta REALPTRS+2
B6CD 84 0B       163      sty REALPTRS+3
B6CF            164      ;
B6CF BD CE B0    165      lda DETLSB1,X
B6D2 A0 B0       166      ldy /DETLSB1
B6D4            167      ;
B6D4 85 0C       168      sta REALPTRS+4
B6D6 84 0D       169      sty REALPTRS+5
B6D8            170      ;
B6D8            171      ;
B6D8            172      ; Setup up imaginary coefficients.
B6D8            173      ;
B6D8 18          174      clc
B6D9            175      ;
B6D9 AD 7E B0    176      lda FPIMAG+0
B6DC AC 7F B0    177      ldy FPIMAG+1
B6DF            178      ;
B6DF 7D CA B0    179      adc DETOFF1,X
B6E2 90 01       180      bcc >3
B6E4            181      ;
B6E4 C8          182      iny

```



```

B6E5          183 ;
B6E5 85 18    184 ^3      sta IMAGPTRS+0
B6E7 84 19    185      sty IMAGPTRS+1
B6E9          186 ;
B6E9 18       187      clc
B6EA          188 ;
B6EA AD 7E B0 189      lda FPIMAG+0
B6ED AC 7F B0 190      ldy FPIMAG+1
B6F0          191 ;
B6F0 7D CC B0 192      adc DETOFF2,X
B6F3 90 01    193      bcc >4
B6F5          194 ;
B6F5 C8       195      iny
B6F6          196 ;
B6F6 85 1A    197 ^4      sta IMAGPTRS+2
B6F8 84 1B    198      sty IMAGPTRS+3
B6FA          199 ;
B6FA BD D0 B0 200      lda DETLSB2,X
B6FD A0 B0    201      ldy /DETLSB2
B6FF          202 ;
B6FF 85 1C    203      sta IMAGPTRS+4
B701 84 1D    204      sty IMAGPTRS+5
B703          205 ;
B703 60       206      rts
B704          207 ;
B704          208 ;
B704          209 ; Point to the next working real, imaginary, and output
B704          210 ; arguments. BLDARDS4 is used by AYINV to compute an
B704          211 ; inverse matrix.
B704          212 ;
B704 A6 1E    213 BLDARGS4 ldx COUNT
B706          214 ;
B706          215 ;
B706          216 ; Setup real coefficients.
B706          217 ;
B706 18       218      clc
B707          219 ;
B707 AD 78 B0 220      lda FPREAL+0
B70A AC 79 B0 221      ldy FPREAL+1
B70D          222 ;
B70D 7D D2 B0 223      adc INVOFF1,X
B710 90 01    224      bcc >1
B712          225 ;
B712 C8       226      iny
B713          227 ;
B713 85 08    228 ^1      sta REALPTRS+0
B715 84 09    229      sty REALPTRS+1
B717          230 ;
B717 18       231      clc
B718          232 ;
B718 AD 7C B0 233      lda FPREAL+4
B71B AC 7D B0 234      ldy FPREAL+5
B71E          235 ;
B71E 7D D6 B0 236      adc INVOFF2,X
B721 90 01    237      bcc >2
B723          238 ;
B723 C8       239      iny
B724          240 ;
B724 85 0C    241 ^2      sta REALPTRS+4
B726 84 0D    242      sty REALPTRS+5
B728          243 ;

```

```

B728      244 ;
B728      245 ; Setup up imaginary coefficients.
B728      246 ;
B728 18    247 ;          clc
B729      248 ;
B729 AD 7E B0 249 ;          lda FPIMAG+0
B72C AC 7F B0 250 ;          ldy FPIMAG+1
B72F      251 ;
B72F 7D D2 B0 252 ;          adc INVOFF1,X
B732 90 01    253 ;          bcc >3
B734      254 ;
B734 C8      255 ;          iny
B735      256 ;
B735 85 18    257 ^3      sta IMAGPTRS+0
B737 84 19    258 ;          sty IMAGPTRS+1
B739      259 ;
B739 18      260 ;          clc
B73A      261 ;
B73A AD 82 B0 262 ;          lda FPIMAG+4
B73D AC 83 B0 263 ;          ldy FPIMAG+5
B740      264 ;
B740 7D D6 B0 265 ;          adc INVOFF2,X
B743 90 01    266 ;          bcc >4
B745      267 ;
B745 C8      268 ;          iny
B746      269 ;
B746 85 1C    270 ^4      sta IMAGPTRS+4
B748 84 1D    271 ;          sty IMAGPTRS+5
B74A      272 ;
B74A 60      273 ;          rts
B74B      274 ;
B74B      275 ;
B74B      276 ; Point to the next working real, imaginary, and output
B74B      277 ; arguments. BLDARDS5 is used by CSMUL to multiply a
B74B      278 ; complex array and a complex scalar.
B74B      279 ;
B74B A6 1E    280 BLDARGS5 ldx COUNT
B74D      281 ;
B74D      282 ;
B74D      283 ; Setup real coefficients.
B74D      284 ;
B74D 18      285 ;          clc
B74E      286 ;
B74E AD 78 B0 287 ;          lda FPREAL+0
B751 AC 79 B0 288 ;          ldy FPREAL+1
B754      289 ;
B754 7D DE B0 290 ;          adc CSOFF,X
B757 90 01    291 ;          bcc >1
B759      292 ;
B759 C8      293 ;          iny
B75A      294 ;
B75A 85 08    295 ^1      sta REALPTRS+0
B75C 84 09    296 ;          sty REALPTRS+1
B75E      297 ;
B75E 18      298 ;          clc
B75F      299 ;
B75F AD 7C B0 300 ;          lda FPREAL+4
B762 AC 7D B0 301 ;          ldy FPREAL+5
B765      302 ;
B765 7D DE B0 303 ;          adc CSOFF,X
B768 90 01    304 ;          bcc >2

```

```
B76A          305 ;
B76A C8       306      iny
B76B          307 ;
B76B 85 0C    308 ^2      sta REALPTRS+4
B76D 84 0D    309      sty REALPTRS+5
B76F          310 ;
B76F          311 ;
B76F          312 ; Setup up imaginary coefficients.
B76F          313 ;
B76F 18       314      clc
B770          315 ;
B770 AD 7E B0 316      lda FPIMAG+0
B773 AC 7F B0 317      ldy FPIMAG+1
B776          318 ;
B776 7D DE B0 319      adc CSOFF,X
B779 90 01    320      bcc >3
B77B          321 ;
B77B C8       322      iny
B77C          323 ;
B77C 85 18    324 ^3      sta IMAGPTRS+0
B77E 84 19    325      sty IMAGPTRS+1
B780          326 ;
B780 18       327      clc
B781          328 ;
B781 AD 82 B0 329      lda FPIMAG+4
B784 AC 83 B0 330      ldy FPIMAG+5
B787          331 ;
B787 7D DE B0 332      adc CSOFF,X
B78A 90 01    333      bcc >4
B78C          334 ;
B78C C8       335      iny
B78D          336 ;
B78D 85 1C    337 ^4      sta IMAGPTRS+4
B78F 84 1D    338      sty IMAGPTRS+5
B791          339 ;
B791 60       340      rts
B792          341 ;
B792          342 ;
```

BSAVE CXFPMATH,A\$0900,B,L\$0792

```
B792          343      usr CXFPMATH
B792          344 ;
B792          345 ;
B792          346      end 111
```

*** End of Assembly

Symbol List starts at 0x7800, ends at 0x7DBE, used 0x05BE, remaining 0x392A

Symbols unsorted:

REALPTRS	0008	IMAGPTRS	0018	COUNT	001E	OFFSET	001F	FRETOP	006F
HIMEM	0073	ARGGUARD	0092	FACEXP	009D	FACMANT	009E	FACSIGN	00A2
ARGSIGN	00AA	XORSIGN	00AB	FACGUARD	00AC	CHRGET	00B1	CHRGOT	00B7
ZERO	0000	MSBCLR	007F	MSBSET	0080	NEGONE	00FF	ARRAYSIZ	0004
VARS2	0004	IMAGOFF	0005	FPLEN	0005	CMDLEN	0005	ARGLEN	0006
VARS3	0006	CXFPOFF	000A	RETURN	008D	ESCAPE	009B	SPACE	00A0
QUOTE	00A2	COMMA	00AC	PAGESIZE	0100	DOSWARM	03D0	KEY	C000
CLRKEY	C010	ROM2WP	C082	PTRGET	DFE3	FSUB	E7A7	FADD	E7BE
FMULT	E97F	LOADARG	E9E3	FDIV	EA66	LOADFAC	EAF9	COPYFAC	EB2B
MOVFA	EB53	FSGN	EB90	FSQR	EE8D	FEXP	EE97	FCOS	EFEA
FSIN	EFF1	FTAN	F03A	FATAN	F09E	CROUT	FD8E	PRBYTE	FDDA
PRHEX	FDE3	COUT	FDED	INITBL	B000	JMPTBL	B009	HASHTBL	B040
HASHSIZ	000F	PTRLNTBL	B04F	FIRSTIME	B070	HASHVAL	B071	CMDNDX	B072
PTRNUM	B073	PTRNDX	B074	PTRLEN	B075	SGNFLG1	B076	SGNFLG2	B077
FPREAL	B078	REAL1L	B078	REAL1H	B079	REAL2L	B07A	REAL2H	B07B
REAL3L	B07C	REAL3H	B07D	FPIMAG	B07E	IMAG1L	B07E	IMAG1H	B07F
IMAG2L	B080	IMAG2H	B081	IMAG3L	B082	IMAG3H	B083	TEMP1	B084
TEMP2	B089	TEMPR1	B08E	TEMPR2	B093	TEMPI1	B098	TEMPI2	B09D
ADDSUB	B0A2	MULOFF1	B0A6	MULOFF2	B0AE	MULLSB1	B0B6	MULLSB2	B0BE
MULOUT	B0C6	DETOFF1	B0CA	DETOFF2	B0CC	DETLSB1	B0CE	DETLSB2	B0D0
INVOFF1	B0D2	INVOFF2	B0D6	INVLSB	B0DA	CSOFF	B0DE	MINUSONE	B0E2
HIGHVAL	B0E7	PIDIV2	B0EC	PI	B0F1	VERSION	B0FE	LOADER	B100
TEMPMOD	B134	LIBENT	B137	LINKER	B17A	PRNTERR	B1C5	ERRS	B1D3
ERR1	B1D3	ERR2	B1F9	ERR3	B212	ERR4	B230	LGNOT	B24D
LGAND	B25D	LGORA	B261	LGEOR	B265	LGCALC	B267	LGMOD1	B271
LGMOD2	B278	CXADD	B27D	CXSUB	B2A7	CXMUL	B2D1	CXDIV	B333
CX2PL	B3F0	C2PEXIT	B466	PL2CX	B46D	AYADD	B49D	AYSUB	B4B2
AYMUL	B4C7	AYINV	B528	INVZERO	B58C	NONZERO	B5C8	CSMUL	B60D
BLDARGS1	B622	BLDARGS2	B652	BLDARGS3	B6AB	BLDARGS4	B704	BLDARGS5	B74B

Symbols alphabetically sorted:

ADDSUB	B0A2	ARGGUARD	0092	ARGLEN	0006	ARGSIGN	00AA	ARRAYSIZ	0004
AYADD	B49D	AYINV	B528	AYMUL	B4C7	AYSUB	B4B2	BLDARGS1	B622
BLDARGS2	B652	BLDARGS3	B6AB	BLDARGS4	B704	BLDARGS5	B74B	C2PEXIT	B466
CHRGET	00B1	CHRGOT	00B7	CLRKEY	C010	CMDLEN	0005	CMDNDX	B072
COMMA	00AC	COPYFAC	EB2B	COUNT	001E	COUT	FDED	CROUT	FD8E
CSMUL	B60D	CSOFF	B0DE	CX2PL	B3F0	CXADD	B27D	CXDIV	B333
CXFPOFF	000A	CXMUL	B2D1	CXSUB	B2A7	DETLSB1	B0CE	DETLSB2	B0D0
DETOFF1	B0CA	DETOFF2	B0CC	DOSWARM	03D0	ERR1	B1D3	ERR2	B1F9
ERR3	B212	ERR4	B230	ERRS	B1D3	ESCAPE	009B	FACEXP	009D
FACGUARD	00AC	FACMANT	009E	FACSIGN	00A2	FADD	E7BE	FATAN	F09E
FCOS	EFEA	FDIV	EA66	FEXP	EE97	FIRSTIME	B070	FMULT	E97F
FPIMAG	B07E	FPLEN	0005	FPREAL	B078	FRETOP	006F	FSGN	EB90
FSIN	EFF1	FSQR	EE8D	FSUB	E7A7	FTAN	F03A	HASHSIZ	000F
HASHTBL	B040	HASHVAL	B071	HIGHVAL	B0E7	HIMEM	0073	IMAG1H	B07F
IMAG1L	B07E	IMAG2H	B081	IMAG2L	B080	IMAG3H	B083	IMAG3L	B082
IMAGOFF	0005	IMAGPTRS	0018	INITBL	B000	INVLSB	B0DA	INVOFF1	B0D2
INVOFF2	B0D6	INVZERO	B58C	JMPTBL	B009	KEY	C000	LGAND	B25D
LGCALC	B267	LGEOR	B265	LGMOD1	B271	LGMOD2	B278	LGNOT	B24D
LGORA	B261	LIBENT	B137	LINKER	B17A	LOADARG	E9E3	LOADER	B100
LOADFAC	EAF9	MINUSONE	B0E2	MOVFA	EB53	MSBCLR	007F	MSBSET	0080
MULLSB1	B0B6	MULLSB2	B0BE	MULOFF1	B0A6	MULOFF2	B0AE	MULOUT	B0C6
NEGONE	00FF	NONZERO	B5C8	OFFSET	001F	PAGESIZE	0100	PI	B0F1

PIDIV2	B0EC	PL2CX	B46D	PRBYTE	FDDA	PRHEX	FDE3	PRNTERR	B1C5
PTRGET	DFE3	PTRLEN	B075	PTRLNTBL	B04F	PTRNDX	B074	PTRNUM	B073
QUOTE	00A2	REAL1H	B079	REAL1L	B078	REAL2H	B07B	REAL2L	B07A
REAL3H	B07D	REAL3L	B07C	REALPTRS	0008	RETURN	008D	ROM2WP	C082
SGNFLG1	B076	SGNFLG2	B077	SPACE	00A0	TEMP1	B084	TEMP2	B089
TEMPI1	B098	TEMPI2	B09D	TEMPMOD	B134	TEMPR1	B08E	TEMPR2	B093
VARS2	0004	VARS3	0006	VERSION	B0FE	XORSIGN	00AB	ZERO	0000

Symbols numerically sorted:

ZERO	0000	VARS2	0004	ARRAYSIZ	0004	IMAGOFF	0005	FPLEN	0005
CMDLEN	0005	VARS3	0006	ARGLEN	0006	REALPTRS	0008	CXFPOFF	000A
HASHSIZ	000F	IMAGPTRS	0018	COUNT	001E	OFFSET	001F	FRETOP	006F
HIMEM	0073	MSBCLR	007F	MSBSET	0080	RETURN	008D	ARGGUARD	0092
ESCAPE	009B	FACEXP	009D	FACMANT	009E	SPACE	00A0	QUOTE	00A2
FACSIGN	00A2	ARGSIGN	00AA	XORSIGN	00AB	FACGUARD	00AC	COMMA	00AC
CHRGET	00B1	CHRGOT	00B7	NEGONE	00FF	PAGESIZE	0100	DOSWARM	03D0
INITBL	B000	JMPTBL	B009	HASHTBL	B040	PTRLNTBL	B04F	FIRSTIME	B070
HASHVAL	B071	CMDNDX	B072	PTRNUM	B073	PTRNDX	B074	PTRLEN	B075
SGNFLG1	B076	SGNFLG2	B077	REAL1L	B078	FPREAL	B078	REAL1H	B079
REAL2L	B07A	REAL2H	B07B	REAL3L	B07C	REAL3H	B07D	IMAG1L	B07E
FPIMAG	B07E	IMAG1H	B07F	IMAG2L	B080	IMAG2H	B081	IMAG3L	B082
IMAG3H	B083	TEMP1	B084	TEMP2	B089	TEMPR1	B08E	TEMPR2	B093
TEMPI1	B098	TEMPI2	B09D	ADDSUB	B0A2	MULOFF1	B0A6	MULOFF2	B0AE
MULLSB1	B0B6	MULLSB2	B0BE	MULOUT	B0C6	DETOFF1	B0CA	DETOFF2	B0CC
DETLSB1	B0CE	DETLSB2	B0D0	INVOFF1	B0D2	INVOFF2	B0D6	INVLSB	B0DA
CSOFF	B0DE	MINUSONE	B0E2	HIGHVAL	B0E7	PIDIV2	B0EC	PI	B0F1
VERSION	B0FE	LOADER	B100	TEMPMOD	B134	LIBENT	B137	LINKER	B17A
PRNTERR	B1C5	ERRS	B1D3	ERR1	B1D3	ERR2	B1F9	ERR3	B212
ERR4	B230	LGNOT	B24D	LGAND	B25D	LGORA	B261	LGEOR	B265
LGCALC	B267	LGMOD1	B271	LGMOD2	B278	CXADD	B27D	CXSUB	B2A7
CXMUL	B2D1	CXDIV	B333	CX2PL	B3F0	C2PEXIT	B466	PL2CX	B46D
AYADD	B49D	AYSUB	B4B2	AYMUL	B4C7	AYINV	B528	INVZERO	B58C
NONZERO	B5C8	CSMUL	B60D	BLDARGS1	B622	BLDARGS2	B652	BLDARGS3	B6AB
BLDARGS4	B704	BLDARGS5	B74B	KEY	C000	CLRKEY	C010	ROM2WP	C082
PTRGET	DFE3	FSUB	E7A7	FADD	E7BE	FMULT	E97F	LOADARG	E9E3
FDIV	EA66	LOADFAC	EAF9	COPYFAC	EB2B	MOVFA	EB53	FSGN	EB90
FSQR	EE8D	FEXP	EE97	FCOS	EFEA	FSIN	EFF1	FTAN	F03A
FATAN	F09E	CROUT	FD8E	PRBYTE	FDDA	PRHEX	FDE3	COUT	FDED