92K0506-007 A

```
      *       FIXED POINT MATH       16 BIT, MUL/DIV
      *
      *  COS                    SINGLE PRECISION FIXED POINT COSINE
      *
      * COS(X) FOR  -PI  <= 4*X <= +PI
      *THIS ROUTINE CALLS ON SUBROUTINE 'POLY'
      *
      XCOS  ,ENTR   ,
            ,JAN    ,*+4            COS(X) FOR X IN -PI TO +PI
            ,CPA    ,               IS EVALUATED AS
            ,IAR    ,               SIN(PI/2-ABS(X), WHICH
            ,ADDI   ,031104         REDUCES THE VARIABLE
            ,ASLA   ,1              TO RANGE -PI/2 TO +PI/2 ...
            ,CALL   ,POLY
      1     ,DATA   ,1,027,-0650,010421,-052525
            ,DATA   ,0,077777,0
      2     ,JMP*   ,XCOS           COS(A) IS SCALED *1/2...
            ,MORE   ,
            ,END    ,
```

```
*       FIXED PØINT MATH        16 BIT, MUL/DIV
*
*  EXN                      SINGLE PRECISIØN FIXED PØINT
*
* EXP(-X) , GIVEN 0 <= X < 1
*THIS RØUTINE CALLS ØN SUBRØUTINE 'PØLY'
*
        ,JAP    ,XEXN           ERRØR EXIT ... FØR /<=X<1
        ,INR    ,XEXN           INPUT AND ØUTPUT
        ,INR    ,XEXN           ARE SCALED *1...
        ,CALL   ,PØLY
1       ,DATA   ,0,-0250,02356,-012447,037770
        ,DATA   ,0,-077777,077777
2       ,JMP*   ,XEXN
XEXN    ,ENTR   ,               EXP (-X) * 1
        ,JMP    ,XEXN-16
        ,MØRE   ,
        ,END    ,
```

```
      *       FIXED PØINT MATH      16 BIT, MUL/DIV
      *
      *  XATN                    SINGLE PRECISIØN FIXED PØINT ARCTANGENT
      *
      XATN  ,ENTR   ,                 ARCTAN(A) FØR -16A6+1..,
            ,CALL   ,PØLY             INPUT AND ØUTPUT ARE SCALED *1
      1     ,DATA   ,1,0272,-01725,04633,-010226,014500
            ,DATA   ,-025242,0,077777,0
      2     ,JMP*   ,XATN
            ,MØRE   ,
            ,END    ,
```

```
*       FIXED PØINT MATH       16 BIT, MUL/DIV
*
*  XLØG                    SINGLE PRECISIØN FIXED PØINT LØGARITHM
*
* LØG(1+X) , GIVEN -1 < X < +1
*
        ,JAN*    ,XLØG
        ,INR     ,XLØG
        ,INR     ,XLØG
        ,CALL    ,PØLY
1       ,DATA    ,0,0532,-03406,010512,-016407,025026
        ,DATA    ,-037763,0,077777,0
2       ,JMP*    ,XLØG
XLØG    ,ENTR    ,
        ,JMP     ,XLØG-18
        ,MØRE    ,
        ,END     ,
```

```
*       FIXED PØINT MATH      16 BIT, MUL/DIV
*
*  PØLY                       SINGLE PRECISIØN FIXED PØINT
*                             PØLYNØMIAL EVALUATØR
*
*THIS UTILITY RØUTINE IS DESIGNED SPECIFICALLY FØR THE
*GRØUP ØF ELEMENTARY FUNCTIØN... IT MAY BE USED,WITH
*SØME CAUTIØN, FØR GENERAL PØLYNØMIAL PRØBLEMS
*
PØL1   ,STX     ,PØLY-4
       ,STA     ,PØLY-3
       ,STA     ,PØLY-2
       ,LDX     ,PØLY
       ,LDA     ,0,1
       ,JAZ     ,PØLL
       ,LDB     ,PØLY-3
       ,LDAI    ,040000
       ,MUL     ,PØLY-3
       ,STA     ,PØLY-2
       ,TZA     ,
PØLL   ,STA     ,PØLY
       ,IXR     ,
       ,LDA     ,0,1
       ,JAZ     ,PØL2
       ,ADD     ,PØLY
       ,TAB     ,
       ,TZA     ,
       ,MUL     ,PØLY-2
       ,JMP     ,PØLL
PØL2   ,IXR     ,
       ,LDA     ,0,1
       ,JAZ     ,PØL3
       ,ADD     ,PØLY
       ,TAB     ,
       ,TZA     ,
       ,MUL     ,PØLY-3
       ,IXR     ,
PØL3   ,XAZ     ,PØLY-1
       ,ADD     ,0,1
       ,IXR     ,
       ,STX     ,PØLY
       ,LDX     ,*+3
```

```
        ,JMP*    ,PØLY
        ,BSS     ,3
        ,DATA    ,014012
  PØLY  ,ENTR    ,
        ,JMP     ,PØL1
        ,MØRE    ,
        ,END     ,
```

```
*       FIXED PØINT MATH        16 BIT, MUL/DIV
*
*  XSIN                    SINGLE PRECISIØN FIXED PØINT SINE
*
* SIN(X) FØR X BETWEEN -PI/4 AND +PI/4
*CALLS SUBRØUTINE 'PØLY'. X IN AR SCALED *1/4
*
XSIN  ,ENTR   ,
      ,JAP    ,*+10
      ,ADD    ,*+27
      ,JAP    ,*+4            REDUCE A TØ BETW. + ØR - PI/2
      ,CPA    ,               ... CALL IT Y
      ,IAR    ,               IF NEG. USE Y=ABS(A+PI/2)-PI/2
      ,SUB    ,*+22
      ,JMP    ,*+8
      ,SUB    ,*+19
      ,JAN    ,*+4
      ,CPA    ,
      ,IAR    ,               IF A +VE USE
      ,ADD    ,*+14           Y=ABS(A-PI/2)+PI/2
      ,ASLA   ,1
      ,CALL   ,PØLY
      ,DATA   ,1,027,-0650,010421,-052525
      ,DATA   ,0,077777,0
2     ,JMP*   ,XSIN
      ,DATA   ,031104
      ,MØRE   ,
      ,END    ,
```

```
*       FIXED PØINT MATH        16 BIT, MUL/DIV
*
*  XEXP                         SINGLE PRECISIØN FIXED PØINT
*                               PØSITIVE EXPØNENTIAL
*
* EXP(+X) X BETO,1
*
BEXP   ,JAN*     ,XEXP
       ,INR      ,XEXP
       ,INR      ,XEXP
       ,CPA      ,
       ,IAR      ,
       ,ADDI     ,077777
       ,CALL     ,XEXN
       ,JMP      ,*+2
       ,TAB      ,
       ,MUL      ,XEXP+3
       ,JMP      ,XEXP
XEXP   ,BES      ,0
       ,JMP      ,BEXP
       ,DATA     ,053374
       ,MØRE     ,
       ,END      ,
```

```
*       FIXED PØINT MATH      16 BIT, MUL/DIV
*
*  XSQT                    SINGLE PRECISIØN FIXED PØINT SQUARE RØØT
*
*    SQUARE RØØT ØF X FØR 0 <= X < 1
*
BSQT   ,INR     ,XSQT
       ,INR     ,XSQT
       ,RØF     ,
       ,JAZ*    ,XSQT
       ,STB     ,XSQT+11
       ,ERA     ,XSQT+12
       ,JAZ     ,*+4
       ,JMP     ,*+5
       ,ERA     ,XSQT+12
       ,JMP*    ,XSQT
       ,ERA     ,XSQT+12
       ,STX     ,XSQT+7
       ,ZERØ    ,06
       ,IXR     ,
       ,LLRL    ,2
       ,JIF     ,022,*-2
       ,DXR     ,
       ,LLRL    ,14
       ,STB     ,XSQT+6
       ,STX     ,XSQT+10
       ,LDA     ,XSQT+12
       ,STA     ,XSQT+8
SQT1   ,LDA     ,XSQT+6
       ,DIV     ,XSQT+8
       ,TBA     ,
       ,SUB     ,XSQT+8
       ,STA     ,XSQT+9
       ,ASRA    ,1
       ,ADD     ,XSQT+8
       ,STA     ,XSQT+8
       ,LDA     ,XSQT+9
       ,ADDI    ,0377
       ,JAN     ,SQT1
       ,LDX     ,XSQT+7
       ,LDR     ,XSQT+10
       ,LDA     ,XSQT+8
```

```
          ,JBZ     ,XSQT-3
          ,ASRA    ,1
          ,DBR     ,
          ,JMP     ,*-4
          ,LDX     ,XSQT+7
          ,LDB     ,XSQT+11
          ,JMP     ,XSQT
   XSQT   ,BES     ,0
          ,STA     ,*+5
          ,JAN*    ,XSQT
          ,JMP     ,BSQT
          ,BSS     ,6
          ,DATA    ,077777
          ,MØRE    ,
          ,END     ,
```

```
*       FIXED PØINT MATH        16 BIT, MUL/DIV
*
*  XMUL                    STANDARD SØFTWARE MULTIPLY
*
*                 A,B>[B*PAR]<A
*                 X IS UNCHANGED 40 WØRDS
*
ADD     ,DATA     ,0124025        ADD MCND
ERA     ,DATA     ,0134023        ERA SIGN
SØF     ,DATA     ,0124013        ADD SIGN
SUB     ,DATA     ,0144007        SUB CND
BGN     ,RØF      ,               RESET ØF
        ,STX      ,XMXR           SAVE XR
        ,LDX      ,XMUL           GET ADDRASS ØF CALL SEQ
        ,LDX      ,0,1            GET ADDR ØF MCND
        ,LDX      ,0,1            GET MCND
        ,STX      ,MCND           AND SAVE
        ,LDX      ,K15            SET BIT CØUNT
RPT     ,LLRL     ,31             A SIGN> LSB ØF MPLR
        ,ADD      ,XSIG           SET ØF IF LSB>1
        ,LLRL     ,1              ALIGN PARTIAL PRØDUCT
        ,XØF      ,ADD            ADD MCND IF LSB>1
        ,LASR     ,1              AND SHIFT RIGHT
        ,XØF      ,ERA            INVERT SIGN IF ØF
        ,DXR      ,               CØUNT BITS DDEVELØPED
        ,JXZ      ,*+4            JMP IF DØNE
        ,JMP      ,RPT            ELSE REPEAT
        ,LLRL     ,NBIT           A SIGN>MPLR SIGN
        ,XAN      ,SØF            SET ØF IF NEG MPLR
        ,LLRL     ,NBIT           RESET PRØDUCT
        ,XØF      ,SUB            SUB MCND IF NEG MPLR
        ,INR      ,XMUL           SET RETURN
        ,LDX      ,XMXR           RESTØRE XR
        ,JMP*     ,XMUL           A,B>B*M<A
XMUL    ,RES      ,0              ENTRY
        ,JMP      ,BGN
XMXR    ,BSS      ,1              TEMPØRARY STØRAGE
MCND    ,BSS      ,1
XSIG    ,DATA     ,0100000        CØNSTANTS
K15     ,DATA     ,15
        ,MØRE     ,
        ,END      ,
```

```
*       FIXED PØINT MATH        16 BIT, MUL/DIV
*
*  XBTD                    FIXED PØINT INTEGER BIN TØ DEC CØNVERSIØN
*
XBTD   ,ENTR    ,
       ,STA     ,B
       ,STX     ,B+1
       ,JAP     ,*+4            JUMP IF PØSITIVE
       ,CPA     ,               ELSE CØMPLEMTNT
       ,IAR     ,               AND ADD ØNE
       ,TAB     ,
       ,LDXI    ,3              INITIALIZE CØUNT
       ,TZA     ,
1      ,DIV     ,BB
       ,STB     ,*+16           SAVE BIN VALUE
       ,LDB     ,*+16           GET PREVIØUS DIGITS
       ,LLSR    ,4              ATTACH DIGIT TØ RESULT
       ,JXZ     ,*+7            JUMP IF CØMPLETE
       ,DXR     ,               ELSE CØUNT DIGITS
       ,STB     ,*+11           SAVE DIGITS ASSEMBLED
       ,LDB     ,*+9            GET BIN VALUE
       ,JMP     ,*-9
       ,LDA     ,*+4            RESTØRE AR
       ,LDX     ,*+4            RESTØRE XR
       ,JMP*    ,XBTD           RETURN
B      ,DATA    ,0,0,0,0        TEMP STØRAGE
BB     ,DATA    ,10             CØNSTANT
       ,MØRE    ,
       ,END     ,
```

```
*       FIXED PØINT MATH        16 BIT, MUL/DIV
*
*  XDTB                     FIXED PØINT INTEGER DEC TØ BIN CØNVERSIØN
*
XDTB   ,ENTR    ,
       ,STA     ,AA
       ,STX     ,AA+1
       ,TAB     ,
       ,LDXI    ,3              INITIALIZE CØUNT
       ,TZA     ,
       ,STA     ,AA+2
       ,LLRL    ,4              GET NEXT DIGIT
       ,STB     ,AA+3
       ,LDB     ,AA+2
1      ,MUL     ,B10
       ,JXZ     ,*+7            JUMP IF CØMPLETE
       ,DXR     ,               ELSE CØUNT DIGITS
       ,STB     ,*+11           SAVE PARTIAL PRØDUCT
       ,LDB     ,*+11           GET REMAINING DIGITS
       ,JMP     ,*-9
       ,LDA     ,*+5            RESTØRE AR
       ,LDX     ,*+5            RESTØRE XR
       ,JMP*    ,XDTB           RETURN
B10    ,DATA    ,10             CØNSTANT
AA     ,DATA    ,0,0,0,0        TEMP STØRAGE
       ,MØRE    ,
       ,END     ,
```

```
*       FIXED PØINT MATH        16 BIT, MUL/DIV
*
*  XDIV                     SØFTWARE DIVIDE
*
*                 A,B/MB [QUØTIENT] < A [REMINDER]
*                 A REG MEMØRY X IS UNCHANGED
*                 QUØTIENT IS ALWAYS TRUE
*                 REMAINDER IS SIGN ØF DIVIDEND [UNLESS R>0]
*
TØP     ,STX      ,XR             SAVE XR
        ,DECR     ,4              SET SIGN INDICATØR
        ,JAP      ,PØSU           SET DIVIDEND PØS
        ,CPX      ,               SET DSIN>NEG
        ,CPB      ,               LØ ØRDER TWØ,S CØMPL
        ,IBR      ,
        ,LRLB     ,1              SIGN>0
        ,LSRB     ,1
        ,CPA      ,               HI ØRDER TWØ,S CØMPL
        ,JBZ      ,*+4
        ,JMP      ,*+3
        ,IAR      ,
PØSU    ,STX      ,DSIN           SAVE DIVDN SIGN
        ,STA      ,DVDN           SAVE DIVDN
        ,LDX      ,XDIV           GET ADDR ØF CALL SEQ
        ,LDX      ,0,1            GET ADDR ØF PARAM
        ,LDA      ,0,1            GET DIVISØR
        ,LDX      ,DSIN           GET DIVDN SIGN
        ,JAP      ,*+5            SET DIVISØR PØS
        ,CPX      ,               SET QUØTIENT SIGN
        ,CPA      ,               TWØ,S CØMPL
        ,IAR      ,
        ,STA      ,DVSR           SAVE DIVISØR
        ,LDA      ,DVDN           GET DIVDN
        ,STX      ,QSIN           SAVE QUØT SIGN
        ,LDX      ,K14            SET CYCLE CØUNT
        ,LRLB     ,1              ADJUST LØ ØRDER [DELETE SIGN]
        ,SUB      ,DVSR           SUB DIVSØR
        ,SØF      ,
        ,JAP      ,XERR           JMP IF ØVERFLØW ERRØR
        ,RØF      ,
NEGU    ,LLRL     ,1              DEVELØP 14 QUØTIENT BITS
        ,ADD      ,DVSR           [NØN RESTØRING ALGØRITHM]
```

```
TEST  ,JXZ    ,ADJ          JMP IF CØMPLETE
      ,DXR    ,             CØUNT BITS
      ,JAN    ,NEGU         JUMP IF NEG REMAINDER
      ,LLRL   ,1            SHIFT QUØTOREM
      ,SUB    ,DVSR         SUBTRACT DIVSR
      ,JMP    ,TEST         GØ TEST
ADJ   ,LRLB   ,1            GET LAST QUØTIENT BIT
      ,JAP    ,*+4          JMP IF ØR
      ,IBR    ,             ELSE SET LØB
      ,ADD    ,DVSR         RESTØRE REMAINDER
      ,LDX    ,QSIN         GET TRUE QUØTIENT
      ,JXZ    ,*+4          JMP IF NEGATIVE QUØT
      ,CPB    ,             ELSE SET PØSITIVE
      ,DBR    ,
      ,IBR    ,
      ,LDX    ,DSIN         GET TRUE REMAINDER
      ,JXZ    ,*+4          JMP IF REMAINDER NEG
      ,JMP    ,*+4          ELSE LEAVE PØS
      ,CPA    ,
      ,IAR    ,
XERR  ,INR    ,XDIV         SET RETURN
      ,LDX    ,XR
      ,JMP*   ,XDIV         A,BØM← B>QUØT A>REM
XDIV  ,BES    ,0            ENTRY
      ,JMP    ,TØP
K14   ,DATA   ,14
XR    ,BSS    ,1            TEMP STØRAGE
DVSR  ,BSS    ,1
DVDN  ,BSS    ,1
DSIN  ,BSS    ,1
QSIN  ,BSS    ,1
      ,MØRE   ,
      ,END    ,
```

```
*       FIXED PØINT MATH        16 BIT, MUL/DIV
*
*   XDAD                    FIXED PØINT DØUBLE PRECISIØN ADD/SUBTRACT
*
        ,STX    ,XDAD+3         SAVE XR
        ,RØF    ,               RESET ØF
        ,LDX    ,XDAD
        ,LDX    ,0,1            XR>ADDR ØF HI B
        ,STA    ,XDAD+4         SAVE HI A
        ,TBA    ,               GET LØ A
        ,ADD    ,1,1            ADD LØ B
        ,ANAI   ,077777         MASK SIGN
        ,TAB    ,               SAVE RESULT
        ,TZA    ,
        ,AØFA   ,               GET CARRY
        ,RØF    ,               RESET ØF
        ,ADD    ,XDAD+4         ADD HI A
        ,ADD    ,0,1            ADD HI B
        ,INR    ,XDAD           SET RETURN
        ,LDX    ,XDAD+3         RESTØRE XR
        ,JMP    ,0              RETURN
XDAD    ,EQU    ,*-1            ENTRY
        ,JMP    ,*-19
        ,DATA   ,0,0            TEMP STØRAGE
        ,MØRE   ,
        ,END    ,
```

```
*       FIXED PØINT MATH      16 BIT, MUL/DIV
*
*  XDCØ                    FIXED PØINT DØUBLE PRECISIØN CØMPLEMENT
*
XDCØ   ,ENTR    ,
       ,CPA     ,
       ,JBZ     ,*+8
       ,CPB     ,
       ,IBR     ,
       ,LRLB    ,1
       ,LSRB    ,1
       ,JMP*    ,XDCØ
       ,IAR     ,
       ,JMP*    ,XDCØ
       ,MØRE    ,
       ,END     ,
```

```
*       FIXED PØINT MATH        16 BIT, MUL/DIV
*
*  XDMU                    FIXED PØINT DØUBLE PRECISIØN MULTIPLY
*
BXDM   ,STX     ,XDMU+4         SAVE XR
       ,RØF     ,               RESET ØF
       ,CØMP    ,4              INITIALIZE PRØDUCT SIGN
       ,JAP     ,*+7            JUMP IF A PØS
       ,IXR     ,               ELSE SET PRØD SIGN
       ,CALL    ,XDCØ           GET ABS VALUE
       ,JAN     ,XDMU-3         JUMP IF 1.000...
       ,STA     ,XDMU+6         SAVE HI A
       ,STB     ,XDMU+5         SAVE LØ A
       ,STX     ,XDMU+7         SAVE PRØD SIGN
       ,LDX     ,XDMU           GET ADDR ØF HI B
       ,LDX     ,0,1
       ,LDA     ,0,1            GET HI B
       ,LDB     ,1,1            GET LØ B
       ,LDX     ,XDMU+7         GET PRØD SIGN
       ,JAP     ,*+7            JUMP IF B PØS
       ,IXR     ,               ELSE SET PRØD SIGN
       ,CALL    ,XDCØ           GET ABS VALUE
       ,JAN     ,XDMU-3         JUMP IF 1.000...
       ,STA     ,XDMU+7         SAVE HI B
       ,LDA     ,XDMU+3         SET TØ RØUND
       ,MUL     ,XDMU+6
       ,LDB     ,XDMU+7
       ,MUL     ,XDMU+6
       ,STB     ,XDMU+6
       ,LDB     ,XDMU+5
       ,STA     ,XDMU+5
       ,LDA     ,XDMU+3
       ,MUL     ,XDMU+7
       ,TAB     ,               SET TØ ADD
       ,TZA     ,               TØ PARTIAL PRØD
       ,CALL    ,XDAD           ADD PARTIAL PRØD
       ,PZE     ,XDMU+5
       ,JXZM    ,XDCØ           CØMPLEMENT IF PRØD NEG
       ,INR     ,XDMU           SET RETURN
       ,LDX     ,XDMU+4         RESTØRE XR
       ,JMP     ,0
       ,ØRG     ,*-1
```

```
XDMU  ,ENTR   ,
      ,JMP    ,RXDM
      ,DATA   ,040000
      ,DATA   ,0,0,0,0
      ,MØRE   ,
      ,END    ,
```

```
*       FIXED PØINT MATH        16 BIT, MUL/DIV
*
*   XDSU                    FIXED PØINT DØUBLE PRECISIØN SUBTRACT
*
          ,STX      ,XDSU+3         SAVE XR
          ,RØF      ,               RESET ØF
          ,LDX      ,XDSU
          ,LDX      ,0,1            XR-ADDR ØF HI B
          ,STA      ,XDSU+4         SAVE HI A
          ,TBA      ,
          ,ØRAI     ,0100000        SET SIGN FØR CARRY
          ,SUB      ,1,1            SUB LØ B
          ,ANAI     ,077777         MASK SIGN
          ,TAB      ,               SAVE RESULT
          ,TZA      ,
          ,SØFA     ,               GET CARRY
          ,RØF      ,               RESET ØF
          ,ADD      ,XDSU+4         ADD HI A
          ,SUB      ,0,1            SUB HI B
          ,INR      ,XDSU           SET RETURN
          ,LDX      ,XDSU+3         RESTØRE XR
          ,JMP      ,0              RETURN
          ,ØRG      ,*-1
   XDSU   ,ENTR     ,               ENTRY
          ,JMP      ,*-21
          ,DATA     ,0,0            TEMP STØRAGE
          ,MØRE     ,
          ,END      ,
```

```
*       FIXED PØINT MATH        16 BIT, MUL/DIV
*
*  XDDI                     FIXED PØINT DØUBLE PRECISIØN
*
BXDD   ,STX     ,XDDI+3         SAVE XR
       ,SØF     ,               PRESET ERRØR IND
       ,STA     ,XDDI+4         SAVE HI A
       ,STB     ,XDDI+5         SAVE LØ A
       ,LDX     ,XDDI           GET ADDR ØF B
       ,LDX     ,0,1
       ,LDA     ,0,1            GET HI B
       ,LDB     ,1,1            GET LØ B
       ,INR     ,XDDI           SET FØR RETURN
       ,CØMP    ,4              XR>-1
       ,STX     ,XDDI+8         INITIALIZE Q SIGN IND
       ,JIF     ,030,XDDI-2     JUMP IF ZERØ DIVISØR
       ,JAP     ,*+7            JUMP IF B PØS
       ,INR     ,XDDI+8         ELSE SET IND
       ,CALL    ,XDCØ           GET ABS VALUE
       ,JAN     ,XDDI-2         JUMP IF 1.000...
       ,LRLB    ,1              JØIN HI B,LØ B
       ,LLRL    ,1              NØRMALIZE HI B,LØ B
       ,IXR     ,               CØUNT SHIFTS
       ,JAN     ,*+4            JUMP IF NØRMALIZED
       ,JMP     ,*-4            ELSE LØØP
       ,LLSR    ,1              ADJUST
       ,LSRB    ,1              FIX LØ B
       ,STA     ,XDDI+6         SAVE HI B
       ,STB     ,XDDI+7         AND LØ B
       ,LDA     ,XDDI+4         GET HI A
       ,LDB     ,XDDI+5         AND LØ A
       ,JAP     ,*+7            JUMP IF A PØS
       ,INR     ,XDDI+8         ELSE SET IND
       ,CALL    ,XDCØ           GET ABS VALUE
       ,JAN     ,XDDI-2         JUMP IF 1.000...
       ,LRLB    ,1              JØIN HI A,LØ A
       ,LLRL    ,1              NØRMALIZE HI A,LØ A
       ,JXZ     ,*+7            JUMP IF NØRMALIZED TØ B
       ,JAN     ,XDDI-2         JUMP IF A GREATER THAN B
       ,DXR     ,               CØUNT SHIFTS
       ,JMP     ,*-6            AND LØØP
       ,RØF     ,               RESET ØVEL IF A LESS THAN B
```

```
        ,LLSR   ,1              ADJUST
        ,LSRB   ,1              FIX LØ A
        ,DIV    ,XDDI+6
        ,STB    ,XDDI+4         SAVE HI A/HI B
        ,STA    ,XDDI+5         SAVE LØ A
        ,TZA    ,
        ,MUL    ,XDDI+7
        ,DIV    ,XDDI+6
        ,STB    ,XDDI+7         SAVE LØ ØRDER CØRRECTIØN
        ,LDA    ,XDDI+5         GET LØ A
        ,DIV    ,XDDI+6
        ,TZA    ,
        ,STA    ,XDDI+6         SET HI ØRDER CØRRECTIØN > Ø
        ,LDA    ,XDDI+4         GET HI A/HI B
        ,CALL   ,XDSU           SUBTRACT...
        ,PZE    ,XDDI+6         ((HI A/HI B)*LØB)/HI B
        ,LDX    ,XDDI+8         GET Q SIGN IND
        ,JXZM   ,XDCØ           CØPLEMENT IF Q NEG
        ,LDX    ,XDDI+3         RESTØRE XR
        ,JMP    ,0              RETURN
        ,ØRG    ,*-1
XDDI    ,ENTR   ,
        ,JMP    ,RXDD
        ,DATA   ,0,0,0          TEMP STØRAGE
        ,DATA   ,0,0,0
        ,MØRE   ,
        ,END    ,
```

```
*       FLØATING PØINT MATH   16 BIT. MUL/DIV                92K0506-005 A
*
*  $HE                        CØMPUTES I**J (FIXD PT. NØS.)
*
          ,JMPM    ,$QS
          ,DATA    ,$HE+7
          ,STX     ,$HE+7
          ,LDX     ,$HE+4
          ,STX     ,*+4
          ,LDX     ,$HE+7
          ,JMPM    ,$PE          EXPØNENTIATIØN SUBRØUTINE
          ,DATA    ,0            MØDIFIED
          ,JMPM    ,$HS
          ,DATA    ,$HE+7
          ,JMP     ,0
          ,ØRG     ,*-1
$HE       ,ENTR    ,
          ,CALL    ,$SE,1
          ,DATA    ,0
          ,JMP     ,*-19
          ,DATA    ,0,0          TEMP. STØRAGE
          ,MØRE    ,
          ,END     ,
```

```
*       FLØATING PØINT MATH   16 BIT, MUL/DIV
*
*  $PE                        CØMPUTES A**I, A FL.PT, I FIXED PT,
*
         ,STX     ,$PE+7
         ,LDX     ,$PE+4
         ,STA     ,$PE+9
         ,STB     ,$PE+10
         ,LDA     ,0,1          PAR, I
         ,LDX     ,$PE+7
         ,JMPM    ,$QS
         ,DATA    ,$PE+7
         ,LDA     ,$PE+9
         ,LDB     ,$PE+10
         ,JMPM    ,$QE          A**B SUBRØUTINE
         ,DATA    ,$PE+7
         ,JMP     ,0
         ,ØRG     ,*-1
$PE      ,ENTR    ,
         ,CALL    ,$SE,1
         ,DATA    ,0
         ,JMP     ,*-20
         ,DATA    ,0,0,0,0
         ,MØRE    ,
         ,END     ,
```

```
*       FLØATING PØINT MATH   16 BIT, MUL/DIV
*
*  $QE                        CØMPUTES A**B FLØATING PT, NØS,
*
BGQE   ,STA      ,$QE+8
       ,STB      ,$QE+9
       ,CALL     ,ALØG,($QE+8)
       ,STX      ,$QE+7
       ,JØF      ,*+19
       ,LDX      ,$QE+4
       ,STX      ,*+3
       ,JMPM     ,$QM          B*LØG A
       ,DATA     ,0
       ,JØF      ,*+11
       ,STA      ,$QE+8
       ,STB      ,$QE+9
       ,CALL     ,EXP,($QE+8)
       ,JØF      ,(*+4)
       ,JMP      ,(*+4)
       ,ZERØ     ,03           A=B=0 ERRØR
       ,SØF      ,             SET ØFLØ F/F
       ,LDX      ,$QE+7
       ,JMP      ,0
       ,ØRG      ,*-1
$QE    ,ENTR     ,
       ,CALL     ,$SE,1,0
       ,JMP      ,BGQE
       ,DATA     ,0,0,0
       ,MØRE     ,
       ,END      ,
```

```
*       FLØATING PØINT MATH   16 BIT, MUL/DIV
*
*  ALØG                       CØMPUTES NATURAL LØG ØF A FLØATING
*                             PØINT NUMBER IN A,B REGISTERS
*
*
*                            ERRØR EXIT W/ A=B=0 IF ARG. NEGATIVE ØFLØ
*                            FLIP/FLØP SET, MAX. -VE RESULT IF ARG=0.
*
LØGE   ,EQU     ,010            ERRØR FLAG
ALØG   ,ENTR    ,
       ,CALL    ,$SE,1,0
       ,STX     ,LØG+3
       ,LDXI    ,(ALØG+4
       ,LDX     ,0,1
       ,LDA     ,0,1
       ,LDB     ,1,1
STRT   ,JAP     ,*+10
       ,TZB     ,
       ,LDAI    ,LØGE
       ,CALL    ,$ER            ERRØR RØUTINE
       ,ZERØ    ,03
       ,JMP     ,EXT-3
       ,JIF     ,030,EXT        MAX. NEGATIVE
       ,CALL    ,$FMS
       ,STX     ,LØG+10
       ,LASR    ,1
       ,STA     ,LØG+4
       ,STB     ,LØG+5
       ,JMPM    ,XDAD           ADD SQRT 2
       ,DATA    ,(LØG+12)
       ,STA     ,LØG+6          F+ SQRT 2
       ,STB     ,LØG+7
       ,LDA     ,LØG+4
       ,LDB     ,LØG+5
       ,JMPM    ,XDSU           SUB SQRT 2
       ,DATA    ,(LØG+12)
       ,JMPM    ,XDDI           CØMPUTE X
       ,DATA    ,(LØG+6)
       ,STA     ,LØG+6          SAVE X
       ,STB     ,LØG+7
       ,JMPM    ,XDMU
       ,DATA    ,(LØG+6)
```

```
          ,STA     ,LØG+8      X*X
          ,STB     ,LØG+9
          ,LDA     ,LØG+14
          ,STA     ,*+17
          ,LDA     ,LØG+19     C SUB 9 DØUBLE PRECISIØN
          ,LDB     ,LØG+20     C SUB 9
*                                CØMPUTE LØG ØF F =(((C9X*X+C7)X*X+C5)X*X+C3
          ,TAX     ,
          ,LDA     ,*+13
          ,IAR     ,
          ,IAR     ,
          ,STA     ,*+10
          ,SUB     ,LØG+15     IS IT END
          ,JAP     ,*+11       EXIT
          ,TXA     ,
          ,JMPM    ,XDMU
          ,DATA    ,(LØG+8)
          ,JMPM    ,XDAD
          ,DATA    ,0          SET BY PRØGRAM
          ,JMP     ,*-15
          ,TXA     ,
          ,LASR    ,2          B2 SCALE
          ,SUB     ,LØG+30     2 AT B 2
          ,JMPM    ,XDMU
          ,DATA    ,(LØG+6)    X
          ,SUB     ,LØG+16     1/2 AT B2
          ,CALL    ,$NML
          ,ADD     ,LØG+11     EXP 0202
          ,XXZ     ,(LØG+33)   CPA
          ,STA     ,LØG+4      TEMP LØG F
          ,STB     ,LØG+5      LØG F
          ,LDA     ,LØG+10     CHARAC.
          ,ASRA    ,NBIT-9
          ,SUB     ,LØG+17     EA-BASE 0200
          ,DAR     ,           EA-1
          ,CALL    ,$QS
          ,DATA    ,(LØG+6)
          ,JMPM    ,$QK        FL. ADD
          ,DATA    ,(LØG+4)
          ,JMPM    ,$QM        FL. MULTIPLY
          ,DATA    ,(LØG+31)   LØG 2 BASE E
          ,LDX     ,LØG+3
```

```
        ,JMP*    ,ALØG
EXT     ,LDA     ,LØG+29
LØG     ,LDB     ,LØG+30
        ,JMP     ,EXT-3
        ,BSS     ,8
        ,DATA    ,040400     BASE CHARAC+2
        ,DATA    ,026501     01.32404755576 SQRT 2 DBL PREC
        ,DATA    ,017333,(LØG+18),(LØG+28)
        ,DATA    ,010000     1/2 AT SCALE B2 FRØM LEFT
        ,DATA    ,0200       BASE CHARACT. AT B15
* DØUBLE PRECISIØN CØEFF. C9 THRU C1
        ,DATA    ,0152143,07361,0145516,023123
        ,DATA    ,0133042,010010,0102343,066136
        ,DATA    ,0107253,042327
        ,DATA    ,0100000,077777,040000   MAX FL.PT NEG.NØ.,2 AT B2
        ,DATA    ,040130,056205   FL.PT. LØG2 BASE E 0.542710277
YYY     ,CPA     ,
        ,MØRE    ,
        ,END     ,
```

```
*       FLØATING PØINT MATH   16 BIT, MUL/DIV
*
*  EXP                      CØMPUTES  E** X, ARG, X A FLØATING PT.
*
*                           CØNSTANT IN A,B REG, NØ ERRØR EXITS, THE
*                           RESULT CLAMPED AT 0,AND MAX ALLØWABLE NØ.IN
*                           FLØATING PT.
*
EXPS  ,STX     ,EXX+3
      ,LDX     ,EXX
      ,LDA     ,0,1
      ,LDB     ,1,1
      ,TZX     ,              RESET FLAG
      ,JAP     ,*+4           IS ARG PØSITIVE
      ,CPA     ,              NØ. CØMPLEMENT
      ,IXR     ,              SET FLAG.
      ,STA     ,EXX+4
      ,STB     ,EXX+5
      ,JMPM    ,$QL           FLØATING PT SUBTRACT,
      ,PZE     ,(EXX+25)
      ,JAN     ,*+11          IS X GREATER THAN 127*LØG2
      ,JXZ     ,*+5           YES. IS X PØSITIVE
      ,ZERØ    ,03            NØ. A=B=0
      ,JMP     ,EXPT          EXIT
      ,LDA     ,EXX+29
      ,LDB     ,EXX+30
      ,JMP     ,EXPT          EXIT
      ,LDA     ,EXX+4
      ,LDB     ,EXX+5
      ,JMPM    ,$QL
      ,PZE     ,(EXX+27)
      ,JAP     ,*+6           CØNTINUE
      ,TZB     ,
      ,LDA     ,EXX+20
      ,JMP     ,(EXPT)
      ,LDA     ,EXX+4
      ,LDB     ,EXX+5
      ,JMPM    ,$QM           MULTIPLY BY
      ,PZE     ,(EXX+21)
      ,STX     ,EXX+7
*                              EXTRACT CHARACTERISTIC
      ,STA     ,EXX+4
```

```
        ,ANA    ,EXX+8
        ,STA    ,EXX+6
        ,ERA    ,EXX+4
        ,LASL   ,8          SAVE MS PART ØF MANTISSA
        ,STA    ,EXX+4
        ,STB    ,EXX+5
        ,LDA    ,EXX+6
        ,SUB    ,EXX+9
        ,ASRA   ,NBIT-9
        ,TAB    ,
        ,JAN    ,*+14       IS EXPØNENT PØSITIVE
        ,SUB    ,EXX+10
        ,CPA    ,
        ,IAR    ,
        ,ADD    ,EXX+11
        ,STA    ,*+5
        ,TBA    ,
        ,ADD    ,EXX+12
        ,STA    ,*+17
        ,LDA    ,EXX+4
        ,PZE    ,0          ASRA, EXTRACT INTEGER I
        ,JMP    ,*+7
        ,CPA    ,           NØ INTEGER, PURE FRACTIØN
        ,IAR    ,
        ,ADD    ,EXX+13
        ,STA    ,*+9
        ,TZA    ,
        ,JXZ    ,*+3        IF -VE
        ,CPA    ,           I = I-1
        ,ASLA   ,NBIT-9
        ,STA    ,EXX+6
        ,LDA    ,EXX+4
        ,LDB    ,EXX+5
        ,PZE    ,0          LASL ØR LASR INSTR, GET F SCALED BØ
        ,SUB    ,EXX+14
        ,JAN    ,*+7        F LESS THAN ,5 ØR
        ,JAZ    ,*+5        EQUAL
        ,SUB    ,EXX+15
        ,JMP    ,*+9
*                           F LESS THAN 1/2
        ,ADD    ,EXX+15
        ,JXZ    ,*+5
```

```
          ,TZX      ,
          ,JMP      ,*+3
          ,IXR      ,
          ,STX      ,EXX+4
          ,LDX      ,EXX+7
          ,JXZ      ,*+11
          ,CPA      ,
          ,JBZ      ,*+7
          ,CPB      ,
          ,IBR      ,
          ,LRLB     ,1
          ,LSRB     ,1
          ,DAR      ,
          ,IAR      ,
          ,LASL     ,1          SCALE AT B-1
          ,JMPM     ,XDMU       FIXD PT. DØUBLE PRECISIØN MULTIPLY
          ,PZE      ,(EXX+23)
          ,TZX      ,
          ,JAP      ,*+12
          ,IXR      ,
          ,CPA      ,
          ,JBZ      ,*+7
          ,CPB      ,
          ,IBR      ,
          ,LRLB     ,1
          ,LSRB     ,1
          ,DAR      ,
          ,IAR      ,
          ,STX      ,EXX+7
*                               CØNVERT F2*LØG 2 TØ FLØATING PT
*                               F2 * LØG 2 LESS THAN 0.2 ØCTAL
          ,DECR     ,04         -1   TØ  X REG
          ,LRLB     ,1
          ,LLRL     ,1          SHIFT TØ SIGN
          ,JAN      ,*+5        NØRMALIZED EXIT
          ,DXR      ,
          ,JMP      ,*-4        LØØP
          ,LLSR     ,9
          ,LSRR     ,1
          ,STA      ,EXX+16
          ,TXA      ,
          ,ASLA     ,NBIT-9
```

```
        ,ADD    ,EXX+9
        ,ADD    ,EXX+16
        ,LDX    ,EXX+7
        ,JXZ    ,*+3
        ,CPA    ,
        ,STA    ,EXX+16
        ,STB    ,EXX+17
*                        COMPUTE  E **(F2 * LOG2) =
*                       ((F*F +11.99999490)+6*F)/((F*F+11.999-)-6*F)
        ,LDX    ,EXX+4
        ,JMPM   ,$QM     FLOAT. MULTIPLY
        ,PZE    ,(EXX+16)
        ,STA    ,EXX+4
        ,STB    ,EXX+5
        ,JMPM   ,$QK     FLOAT. ADD
        ,PZE    ,(EXX+18)
        ,STA    ,EXX+4
        ,STB    ,EXX+5
        ,LDA    ,EXX+16
        ,LDB    ,EXX+17
        ,JMPM   ,$QM     6*F
        ,PZE    ,(EXX+31)
        ,STA    ,EXX+16
        ,STB    ,EXX+17
        ,CPA    ,        COMPL.
        ,JMPM   ,$QK     FL. ADD
        ,PZE    ,(EXX+4)
        ,STA    ,EXX+33
        ,STB    ,EXX+34
*                        GENERATE G ADDRESS
        ,TXA    ,
        ,ASLA   ,1
        ,ADD    ,EXX+39
        ,STA    ,*+11
*                        CONTINUE COMP OF  E**F
        ,LDA    ,EXX+4
        ,LDB    ,EXX+5
        ,JMPM   ,$QK     FL. ADD
        ,PZE    ,(EXX+16)
        ,JMPM   ,$QN     FL. DVD.
        ,PZE    ,(EXX+33)
        ,JMPM   ,$QM     G* E**F
```

```
          ,PZE    ,0          ADDRESS STØRED BY PRØGRAM
          ,ADD    ,EXX+6
    EXPT  ,LDX    ,EXX+3
          ,JMP    ,0
          ,ØRG    ,*-1
    EXP   ,ENTR   ,
          ,CALL   ,$SE,1,0
          ,JMP    ,EXPS
    EXX   ,EQU    ,EXP+4
          ,BSS    ,5
          ,DATA   ,077600,040000,NBIT-1
          ,ASRA   ,0
          ,LASL   ,0
          ,LASR   ,0
          ,DATA   ,040000,020000      ,5,,25 AT B0
          ,BSS    ,2
          ,DATA   ,041137,077776,040300
          ,DATA   ,040334,025216,054271,02776
          ,DATA   ,041730,01715,036535,003635
          ,DATA   ,077777,077777,040740,0
          ,BSS    ,2
          ,DATA   ,040353,050500,040314,06775
          ,DATA   ,(EXX+35)
          ,MØRE   ,
    20    ,END    ,
```

```
*       FLØATING PØINT MATH   16 BIT, MUL/DIV
*  CØS                    CØMPUTES CØSINE ØF FLØATING PT.(RADIANS)
*
CØS     ,ENTR    ,
        ,CALL    ,$SE,1,0
        ,LDA     ,CTMP-2
        ,LDB     ,CTMP-1
        ,JMPM    ,$QL
        ,DATA    ,(CØS+4)*
        ,STA     ,CTMP
        ,STB     ,CTMP+1
        ,JMPM    ,SIN
        ,DATA    ,(CTMP)
        ,JMP*    ,CØS
        ,DATA    ,040344,041767  PI/2 FL. PØINT
CTMP    ,DATA    ,0,0
        ,MØRE    ,
        ,END     ,
```

```
*       FLØATING PØINT MATH   16 BIT, MUL/DIV
*
*
*  SIN                        CØMPUTES SINE ØF FLØATING PT.(RADIANS)
*
SIN    ,ENTR    ,             ENTER
       ,CALL    ,$SE,1,0
       ,STX     ,Y
       ,LDXI    ,(SIN+4)
       ,LDX     ,0,1
       ,LDA     ,0,1
       ,LDB     ,1,1
       ,JIF     ,030,(EXI+1)
       ,JMPM    ,$QM          FL. MPY BY 2/PI
       ,DATA    ,(Y+6)
       ,TZX     ,
       ,JAP     ,*+4
       ,CPA     ,
       ,IXR     ,             FLAG FØR NEGATIVE ARG.
       ,STX     ,Y+5          SAVE SIGN
       ,CALL    ,$FMS
       ,STA     ,Y+1
       ,TXA     ,
       ,SUB     ,Y+8          BASE  0200 B8
       ,TZX     ,
       ,JAP     ,*+5
       ,CPA     ,
       ,IAR     ,
       ,IXR     ,
       ,ASRA    ,NBIT-9
       ,SUB     ,Y+11         22,26 SZ MANT.
       ,JAP     ,EXI          A=B=0 SINE=0
       ,ADD     ,Y+11
       ,IXZ     ,*+8
       ,ARA     ,Y+9          LASR
       ,STA     ,*+2
       ,LDA     ,Y+1          ARG A
       ,PZE     ,0            LASR
       ,JMP     ,(Z+1)
       ,JAZ     ,Z
       ,DAR     ,
       ,IAR     ,
```

```
        ,JAP    ,*+6
        ,LDA    ,Y+1
        ,LRLB   ,1
        ,JMP    ,*+14
        ,ØRA    ,Y+10       LASL
        ,STA    ,*+2
        ,LDA    ,Y+1
        ,PZE    ,0
        ,LDX    ,Y+5        SIGN ØF ARG
        ,LRLB   ,1
        ,LLRL   ,1
        ,JAP    ,*+4
        ,DXR    ,
        ,ERA    ,Y+12       100000
        ,STX    ,Y+5
        ,LLRL   ,1
        ,JAP    ,*+4        CØMPUTE SIVE(A)
        ,CPA    ,           CØMUTE SINE(PI/2-A)
        ,CPB    ,
        ,LSRB   ,1
        ,JMP    ,Z+1
Z       ,LDA    ,Y+1        A  (FØR EXP=0 PATH)
        ,STA    ,Y+1        X
        ,STB    ,Y+2        XX
        ,JMPM   ,XDMU       CØMPUTE X**2 DØUBL PRECISIØN
        ,DATA   ,(Y+1)
        ,STA    ,Y+3
        ,STB    ,Y+4
        ,LDA    ,Y+27
        ,STA    ,*+17
*                           C11)X*X+C9)X*X+C7)X*X+C5 ETC.
        ,LDA    ,Y+15       C SUB 11 DØUBL PRECISIØN
        ,LDB    ,Y+16
        ,STA    ,Y+13       SAVE PARTIAL ANSW.
        ,LDA    ,*+13
        ,IAR    ,
        ,IAR    ,
        ,STA    ,*+10
        ,SUB    ,Y+28
        ,JAP    ,(*+11)     EXIT
        ,LDA    ,Y+13       RESTØRE PARTIAL ANSWER
        ,JMPM   ,XDMU
```

```
        ,DATA   ,(Y+3)      X**2
        ,JMPM   ,XDAD
        ,DATA   ,0          MØDIFIED
        ,JMP    ,(*-15)     BACK TØ LØØP
        ,LDA    ,Y+13
        ,LASR   ,1            SCALE B1
        ,ADD    ,Y+8          1 AT B1(=040000)
        ,JMPM   ,XDMU
        ,DATA   ,(Y+1)      X
        ,STX    ,Y+13       SAVE SIGN
        ,JIF    ,030,*+9
        ,JMPM   ,$NML       GØ TØ NØRMALIZE RØUTINE.
        ,ADD    ,Y+14       EXP1 0201 AT B8
        ,LDX    ,Y+5
        ,JXZ    ,(*+3)      IS RESULT PØSITIVE
        ,CPA    ,           NØ. CØMPLEMENT
        ,LDX    ,Y
        ,JMP*   ,SIN        EXIT
EXI     ,ZERØ   ,03         A=B=0
        ,LDX    ,Y
        ,JMP*   ,SIN        EXIT
Y       ,BSS    ,5
        ,DATA   ,0,040121,037141,040000
        ,LASR   ,0
        ,LASL   ,
        ,DATA   ,22                 MAX BITS ØF MANT.
        ,DATA   ,0100000    SIGN BIT
        ,DATA   ,0,040200   TEMP,EXPØNENT 1
*                                DBL PREC. CØEFF C11 THRU C1
        ,DATA   ,0177777,070652,05,017774
        ,DATA   ,0177546,045735,05063,027360
        ,DATA   ,0126521,03071,044417,066521
        ,DATA   ,(Y+15),(Y+26)
        ,MØRE   ,
        ,END    ,
```

```
*       FLØATING PØINT MATH   16 BIT, MUL/DIV
*
*  ATAN                    CØMPUTES ARCTANGENT ØF FLØATING PT. NØ.
*
ATAN   ,ENTR    ,
       ,CALL    ,$SE,1,0
BGIN   ,STX     ,STT
       ,LDXI    ,(BGIN-1)
       ,LDX     ,0,1
       ,LDA     ,0,1
       ,LDB     ,1,1
       ,INCR    ,04          ØNE TØ XR
       ,STA     ,STT+1       SAVE A
       ,STB     ,STT+2       SAVE AA
       ,JMPM    ,$QM         N*N FL MPY
       ,DATA    ,(STT+1)
       ,STA     ,STT+3       N**2
       ,STB     ,STT+4       N**2
       ,LDA     ,STT+1
       ,LDB     ,STT+2
       ,JAP     ,*+4
       ,CPA     ,
       ,DXR     ,            FLAG FØR -VE
       ,STA     ,TEMP+2      SAVE ABS N
       ,JMPM    ,$QL         FL SUB
       ,DATA    ,(STT+5)     10**-3=.0004061115645
       ,JAP     ,(*+7)
       ,LDA     ,STT+1
       ,LDB     ,STT+2
       ,LDX     ,STT
       ,JMP*    ,ATAN        EXIT
       ,LDA     ,TEMP+2      ABS.N
       ,LDB     ,STT+2       NN
       ,JMPM    ,$QL         FL.SUB
       ,DATA    ,(STT+7)     TAN PI/24=0.10331720371
       ,JAN     ,(CF0)
       ,LDA     ,TEMP+2      ABS. N
       ,LDB     ,STT+2       NN
       ,JMPM    ,$QL         FL SUB
       ,DATA    ,(STT+9)     1.0
       ,JAN     ,(CF1)
       ,LDA     ,TEMP+2      ABS. N
```

```
        ,LDB    ,STT+2      NN
        ,JMPM   ,$QL        FL SUB
        ,DATA   ,(STT+11)   10**8=0575,360,400
        ,JAN    ,(*+9)
        ,LDA    ,STT+13     PI/2
        ,LDB    ,STT+14     PI/2
        ,XXZ    ,(STT+41)   CPA IF XR=0
        ,LDX    ,STT
        ,JMP*   ,(ATAN)     EXIT
*                           CØNTD FRACT. 2
        ,JMPM   ,(CFEV)
        ,DATA   ,(STT+25)   ADDR ØF 1ST CØEFF
        ,JMPM   ,$QN        FL DIV (N)
        ,DATA   ,(STT+1)    N
        ,LDX    ,STT        RESTØRE XR
        ,CPA    ,
        ,CALL   ,$QK,(STT+13)
        ,JMP*   ,(ATAN)     EXIT
*                           CØNTD FRACTIØN 1
CF1     ,JMPM   ,(CFEV)     EVALUATE CØNTD FRAC.
        ,DATA   ,(STT+15)   ADDR ØF 1ST CØEFF
        ,JMPM   ,$QM        FL MULTIPLY (N)
        ,DATA   ,(STT+1)    N
        ,LDX    ,STT        RESTØRE XR
        ,JMP*   ,(ATAN)     EXIT
*                           PØLYNØMIAL  C3)N*N+C2)N*N+C1)N
CF0     ,LDA    ,STT+39     C3
        ,LDB    ,STT+40     C3
        ,JMPM   ,$QM        FL MUL
        ,DATA   ,(STT+3)    N**2
        ,JMPM   ,$QK        ADD
        ,DATA   ,(STT+37)   C2
        ,JMPM   ,$QM        MUL
        ,DATA   ,(STT+3)    N**2
        ,JMPM   ,$QK        ADD
        ,DATA   ,(STT+35)   C1
        ,JMPM   ,$QM        MUL
        ,DATA   ,(STT+1)    N
        ,LDX    ,STT        RESTØRE XR
        ,JMP*   ,(ATAN)     EXIT
*                           EVALUATIØN ØF CØNTD FRACTIØN
CFR     ,LDX    ,CFEV       REF.
```

```
        ,LDX    ,0,1        ADDRESS ØF ØRIGIN ØF TABLE ØF CØNSTS.
        ,LDA    ,8,1
        ,LDB    ,9,1
        ,JMPM   ,$QK        ADD
        ,DATA   ,(STT+3)    N**2
        ,STA    ,TEMP       SAVE
        ,STB    ,TEMP+1     SAVE
        ,LDA    ,6,1
        ,LDB    ,7,1
        ,JMPM   ,$QN        DIVIDE
        ,DATA   ,(TEMP)
        ,STA    ,TEMP
        ,STB    ,TEMP+1
        ,LDA    ,4,1
        ,LDB    ,5,1
        ,JMPM   ,$QK        ADD
        ,DATA   ,(TEMP)
        ,JMPM   ,$QK        ADD
        ,DATA   ,(STT+3)    N**2
        ,STA    ,TEMP
        ,STB    ,TEMP+1
        ,LDA    ,2,1
        ,LDB    ,3,1
        ,JMPM   ,$QN        FL DVD
        ,DATA   ,(TEMP)
        ,STX    ,*+3
        ,JMPM   ,$QK        ADD
        ,DATA   ,0          ADDRESS STØRED BY PRGRM.
        ,INR    ,CFEV
        ,JMP*   ,(CFEV)
CFEV    ,ENTR   ,
        ,JMP    ,(CFR)
TEMP    ,BSS    ,3
STT     ,BSS    ,5
*                           CØNST. FØR 16 BIT WØRD
        ,DATA   ,035701,042233,037503,031720      10**-3,TAN PI/24
        ,DATA   ,040300,0,046737,027410           1.0,10**8
        ,DATA   ,040344,041767   PI/2
*                           CØNSTANTS FØR CØNT. FRACT. NØ 1  FL.PT.
        ,DATA   ,037572,021600                    ,2388229612
        ,DATA   ,040516,017617                    2.445205396
        ,DATA   ,040576,014262                    3.943529798
```

```
        ,DATA   ,0137453,011150                         -1.314747223
        ,DATA   ,040363,05502                           1.798249626
*                       CØNSTANTS FØR CØNTD. FRACT,NØ,2 FL.PT.
        ,DATA   ,040177,077774                          .9999992083
        ,DATA   ,0140052,024446                         -.3332870775
        ,DATA   ,040114,047565                          .5985998078
        ,DATA   ,0140476,04640                          -.0635500089
        ,DATA   ,037745,015371                          .3953544718
*                       CØEFF. FØR PØLYNØMIAL.
        ,DATA   ,040177,077777                          .9999999207
        ,DATA   ,0140052,024566                         -.3332966338
        ,DATA   ,037544,016021                          .1957408066
END1    ,CPA    ,
        ,MØRE   ,
        ,END    ,
```

```
*       FLØATING PØINT MATH   16 BIT, MUL/DIV
*
*  SQRT                     CØMPUTES SQUARE RØØT ØF A FLØATING PØINT
*                           NØ. IN A,B REGISTERS. ØVERFLØW FLIP/FLØP
*                           SET IF ARGUMENT IS NEGATIVE.
*
SQRT   ,ENTR    ,           ENTER
       ,CALL    ,$SE,1,0
       ,STX     ,SQT
       ,LDXI    ,(SQRT+4)
       ,LDX     ,0,1
       ,LDA     ,0,1
       ,LDB     ,1,1
       ,SØF     ,
       ,JAN     ,SQT-3
       ,RØF     ,
       ,JIF     ,030,SQT-3
       ,STA     ,SQT+1      SAVE AR
       ,ANA     ,SQT+10     EXTRACT CHARACTERISTIC
       ,STA     ,SQT+3      SAVE
       ,TZX     ,
       ,LRLA    ,8          IS EXPØNENT ØDD
       ,XAN     ,SQT+11     IF YES IXR
       ,LRLA    ,NBIT-8
       ,ERA     ,SQT+1      EXTRACT MANTISSA
       ,LASL    ,6
       ,XXZ     ,SQT+12     LASL 1 IF EVEN EXPØNENT
       ,STA     ,SQT+1      M/2 AT B0
       ,STB     ,SQT+2      M/2 AT B0
       ,TAB     ,
*                           ØDD EXPØN. CØMPUTE M*.7+.22
       ,JXZ     ,(EVEN)     JMP IF EXP. IS EVEN
       ,ASRB    ,1
       ,STB     ,SQT+4      TEMP
       ,ADD     ,SQT+4
       ,ASRB    ,1
       ,STB     ,SQT+4
       ,ADD     ,SQT+4
       ,ADD     ,SQT+8      0.22. X SUB I AT B0 RESULT
       ,JMP     ,(*+6)
*                           EVEN PATH.X SUB I= M*0.44+0.34
EVEN   ,SRB     ,3
```

```
        ,STB    ,SQT+4
        ,ADD    ,SQT+4
        ,ADD    ,SQT+9      0,34 AT B0
        ,STA    ,SQT+5      X SUB I AT B0
        ,ASRA   ,1          X SUB I*(1/2) AT B0
        ,STA    ,SQT+4      1/2(X SUB I) AT B0
        ,LDA    ,SQT+1      M/2 B0
        ,TZB    ,
        ,DIV    ,SQT+5      X AT B0
        ,TBA    ,
        ,ADD    ,SQT+4      X/2 AT B0
        ,STA    ,SQT+5      XSUB(I+1) AT B0
        ,LDA    ,SQT+1      M/2
        ,LDB    ,SQT+2      MM/2
        ,CALL   ,XDDI       1/2(M/XSUB(I+1)) B0
        ,DATA   ,(SQT+5)
        ,LRLB   ,1
        ,LLRL   ,1
        ,ADD    ,SQT+5      X SUB (I+1) AT B0,1/2 X AT B-1
        ,LLSR   ,9          PØSITIØN MANTISSA
        ,LSRB   ,1
        ,STA    ,SQT+1
        ,LDA    ,SQT+3      CHARACT.
        ,LSRA   ,NBIT-9     EXP AT B15
        ,IAR    ,           ADD ØNE
        ,ASRA   ,1          E/2 (ØDD EXP +1)/2,(EVEN EXP+0)/2
        ,ADD    ,SQT+7      BASE EXP/2 AT B15
        ,ASLA   ,NBIT-9     EXP      E-1 AT B8
        ,ØRA    ,SQT+1
        ,LDX    ,SQT        RESTØRE XR
        ,JMP*   ,SQRT       EXIT
SQT     ,BSS    ,6
        ,DATA   ,0,0100     0100=BASE EXP/2 AT B15
        ,DATA   ,022000,034000,077600  CØNSTANTS AT B0,MSK AT BE
        ,IXR    ,
        ,LASL   ,1
        ,MØRE   ,
        ,END    ,
```

```
*       FLØATING PØINT MATH   16 BIT, MUL/DIV
*
*  SQM                      FLØATING DIVIDE AND FLØATING MULTIPLY
*                           ENTER W/ PARAMETER IN A,B REGS AND
*                           ADDRESS ØF 2ND PARAMETER IN CALL SEQUENCE
*                           ERRØR EXIT W/ A=B=0 AND ØVERFLØW SET,
*                           USES FXD PT DBL PREC, MUL AND DIVIDE
*                           XDMU , XDDI SUBRØUTINES,
*                           CØMMØN PATH
*
QMBG   ,STX     ,$QM+4      SAVE XR
       ,JIF     ,030,QMCP+6    RG A=0,EXIT
       ,TZX     ,
       ,JAP     ,*+4        IS IT PØSITIVE
       ,CPA     ,           NØ, CØMPLEMENT
       ,IXR     ,           SET FLAG
       ,STX     ,$QM+11     SIGN
       ,CALL    ,$FMS
       ,LASR    ,1
       ,STA     ,$QM+5      SAVE A
       ,STB     ,$QM+6      AND AA
       ,STX     ,$QM+9      SAVE CHAR
       ,LDX     ,$QM-3      GET PAR ADDRESS
       ,LDA     ,0,1        EB,B
       ,LDB     ,1,1        BB
       ,JIF     ,030,QMCP+7    ARG B=0 ERRØR FØR DIV
       ,LDX     ,$QM+11     CØNTINUE  SIGN
       ,JAP     ,*+4
       ,CPA     ,
       ,DXR     ,
       ,STX     ,$QM+11     SIGN
       ,CALL    ,$FMS
       ,STX     ,$QM+10     CHAR EB
       ,STA     ,$QM+7
       ,STB     ,$QM+8
       ,JØF     ,QMDV       GØ TØ DIVISIØN
       ,LDA     ,$QM+9      EA
       ,SUB     ,$QM+12     BASE CHARACT,
       ,ADD     ,$QM+10     EB
       ,STA     ,$QM+9      EA+EB
       ,LDA     ,$QM+5      A
       ,LDB     ,$QM+6      AA
```

```
         ,LASL    ,1          NØRMALIZE FØR MULTIPLICATIØN
         ,JMPM    ,XDMU       FXD.PT.DØUBLE PRECISIØN MULTIPLY
         ,PZE     ,($QM+7)    ADDRESS ØR ARG B.
         ,RØF     ,
QCMN     ,STA     ,$QM+5      SAVE RESLT TEMP
         ,LRLA    ,1          MØVE TØ SIGN
         ,JAP     ,*+20
         ,LDA     ,$QM+9      RESULTANT CHARACTERISTIC.
         ,XØF     ,QMER+4
         ,RØF     ,
         ,STA     ,$QM+9      SAVE
         ,LDA     ,$QM+5      GET RESULT
         ,LASR    ,8          FØRMAT FØR FL.PT.
         ,ØRA     ,$QM+9      CØMMØN PATH
QMCP     ,JAN     ,QMER
         ,LDX     ,$QM+11     SIGN
         ,JXZ     ,*+3        ADD CHARACTERISTIC
         ,CPA     ,           IS RESULT PØSITIVE (X)=0
         ,RØF     ,           NØ. CØMPLEMENT
         ,LDX     ,$QM+4      RESTØRE X REG
         ,JMP*    ,$QM-7      RETURN
         ,LDA     ,$QM+9      RESULTANT CHARACTERISTIC
         ,JØF     ,*+3        SUB ØNE FRØM CHARA. IF MULTIPLY
         ,SUB     ,$QM+14
         ,JAN     ,QMER       NØ TØØ LØW
         ,STA     ,$QM+9      SAVE RESULT. CHARACT.
         ,LDA     ,$QM+5      GET RESULT
         ,LASR    ,7          FØRMAT FØR PL.PT.
         ,JMP     ,QMCP-1
*                              DIVISIØN ØNLY
*
QMDV     ,LDA     ,$QM+9      EA
         ,SUB     ,$QM+10     EA-EB
         ,ADD     ,$QM+12     BASE CHARACTERISTIC
         ,JAN     ,*+12       ERRØR. RESULT LØW.
         ,STA     ,$QM+9      EA-EB
         ,LDA     ,$QM+5      A
         ,LDB     ,$QM+6      AA
         ,JMPM    ,XDDI       FXD PØINT DØUBLE PRECISIØN DIVIDE
         ,PZE     ,($QM+7)    ADDRESS ØF ARG B.
         ,SØF     ,
         ,JMP     ,QCMN
```

```
QMER  ,SØF    .
      ,ZERØ   ,03          A=B=0
      ,JMP    ,QMCP+7
      ,DATA   ,0124122     ADD $QM+14
      ,STA    ,$QM+4
      ,LDA    ,$QN
      ,STA    ,$QM-7
      ,LDA    ,$QM+4
      ,JMP    ,$QM-6
$QN   ,ENTR   .
      ,SØF    .
      ,JMP    ,*-8
      ,STA    ,$QM+4
      ,LDA    ,$QM
      ,STA    ,*+4
      ,LDA    ,$QM+4
      ,JMP    ,*+3
      ,PZE    ,0
      ,CALL   ,$SE,1,0
      ,JMP    ,QMBG
$QM   ,ENTR   .
      ,RØF    .
      ,JMP    ,*-15
      ,BSS    ,7
      ,DATA   ,0,040000,0100000,0200
      ,MØRE   .
4     ,END    .
```

```
*       FLØATING PØINT MATH  16 BIT, MUL/DIV
*
*  $QK                        FLØATING PØINT ADD
*
$QK    ,ENTR    ,
       ,RØF     ,
       ,CALL    ,$FAS
       ,MØRE    ,
       ,END     ,
```

```
*       FLØATING PØINT MATH  16 BIT, MUL/DIV
*
*  $QL                        FLØATING PØINT SUBTRACT
*
$QL    ,ENTR    ,
       ,SØF     ,
       ,CALL    ,$FAS
       ,MØRE    ,
       ,END     ,
```

```
*       FLØATING PØINT MATH   16 BIT, MUL/DIV
*
*  $FAS                       FLØATING PØINT ADD/SUBTRACT
*
FASS   ,TZX      ,             SET FLAG
       ,LDA      ,$FAS+4
       ,JAP      ,*+4
       ,CPA      ,             CØMP  AND
       ,DXR      ,
       ,STX      ,$FAS+8       TEMP FLAG 1
       ,CALL     ,$FSM
       ,STA      ,$FAS+4       A   (MS MANT)
       ,STX      ,$FAS+11      EXP (EA)
       ,STB      ,$FAS+6       AA  (LS MANT)
       ,LDX      ,ENT1+7
       ,LDA      ,0,1
       ,LDB      ,1,1
       ,XØF      ,($FAS+13)    CPA IF FL.SUB PATH
       ,LDX      ,$FAS+8       FLAG 1
       ,JAP      ,*+4
       ,CPA      ,
       ,IXR      ,
       ,STX      ,$FAS+9       TEMP FLAG 3
       ,CALL     ,$FSM
       ,STX      ,$FAS+10      EB  (EXP ØF B)
       ,STA      ,$FAS+5       B   (MS MANT)
       ,STB      ,$FAS+7       BB  (LS MANT)
       ,TXA      ,             EB
       ,SUB      ,$FAS+11      EB -EA
       ,ASRA     ,NBIT-9       SF 15
       ,TZX      ,
       ,JAP      ,*+5          IF +VE BYPASS
       ,CPA      ,             NEGATE
       ,IAR      ,              SET
       ,IXR      ,              FLAG ?
       ,TAB      ,
       ,SUB      ,$FAS+12      CHK FØR MAX
       ,JAN      ,*+3          ØK
       ,LDB      ,$FAS+12
       ,TBA      ,
       ,ØRA      ,$FAS+14      GEN LASR INSTR
       ,STA      ,*+7
```

```
        ,TXA    ,
        ,ADDI   ,($FAS+4)  ADDR. ØF ARG A
        ,TAX    ,
        ,LDA    ,0,1       SMALLER ØF A,B
        ,LDB    ,2,1
        ,PZE    ,0         MØDIFIED SHIFT INSTR
        ,STA    ,0,1
        ,STB    ,2,1
        ,LDA    ,6,1       EXPØNENT (LARGER)
        ,STA    ,$FAS+10   SAVE
        ,LDA    ,$FAS+6    AA
        ,LDB    ,$FAS+4    A
        ,LDX    ,$FAS+9
        ,JXZ    ,FADP      GØ TØ ARITH. ADD
* ARITHMETIC SUBTRACTIØN PATH
        ,SUB    ,$FAS+7    BB
        ,JAP    ,*+4
        ,DBR    ,
        ,ERA    ,$FAS+15   SET SIGN PLUS
        ,LLRL   ,NBIT      EXCHANGE A,AA-BB
        ,SUB    ,$FAS+5    A-B
        ,JIF    ,030,FASE  EXIT PATH
        ,JMPM   ,$NML      NØRM.
        ,JMP    ,*+16
FADP    ,ADD    ,$FAS+7    BB****ARITH ADDITIØN PATH*****
        ,JAP    ,*+5
        ,IBR    ,          A+1
        ,ERA    ,$FAS+15   SET SIGN PLUS
        ,RØF    ,
        ,LLRL   ,NBIT      EXCHANG A,AA+BB
        ,ADD    ,$FAS+5    A+B
        ,LASR   ,1
        ,JAP    ,*+4       IS IT ØVER FLØW
        ,LASR   ,1         YES. ADD ØNE IN CHAR. FIELD
        ,ERA    ,$FAS+15   SET SIGN
        ,LASR   ,7
        ,ADD    ,$FAS+10   LARGER EXPØNENT
        ,JAP    ,FASE+3    NØ ERRØR
* A IS NEGATIVE - ØVERFLØ FØR ADDITIØN, UNDERFLØ FØR SUBTRACTIØN
        ,TZR    ,
        ,LDAI   ,AERR
        ,CALL   ,$ER
```

```
        ,ZERØ   ,03         A=B=0
FASE    ,LDX    ,$FAS+3     RESTØRE XR
        ,JMP*   ,$FAS
        ,RØF    ,
        ,STA    ,$FAS+4     TEMP
        ,TXA    ,           1- NØ.CØMPL. BY $NML,0-ØRIG. RESULT
        ,ADD    ,$FAS+8     SIGNØF ARGA A
        ,TAX    ,
        ,LDA    ,$FAS+4     RESTØRE AR
        ,JXZ    ,*+3
        ,CPA    ,
        ,LDX    ,$FAS+3     RESTØRE XR
        ,JMP*   ,$FAS       RETURN
AERR    ,EQU    ,0              ERRØR FLAG
        ,STX    ,$FAS+3
        ,STA    ,$FAS+4
        ,LDA    ,$FAS
        ,SUBI   ,4
        ,TAX    ,
        ,LDA    ,0,1
        ,STA    ,*+5
        ,IAR    ,
ENT1    ,STA    ,$FAS       RETURN PØINTER
        ,JMP    ,*+3
        ,DATA   ,0
        ,CALL   ,$SE,1,0
        ,JMP    ,FASS       GØ TØ START
$FAS    ,ENTR   ,
        ,JMP    ,ENT1-9
        ,BSS    ,9
        ,DATA   ,22,05211   SIZE MANT   ,CPA
        ,LASR   ,0
        ,DATA   ,0100000
        ,MØRE   ,
        ,END    ,
```

```
*       FLØATING PØINT MATH   16 BIT, MUL/DIV
*
*  $FSM                         SEPARATE MANTISSA
*
$FSM   ,ENTR    ,               AR,BR CØNTAIN FL.PT.
       ,STA     ,TFSM           EXIT W/ EXP IN XR,MANT
       ,ANA     ,TFSM+1          AT SF0
       ,TAX     ,
       ,ERA     ,TFSM
       ,LASL    ,8              MANT. TØ HI ØRDER SF 1
       ,JMP*    ,$FSM
TFSM   ,DATA    ,0
       ,DATA    ,077600         MASK
$FMS   ,EQU     ,$FSM
       ,MØRE    ,
       ,END     ,
```

```
*       FLØATING PØINT MATH   16 BIT, MUL/DIV
*
*  $NML                       NØRMALIZE RØUTINE
*                             ENTER WITH NR. IN A,B SF 0
*
$NML   ,ENTR    ,
       ,JIF*    ,030,$NML    EXIT IF ZERØ
       ,TZX     ,
       ,JAP     ,*+5
       ,IXR     ,
       ,CALL    ,XDCØ        NEGATE A,B
       ,STX     ,NMLT
       ,TZX     ,
       ,LRLB    ,1
       ,LLRL    ,1
       ,JAN     ,*+5         ØUT IF NØRMALIZED
       ,IXR     ,            NØ. UPDATE NR.
       ,JMP     ,*-4          ØF LEADING ZERØS
       ,LLSR    ,9
       ,LSRB    ,1
       ,STA     ,NMLT+1      SAVE MANT.
       ,TXA     ,
       ,ASLA    ,NBIT-9      FØRMAT FØR FL.PT.EXP.
       ,CPA     ,
       ,IAR     ,
       ,ØRA     ,NMLT+1      ADD MANT.
       ,LDX     ,NMLT        FLAG FØR -VE NR
       ,JMP*    ,$NML
NMLT   ,DATA    ,0,0
       ,MØRE    ,
       ,END     ;
```

```
*       FLØATING PØINT MATH   16 BIT, MUL/DIV
*
*  $IS                      CØNVERTS FIXED PT. INTEGER TØ FLØATING PT.
*                           INPUT IN A, ØUTPUT IN A,B REGS.
*
BIS     ,LDX     ,$IS-1
        ,TZB     ,           ZERØ TØ B
        ,JAZ     ,*+23
        ,STA     ,$IS+4      SAVE NØ.
        ,LASR    ,NBIT-1         SHIFT RIGHT 15,17 PLACES
        ,STA     ,$IS+5
        ,LDA     ,$IS+4
        ,JAP     ,*+4        IS NØ NEGATIVE
        ,CPA     ,           YES. TWØS CØMPLEMENT
        ,IAR     ,           THE INTEGER.
        ,LDBI    ,NBIT+0200      BASE CHARACTERISTIC +EXP 16,18
        ,JAN     ,*+8        GØ FØRMAT IF NEGATIVE
        ,LRLA    ,1          LØGICAL LEFT 1
        ,DBR     ,           IF SIGN SET, GØ FØRMAT THE NØ.
        ,JAN     ,*+4        IF NØT DECREMENT EXPØNENT AND NØRMALIZE
        ,JMP     ,*-4
        ,LLRL    ,2*NBIT-9       EXPØNENT AND MØST SIG. MANTISSA IN AR
        ,ERA     ,$IS+5      CØMPLEMENT IF NEGATIVE
        ,LSRB    ,1          SHIFT SIGN ØF B TØ BIT 14 AND SET SIGN=0
        ,STA     ,0,1        RESULT IN LØCATIØN SPECIFIED.
        ,STB     ,1,1        SECØND WØRD ØF RESULT IN SPECIFIED LØC.
        ,LDX     ,$IS+3      RESTØRE X
        ,JMP     ,0
        ,ØRG     ,*-1
$QS     ,ENTRY   ,
        ,CALL    ,$SE,1,0
$IS     ,STX     ,*+3
        ,JMP     ,BIS
        ,DATA    ,0,0,0
        ,MØRE    ,
        ,END     ,
```

```
NBIT   ,SET     ,16
$MDV   ,SET     ,0              NØ MULTIPLY ØR DIVIDE
*
*  $PS                          CØNVERTS  FLØATING PT, TØ AN INTEGER
*                               INPUT A,B ØUTPUT IN A AND LØCATIØN SPECIFID
*
BPS    ,DECR    ,04
       ,JIF     ,030,*+38      GØ SAVE RESULT IF A=B=0,
       ,JAP     ,*+4           IF PØSITIVE BYPASS
       ,CPA     ,              CØMPLEMENT
       ,IXR     ,
       ,SUB     ,$PS+4         BASE CHARACTERISTIC+1
       ,JAN     ,*+15          IF NEGATIVE, THE NØ IS A FRACTIØN
       ,SUB     ,$PS+5         CHECK FØR MAX EXP ØF  16
       ,JAN     ,*+15          IF LESS CØNTINUE,
       ,SUB     ,$PS+6
       ,JIF     ,070,*+6       IF -2**15 GØ TØ EXIT PATH, A=B=X=0
       ,SØF     ,              SET ØVERFLØW FØR ERRØR
       ,ZERØ    ,03            ZERØ TØ A,B REGS,
       ,JMP     ,*+22          GØ TØ EXIT PATH
       ,LDA     ,$PS+7
       ,JMP     ,*+17          GØ TØ EXIT PATH
       ,ZERØ    ,03            ZERØ TØ A,B
       ,JMP     ,*+15          EXIT PATH
       ,LASR    ,NBIT-9            MANTISSA TØ BR
       ,CPA     ,              CCA)= EXPØN, -16  A -VE VALUE
*                               A NØW CØNTAINS THE NØ ØF FRACTIØNAL BITS
*                               PRESENT IN THE B REG,
       ,ØRA     ,$PS+8        ASRB
       ,STA     ,*+1
       ,PZE     ,**0           LØCATIØN MØD, BY PRØGRAM,
       ,TBA     ,              A=FIXD PT, INTEGER
       ,CPX     ,
       ,JXZ     ,*+4               JUMP IF ARG PØSITIVE
       ,CPA     ,              NEGATE
       ,IAR     ,
       ,TZB     ,              0 TØ B
       ,RØF     ,              FLAG FØR NØRMAL RETURN
       ,LDX     ,$PS-1         FETCH STØRE ADDRESS
       ,STA     ,0,1           SAVE RESULT
       ,LDX     ,$PS+3         RESTØRE X
       ,JMP     ,0
```

```
        ,ØRG     ,*-1
$HS     ,ENTRY   ,
        ,CALL    ,$SE,1,0
        ,STX     ,*+3
        ,IMP     ,BPS
        ,        ,1
        ,DA      ,0200,03600,0100,0100000
2       ,ASRB    ,
        ,MØRE    ,
        ,END     ,
```

```
*       FLØATING PØINT MATH   16 BIT, MUL/DIV
*
*  IABS            CØMPUTES ABSØLUTE VALUE ØF FIXED PØINT NØ.IN A REG.
*
IABS  ,ENTR    ,
      ,CALL    ,$SE,1,0
      ,LDXI    ,(IABS+4)
      ,LDX     ,0,1
      ,LDA     ,0,1
      ,JAP*    ,IABS        IF PØSITIVE EXIT
      ,CPA     ,            CØMPLEMENT AND
      ,IAR     ,            TWØ'S CØMPLEMENT
      ,JMP*    ,IABS        EXIT.
      ,MØRE    ,
      ,END     ,
```

```
*       FLØATING PØINT MATH   16 BIT, MUL/DIV
*
*  ABS              CØMPUTES ABSØLUTE VALUE ØF FLØATING PØINT NUMBER
*
ABS     .ENTR    ,
        ,CALL    ,$SE,1,10
        ,LDXI    ,(ABS+4)
        ,LDX     ,0,1
        ,LDA     ,0,1
        ,LDB     ,1,1
        ,JAP*    ,ABS
        ,CPA     ,          CØMPLEMENT HIGH ØRDER WØRD
        ,JMP*    ,ABS       AND EXIT.
        ,MØRE    ,
        ,END     ,
```

```
*       FLØATING PØINT MATH   16 BIT, MUL/DIV
*
*  ISIG                    SETS SIGN ØF FIXED PT, NØ EQUAL TØ SIGN
*                          ØF NØ, SPECIFIED IN CALLING SEQUENCE.
*
*
        ,JAP     ,*+4
        ,CPA     ,
        ,IAR     ,
        ,STA     ,ISIGN+8
        ,STX     ,ISIGN+7
        ,LDX     ,ISIGN+4
        ,LDA     ,0,1
        ,ASRA    ,NBIT-1
        ,ERA     ,ISIGN+8
        ,JAP     ,*+3
        ,IAR     ,
        ,LDX     ,ISIGN+7
        ,JMP     ,0
        ,ØRG     ,*-1
ISIGN   ,ENTR    ,
        ,CALL    ,$SE,1
        ,DATA    ,0
        ,JMP     ,*-20
        ,DATA    ,0,0
        ,MØRE    ,
        ,END     ,
```

```
*       FLØATING PØINT MATH   16 BIT, MUL/DIV
*
*  SIGN                    SETS SIGN ØF INPUT PARAMETER EQUAL TØ SIGN
*                          ØF SPECIFIED NØ IN CALL. SEQ . FLØATING PT.
*
        ,JAP     ,*+3       IF NØ NEGATIVE
        ,CPA     ,              CØMPLEMENT AND
        ,STA     ,SIGN+8
        ,STX     ,SIGN+7    SAVE XR
        ,LDX     ,SIGN+4
        ,LDA     ,0,1       (A)= THE SECØND NØ.
        ,ASRA    ,NBIT-1
        ,ERA     ,SIGN+8
        ,LDX     ,SIGN+7    RESTØRE XR
        ,JMP     ,0
        ,ØRG     ,*-1
SIGN    ,ENTR    ,
        ,CALL    ,$SE,1
        ,DATA    ,0
        ,JMP     ,*-16
        ,DATA    ,0,0
        ,MØRE    ,
        ,END     ,
```

```
*       FLØATING PØINT MATH   16 BIT, MUL/DIV
*
*
*  $HM                        INTEGER MULTIPLY (HARDWARE)
*
        ,LDX     ,$HM+4
        ,LLSR    ,NBIT          SCALE MULTIPLICAND
        ,MUL     ,0,1           MULTIPLY
        ,LASL    ,NBIT-1        SCALE PRØDUCT
        ,JMP     ,0             RETURN
        ,ØRG     ,*-1
$HM     ,ENTRY   ,              ENTRY
        ,CALL    ,$SE,1,0       GET ADDR ØF MULTIPLIER
        ,JMP     ,*-10
        ,MØRE    ,
        ,END     ,
```

```
*       FLØATING PØINT MATH   16 BIT, MUL/DIV
*
*
*  $HN                          INTEGER DIVIDE (HARDWARE)
        ,LDX     ,$HN+4             GET DIVISØR
        ,LDB     ,0,1
        ,STB     ,$HN+4             AND SAVE
        ,[illegible]                SIGN IND = PØS
        ,[illegible]                JUMP IF DIVIDEND PØS
        ,CPA     ,                  [illegible] DIVIDEND PØS
        ,IAR     ,
        ,IXR     ,                  SET SIGN IND NEG
        ,LASR    ,NBIT-1            SCALE DIVIDEND
        ,DIV     ,$HN+4             DIVIDE
        ,JXZ*    ,$HN               RETURN IF PØS DIVIDEND
        ,CPA     ,                  ELSE INVERT QUØTIENT
        ,IAR     ,
        ,TBA     ,                  AR = QUØTIENT
        ,JMP     ,0                 RETURN
        ,ØRG     ,*-1
$HN     ,ENTRY   ,                  ENTRY
        ,CALL    ,$SE,1,0           GET ADDR ØF DIVISØR
        ,JMP     ,*-22
        ,MØRE    ,
        ,END     ,
```

```
*       FLØATING PØINT MATH   16 BIT, MUL/DIV
*
*       $SE     SUBPRØGRAM ENTRY CØNTRØL
*
*       $ER     ERRØR
*
SE1     ,STA    ,A              SAVE A,B,X
        ,STB    ,B
        ,STX    ,X
        ,LDA    ,$SE
        ,SUBI   ,3
        ,TAB    ,               BR=ADDR ØF SUBPGM ENTRY
        ,LDA    ,0,2            GET ADDR ØF 1ST PARAM
        ,STA    ,T              AND SAVE
        ,ADD    ,3,2
        ,STA    ,0,2            SET EXIT ADDR TØ MAIN PGM
        ,LDX    ,$SE
        ,INCR   ,045            XR=ADDR ØF DUMMY PARAM
        ,ADD    ,3,2
        ,STA    ,$SE            SET EXIT ADDR TØ SUB-PGM
SE2     ,LDA    ,T              GET ADDR ØF PARAM
        ,TAB    ,
        ,LDA    ,0,2            GET ADDR ØF ITEM
        ,JAN    ,*-2            JUMP IF INDIRECT
        ,STA    ,0,1            ELSE STØRE AT DUMMY
        ,INCR   ,045            LØØK AT NEXT DUMMY
        ,SUB    ,$SE            TEST DUMMY FILLED
        ,INR    ,T              LØØK AT NEXT PARAM
        ,JAN    ,SE2            JUMP IF MØRE PARAMS
        ,LDA    ,A              RESTØRE A,B,X
        ,LDB    ,B
        ,LDX    ,X
        ,JMP*   ,$SE            RETURN TØ SUBPGM
        ,ØRG    ,*-1
$SE     ,ENTR   ,               ENTRY
        ,JMP    ,SE1            ALLØW REL FØRWARD ACCESS
$ER     ,ENTR   ,
        ,ØRA    ,E              SET ERRØR BIT
        ,STA    ,E
        ,JMP*   ,$ER
A       ,DATA   ,0              AR - STØP/PAUSE NØ.
B       ,DATA   ,0              BR - STØP/PAUSE FLAG
```

```
X       ,DATA   ,0          XR
E       ,DATA   ,0          ERRØR BITS
T       ,DATA   ,0          TEMP CELL
        ,MØRE   ,
        ,END    ,
```