

FLOATING ARITHMETIC ROUTINES 2/24/67 (FARITH,14)

/APM 8=64, MOD SRW 12=65

```

FPOVFL,   IDX I (USERPC) /FLOATING OVERFLOW
          DAP I (FOV+1) /SETUP FOV AREA
          LAC CJ
          DIP I (FOV+1)
          LAC (ADD FOV)
          DAC I (USERPC)
          C16RET

```

```

FPNORM,   DAP H           /NORMALIZE 36-BIT MANTISSA IN G AND IO
          DCM F           /SHIFT COUNTER
          CLL             /ROUTINE SETS LINK FOR +@= 0

```

```

FPNOR1,   LAC G           /HI MANTISSA
FPNOR2,   ADD B1          /((2000000)
          SPQ             /SKIPS IF AC WASN'T NORMALIZED
          JMP HEXIT       /EXIT WITH NORMALIZED MANTISSA IN G AND IO
          LAC G           /SETUP FOR NORMALIZING SHIFT
          SCL 1S
          DAC G
          IDX F           /INCREMENT DECREMENT
          SAS (36,)       /DO WE HAVE +@= 0
          JMP FPNOR1      /NO, GO AROUND AGAIN
          CLL+UCML        /+@= 0 INDICATOR
          JMP HEXIT

```

/FLOATING ROUND SUBROUTINE

```

FROUND,   DAP H           /MANTISSA IN G AND IO
          SEL             /+@= 0?
          JMP H           /YES, RETURN, H HAS A JMP FROM FPNORM
          LAW I 1         /DECREMENT THE DECREMENT TO THE CHARACTERISTIC
          ADD F
          DAC F
          LAC G           /UN-NORMALIZE TO PROTECT SIGN BIT
          SCR 1S
          SWP+UCLF 1
          SPI
          CHA+UCMI+USTF 1
          TAD B11         /ROUNDING ADD
          SWP
          TAD CZ          /ADD IN ANY POSSIBLE CARRY
          SZF 1
          CHA+UCMI
          DAC G           /SET UP TO JUMP INTO FPNORM
          JMP FPNOR2      /TO RE-NORMALIZE

```

/FLAG 10T

```

FLAG,     JSP TRACE
          LIC I A
          IDX A
          LAC I A
          DIO I (FAC)
          DAC I (FAC+1)
          JMP R1

```

/FDAC IOT

```

FDAC,      JSP TRACE
           LAC I (FAC)
           DAC I A
           IDX A
           LAC I (FAC+1)
           DAC I A
           JMP R1

```

/FLOAT IOT

```

FLOAT,     LAC I (USERAC) /GET NUMBER TO BE FLOATED
           SCR 98         /CONVERT TO 36-BIT UN-NORM. MANTISSA
           SCR 98
FPFLT2,    DAC G          /SET UP NORMALIZE SUBROUTINE
           IDX I (USERPC) /GET PTR. TO BINARY POINT SPEC.
           DAC A
           LAW 35,        /CHARACTERISTIC
           ADD I A
           DAC C          /STORE CHARACTERISTIC IN C, LIKE FADD DOES
           JMP FADD4      /GO NORMALIZE AND PUT THE NUMBER TOGETHER

```

/FLOAT2 IOT

```

FLOAT2,    LAC I (USERAC) /GET THE NUMBER TO BE FLOATED
           LID I (USERIO) /AS A 36 BIT MANTISSA
           JMP FPFLT2     /GO HANDLE IT JUST LIKE FLOAT1 DOES

```

/FIX IOT

```

FIX,       JSP FPFIX      /CONVERT TO FIXED POINT MANTISSA
           ADD C2         /TEST AC FOR +0-0
           SZA           /IS IT?
           JMP FPROVFL    /NO, NUMBER WON'T FIT INTO AC, OVERFLOW.
           LAI           /SIGN BITS OF BOTH HALVES MUST ALSO AGREE
           XOR C
           SPA           /DO THEY?
           JMP FPROVFL    /NO, NUMBER NEEDS 19 BITS, OVERFLOW.
           DID I (USERAC) /PUT LOW ORDER PART IN AC
           JMP R1

```

/FIX2 IOT

```

FIX2,     JSP FPFIX
           DID I (USERIO) /GIVE IT TO USER
           DAC I (USERAC)
           JMP R1

```

/SUBROUTINE TO GET A FIXED POINT MANTISSA

```

FFFIX,   DAP A
          LAC CJ
          DIP A           /SETUP EXIT
          IDX I (USERPC) /GET PTR. TO BINARY PT. SPECS;
          DAC B           /B POINTS AT USER'S CORE
          LAC I (FAC+1) /GET CHARACTERISTIC OF FAC ALONE
          RAR 8S
          SAR 9S
          SAP 1S
          ADD I B         /ADD IN USER'S BINARY POINT SPECIFICATION
          SUP (36,)       /SHIFT BIN. PT. TO FAR RIGHT AND 1 EXTRA
          /FOR FIRST ISP AT FFFIX2+1
          SRA             /TEST CONVERTED CHARACTERISTIC
          JMP FPROVL      /TOO BIG FOR A 36-BIT INTEGER; OVERFLOW,
          DAC B           /SAVE NUMBER TO ISP ON LATER
          LAC I (FAC)     /GET 36-BIT MANTISSA
          LID I (FAC+1)
          SCR 8S          /GET RID OF CHARACTERISTIC
          SQL 8S
          JMP FFFIX2      /ISP FIRST IN CASE B CONTAINS =1
FFFIX1,   LAC C           /UN-NORMALIZING LOOP
          SCR 1S
FFFIX2,   DAC C
          ISP B           /DONE UN-NORMALIZING?
          JMP FFFIX1      /NO, GO AROUND AGAIN;
          LAC C           /SET UP FOR RETURN
          SAS (=0)        /IS HIGH ORDER HALF MINUS ZERO?
          JMP A           /NO, RETURN RIGHT AWAY
          SWP             /YES, GET LOW ORDER HALF IN AC FOR TESTING,
          SAC (=0)        /IS LOW ORDER HALF MINUS ZERO ALSO?
          CLA+UCL         /YES, BOTH HALVES ARE =0; MAKE THEM +0.
          SWP             /GET THE HALVES BACK IN NORMAL ORDER
          DAC C
          JMP A           /RETURN WITH INTEGER IN AC AND IO

```

```

/FADD AND FSUB IOTS
FADD,    JSP TRACE      /FLAG 1 SET FOR FSUB
        LIO I A         /GET THE ADDEND
        IOX A
        LAC I A         /GET 18-BIT CHARACTERISTIC ALONE
        RAP 8S
        SAR 9S
        SAR 1S
        ADD B17         /A 1-BIT UN=NORM,ING SFT IS DONE LATER
        DAC C
        LAC I A         /GET 36-BIT MANTISSA ALONE
        SWP
        SCR 8S          /GET RID OF CHARACTERISTIC
        SCL 7S          /1 BIT UN=NORM,ING SHIFT
        SZF 1           /FSUB?
        CIA+UCMI        /COMPLEMENT ADDEND
        DAC A           /STORE MANTISSA
        DIO B
        LAC I (FAC+1)
        RAP 8S          /GET 18-BIT CHARACTERISTIC ALONE
        SAR 9S
        SAR 1S
        ADD B17         /A 1-BIT UN=NORM,ING SFT IS DONE LATER
        DAC F
        LAC I (FAC)      /GET 36-BIT MANTISSA ALONE
        LIO I (FAC+1)
        SCR 8S          /GET RID OF CHARACTERISTIC
        SCL 7S          /1-BIT UN=NORMALIZING SHIFT, AS BEFORE
        DAC D
        DIO E           /LEAVES CONTENTS OF E IN I.O., FOR USE LATER
        LAC C           /COMPARE CHARACTERISTICS
        SWP F
        SHA             /IF C<F, SWITCH A,B,C WITH D,E,F
        JMP FADD1        /IF C=F OR C>F, GO AROUND
        LAC A           /SWAP NUMBERS, SEMI-EXEC-MODE CROCK,
        LIO D
        DAC D
        DIO A
        LAC C
        LIO F
        DAC F
        DIO C
        LAC E
        LIO B
        DAC B           /LAEVES WHAT WOULD BE CONTENTS OF E IN IO
        LAC A           /LARGER # IN A,B,C
        SZA             /TEST FOR NORM,ED +0= 0
        SAD (-0)        /POSSIBLE COMPLEMENTED ZERO (FSUB)
        JMP FADD9        /FOUND A PLUS-OR-MINUS ZERO
FADD1,

```

```

FADD2,  LAC F      /UN-NORMALIZING LOOP
        SBC C      /END TEST FOR LOOP, QUIT WHEN C=F,
        JNP FADD3
        IOX F      /COUNT SHIFTS
        LAC D      /SHIFT MANTISSA IN D AND IO RIGHT 1 BIT
        SCR 1S
        DAC D
        JNP FADD2  /GO AROUND AGAIN

FADD3,  LAC D      /GET 36-BIT MANTISSA FROM D AND IO
        SWP        /LINK IS CLEAR
        TAD R      /ADD IN LOW ORDER HALF OF OTHER MANTISSA
        SWP
        TAD A      /ADD IN HIGH ORDER HALF
        S&P
        TAD CR
        SWP
        TAD CE
        DAC G      /SET UP NORMALIZE SUBROUTINE
FADD4,  JSP FPNORM  /GO NORMALIZE 36-BIT MANTISSA
        JSP FROUND  /GO ROUND-OFF TO A 28-BIT MANTISSA
FADD5,  S&L I      /TEST FOR A PLUS-OR-MINUS ZERO MANTISSA
        JNP FADD7
FADD6,  CLA+UCL I   /+0= 0
        JNP FADD8
FADD7,  LAC C      /PUT ZERO IN FLOATING AC AND RETURN TO USER
        SWP F      /CHARACTERISTIC IS IN C
        SCR 7S     /SUBTRACT DECREMENT
        RAL 1S     /GET CHARACTERISTIC INTO LEFT OF IO
        SCR 1S     /PUT SIGN OF CHARACTERISTIC INTO SIGN(IO)
        ADD G      /FOR OVERFLOW TESTING
        SAS C      /AC SHOULD NOW BE +0- 0, GET HI-ORD. MANT.
        JNP FPROVL /WAS AC +0- 0
        RIL 8S     /CHARACTERISTIC WON'T FIT, OVERFLOW
        DAC I (FAC) /GET CHARACTERISTIC ON RIGHT END OF I.O.
FADD8,  DID I (FAC+1) /STORE ANSWER
        JMP R1

/FOUND IT WAS TRYING TO FADD 0 TO A NUMBER WITH 00= CHARACTERISTIC
FADD9,  LAC F      /RETURN WITH NON=0 NUMBER AS ANSWER
        DAC C
        LAC D
        DAC G
        JMP FADD4

```

/FMUL 10T

```

FMUL,   JSP FPSBR1      /CODING SHARED BY FMUL & FDIV
        DIO I (ATEM)    /SAVE HIGH ORDER MANT1
        CLI             /GET LOW MANT1 ALONE AS *INTEGER IN AC
        RGR 8S
        MUL C           /MULTIPLY BY HIGH ORDER MANT2
        SCR 9S          /DIV BY 2+10
        SCR 2S
        DIO H           /SAVE PARTIAL PRODUCT IN H
        LAC D           /GET LOW ORDER MANT2
        MUL I (ATEM)    /MUL BY HIGH MANT1
        SCR 9S          /TRUNCATE LOW ORDER 10 BITS AS BEFORE
        SCR 2S
        DIO G           /SAVE PARTIAL PRODUCT IN G
        LAC I (ATEM)    /MULTIPLY HI ORD PARTS OF MANTS TOGETHER
        MUL C           /THIS YIELDS THE LARGEST PARTIAL PRODUCT
        SCR 1S          /UN=NORM, INC SFT, COMPENSATED FOR LATER
        SWP             /GET LO PARTIAL PRODUCT INTO AC
        TAD G           /ADD IN OTHER PARTIAL PRODUCTS
        SWP
        TAD C2          /((0)
        SWP
        DAC D           /SET UP FPSBR2
        DEM F
        JSP FPSBR2      /FPSBR2 IS MOPE CODING SHARED BY FMUL AND FDIV
        ADD B           /ADD IN CHAR1
        IDA             /1 BIT UN=NORM, INC SFT DONE BEFORE
        DAC C           /SET UP FOR JMP INTO FADD
        LIO H
        JMP FADD5       /GO PUT RESULTS IN FLOATING AC AND RETURN

```

/FDIV 10T

```

FDIV,   JSP FPSBR1      /SET UP FOR DIVIDE
        SWP
        SCR 8S          /GET RID OF CHARACTERISTIC
        SCR 6S          /EFFECTIVE RIGHT SHIFT OF 2 OR 3 BITS
        DIV C           /2+31<=N<2+32, DIVIDED BY 2+16<=C<2+17
        JMP R1          /FIRST RETURN, FOR DIVISION BY ZERO
        DAC G           /YIELDS 2+14<AC<2+16
        IDX I (USERPC) /SETUP FOR 2ND RETURN TO USER
        CLA+USWP        /MULTIPLY REMAINDER BY 2+17
        DIV C           /((R<C)+T2+17), DIVIDED BY 2+16<=C<2+17
        C14, 14         /((12,)
        RAL 1S          /YIELDS AC<2+17
        DAC H           /TACK IT ON AS AN 18-BIT EXTENSION OF G
        LAC D           /COMPUTE CORRECTION TERM
        RAL 6S
        CLI             /DELETABLE IF 10 DOESN'T AFFECT ACCURACY
        DIV C
        C141, 141
        MUL G
        RCL 2S          /CORRECTION TERM IN AC WITH PROPER SCALE
        CHA             /CORRECTION TERM IS NEGATIVE

```

```

LID G          /SET UP FPSBR2
DAC D
CLC           /-0 INTO E, SINCE CORR. TERM IS NEG.
DAC F
JSP FPSBR2
CMA          /SUBTRACT CHARACTERISTICS
ADD B
ADD B16      /CORRECT FOR UN-NORMALIZING SFT OF 2 PLACES
JMP FMUL1    /GO NORMALIZE AND RETURN

FPSBR1, DAP E          /SUBROUTINE SHARED BY FMUL AND FDIV
        LAC CJ
        DIP E          /RETURN IS IN E
        JSP TRACE
        LAC I A        /GET SIGN OF RESULT AND SAVE IN B
        XOR I (FAC)
        DAC B
        LID I A        /GET HIGH ORDER MANT2
        IDX A
        LAC I A        /GET LOW ORDER MANT2
        SPI           /MAKE MANTISSA POSITIVE
        CMA+UCM1
        DIO C          /SAVE HIGH ORDER MANT2 IN C
        CLI           /GET LO ORDER MANT2 ALONE AS A +INTEGER
        RQR BS
        DAC D          /SAVE LOW ORDER MANT2 IN D
        LID I (FAC)    /GET MANT1
        LAC I (FAC+1)
        SPI           /MAKE MANTISSA POSITIVE
        CMA+UCM1
        JMP E          /RETURN

FPSBR2, DAP E          /SUBR SHARED BY FMUL & FDIV
        LAC D          /RESTORE AC PREVIOUS TO JSP
        TAD H          /CONTINUE ADDING PARTIAL RESULTS TOGETHER
        SHP
        TAD F          /ADD IN POSSIBLE CARRY, #0 FOR FDIV
        DAC G          /SET UP NORMALIZE ROUTINE
        JSP FPNORM     /GO NORMALIZE
        JSP FROUND     /ROUND TO A 28-BIT MANTISSA
        LAC B          /RESTORE SIGN OF RESULT
        SPA
        CHI
        DIO H
        LID G
        SPA
        CHI
        DIO G          /RESULT IS NOW IN G AND H WITH PROPER SIGN
        LAC I (FAC+1)  /GET CHAR1 ALONE
        RAR BS
        SAR 9S
        SAR 1S
        DAC R          /SAVE CHAR1 IN R
        LAC I A        /GET CHAR2 ALONE
        RAR BS
        SAR 9S
        SAR 1S

```

JMP E

/RETURN

START

END FARITH,LST [IMPIMP]

CREATED: 01-FEB-1972 14149

NUMBER OF COPIES: 1

TOTAL NUMBER OF PAGES: 0