

FLOATING POINT PACKAGE

A. Introduction

Floating point arithmetic has several advantages for the programmer when compared to fixed point arithmetic.

1. Very large and very small numbers may be efficiently stored.
2. Computations on numbers of widely varying magnitude are as easy as computations of numbers of the same or similar magnitude.
3. It is not necessary to align binary points to perform addition and subtraction.
4. * It is not necessary to retain the zeros (placeholders) between the binary point and the nearest significant digit.

By using the Nicolet Instrument Corporation (NIC) supplied Floating Point package, the programmer has available a series of functional subroutines. These subroutines use common registers; consequently, relatively complex mathematical equations are handled by placing data in the proper registers and calling the subroutines in logical sequence.

If, for example, the following equation was to be solved:

$$Y = A \sin B$$

the programmer calls the subroutines needed to perform the desired functions.

MEMA (B	/get address of data
JMS FGETAC	/place data in floating accumulator
JMS FSIN	/results left in floating AC
MEMA (A	/get address of data

JMS FGETAR /move data to floating argument

JMS FMULT /results left in floating AC

In the example above, the use of the pseudo-registers, the Floating ACc-
mulator (FAC) and the Floating ARgument (FAR), is illustrated. Functions which
operate on a single number (i. e. , a sine), use the data found in FAC. Functions
which require two numbers (i. e. , a multiplication), operate on data stored in
FAC and in FAR.

NOTE: For illustrative purposes, the sample program was slightly
simplified. Normally, user programs are written "off page," consequently
the subroutines would be called using indirect instructions.

B. Available Commands

Routines available can generally be divided into three categories: math,
data routing, and Input/Output (I/O).

Math routines:

FADD	- double precision addition
FSUB	-double precision subtraction
FNEG	-negate the Floating ACcumulator (FAC)
FMULT	-single precision multiplication
FDIV	-single precision division
FSQAR	-single precision square of FAC
FSQRT	-single precision square root of FAC
FSIN	-sine of FAC
FCOS	-cosine of FAC

FARCTN	-arctangent of FAC
FEXPO	-raise e to the power of the contents of the FAC
FLOG	-find the natural log of the contents of the FAC
FNOR	-normalize the data in the FAC

Data routing:

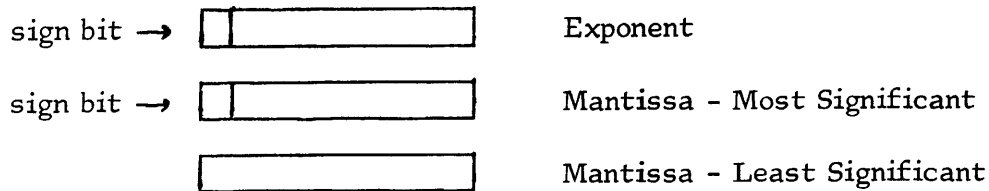
FPUTAC	-move data from the FAC
FGETAC	-move data to the FAC
FGETAR	-move data to the Floating ARgument (FAR)
FACFAR	-move data from FAC to FAR

I/O routines include:

FLOINP	-input data from teletype keyboard
OUTFLO	-output data on teletype printer

C. Format

1. FAC and FAR are registers each composed of three words: the first containing the exponent, the second containing the most significant part of the mantissa and the sign, and the third word containing the least significant part of the mantissa.



2. FAC and FAR are assigned permanent locations in core storage.

3. Routines involving one number (square, sine, etc.) operate on data located in FAC.

4. Routines involving two numbers (addition, division, etc.) operate on data located in FAC and FAR.
5. The results of the above operations are deposited in FAC.
6. All math routines assume data to be in normalized form upon entry.
7. Mathematical operations exit with the results in normalized form.
8. In normalized form, $1/2 \leq |\text{mantissa}| < 1$.
9. The binary point is implied immediately to the right of the sign bit of the mantissa.
10. An accuracy of 5.5 decimal digits is maintained throughout all operations including input and output. The exponent may range within approximately $\pm 300,000_{10}$.

An example of the conversion of a decimal fixed point number to a normalized binary floating point number follows.

First, a logical step is to convert the decimal fixed point number 11.53 (assumed to be an exact number) to an octal fixed point number. The second step in this example is to convert the octal number into a fixed point binary number.

$$\begin{aligned}
 11.53_{10} &= 1 \quad 3 \quad . \quad 4 \quad 1 \quad 7 \quad 2 \quad 7 \quad 0 \quad 2 \quad 4 \quad 3 \quad 6 \quad 5 \quad 6_8 \\
 &= 1 \quad 011 \quad . \quad 100 \quad 001 \quad 111 \quad 010 \quad 111 \quad 000 \quad 010 \quad 100 \quad 001 \quad 110 \quad 101 \quad 110_2
 \end{aligned}$$

Next, the number is changed from fixed point to floating point by moving the binary point four places to the left and assigning the exponent a value of four.

$$= (2^4) \cdot 1\ 011\ 100\ 001\ 111\ 010\ 111\ 000\ 010\ 100\ 001\ 110\ 101\ 110$$

Lastly, to normalize the above floating point number into the form used by the Floating Point Package, the sign bit must be accounted for. Therefore, the number 11.53_{10} would properly be found in FAC or FAR as:

	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0	Exponent
sign bit →	0 1 0 1 1 1 0 0 0 0 1 1 1 0 1 0 1 1 1	Upper part of mantissa
	0 0 0 0 1 0 1 0 0 0 0 1 1 1 0 1 0 1 1 1	Lower part of mantissa

D. Description of Routines

Some of the routines below are described in short form. For example, FADD adds the contents of FAC to the contents of FAR and deposits the sum in FAC.

1. Math Routines

FADD:	$C(FAC) + C(FAR) \rightarrow C(FAC)$	double precision
FSUB:	$C(FAC) - C(FAR) \rightarrow C(FAC)$	
FNEG:	$C(FAC) \rightarrow -C(FAC)$ C(FAR) preserved	
FMULT:	$C(FAC) \times C(FAR) \rightarrow C(FAC)$ C(FAR) preserved	
FDIV:	$C(FAC) \div C(FAR) \rightarrow C(FAC)$; $1 \rightarrow C(ERRDIV)$; $\emptyset \rightarrow C(FAC)$ if division by zero C(FAR) preserved	
FSQAR:	$C(FAC)^2 \rightarrow C(FAC)$	
FSQRT:	$C(FAC)^{1/2} \rightarrow C(FAC)$ $1 \rightarrow C(ERRSQT)$ for negative number C(TEMP) destroyed	

FSIN:	$\sin C(\text{FAC}) \rightarrow C(\text{FAC})$	)	
		)	argument in
FCOS:	$\cos C(\text{FAC}) \rightarrow C(\text{FAC})$	)	
		)	radians $\div \frac{\text{PI}}{2}$
FARCTN:	$\arctan (C(\text{FAC})) \rightarrow C(\text{FAC})$		results in radians $\div \frac{\text{PI}}{2}$
FEXPO:	$e^{C(\text{FAC})} \rightarrow C(\text{FAC})$		
FLOG:	$\log C(\text{FAC}) \rightarrow C(\text{FAC})$	)	base e

C(FAR) destroyed except where noted.

FNOR: $C(\text{FAC})_{\text{normalized}} - C(\text{FAC})$ (double precision)

2. Data Routing

FPUTAC X:	$C(FAC) \rightarrow C(X)$	)	
		)	
FGETAC X:	$C(X) \rightarrow C(FAC)$	)	X specifies the location of data and
		)	must be in the accumulator when
FGETAR X:	$C(X) \rightarrow C(FAR)$	)	entering the subroutine.
		)	
FACFAR:	$C(FAC) \rightarrow C(FAR)$	)	

3. Input/Output Routines

FLOINP: The decimal number typed on the teletype keyboard is converted to a floating point binary number and deposited on FAC. Any of the following formats are acceptable.

19.

1.9 E 1

1900 E-2

Etc.

Input is terminated by typing a Carriage Return (CR).

If an error is made in typing, the line can be restarted by typing a "rubout" before the line is terminated, i. e. , 715 rubout 3.29. would be interpreted as 3.29.

OUTFLO: The C(FAC) is printed in decimal floating point format.

E. Physical Dimensions of Program

The Floating Point package comes in two versions. The basic package contains all except the teletype Input/Output routines. This program occupies addresses 6000₈ through 7577₈.

The complete package, including FLOINP and OUTFLO, occupies addresses 5350₈ through 7577₈.

F. Location of Subroutines

All address locations given in octal.

FADD:	6610	FPUTAC:	7264	FLOINP:	5700
FSUB:	6671	FGETAC:	7230	OUTFLO:	5350
FNEG:	7103	FGETAR:	7246		
FMULT:	7057	FACFAR:	7200		
FDIV:	7020				
FSQAR:	6567				
FSQRT:	6520				
FSIN:	6271				
FCOS:	6351				
FARCTN:	6140				
FEXPO:	6000				

FLOG: 6357

FNOR: 7114

G. Location of Flags and Registers

ERRSQT: 7562 flag for square root of negative number

ERRDIV: 7463 flag for division by zero

FAR: 7570 FARE floating argument - composed of
FARM 3 addresses
FARML

FAC: 7574 FACE floating accumulator - composed of
FACM 3 addresses
FACML

H. Miscellaneous Routines

The Floating Point package contains other subroutines which may be useful.

FACTEM: C(FAC) → C(TEM)

TEMFAC: C(TEM) → C(FAC)

TEM is a register similar to FAC and FAR where data may be temporarily stored. Note that FACTEM and TEMFAC are used internally by the Floating Point package.

Four routines are available for conversion of numbers from fixed point to floating point or floating point to fixed point. The four routines are composed of two sets of matched pairs.

FIXFLO - operates on the fixed point number found in the accumulator (AC);

converts the number into floating point notation and leaves the result
in the Floating ACcumulator (FAC).

FLOFIX - reverses the effect of FIXFLO. The floating point number found

in FAC is converted to fixed point notation with the result left in the AC.

In FIXFLO and FLOFIX, the fixed point number is assumed to be a number less than 1. In other words, bit 19 contains the sign and the binary point is implied between bits 18 and 19.

FLOAT and FIX are subroutines written to handle fixed point numbers which may be larger than 1.

FLOAT - operates on a fixed point number found in FAC. FACML contains the fractional part of the number (the binary point implied to the left of bit 19 - the most significant bit of FACML) while the integer portion of the fixed point number is contained in FACM. The results are left in FAC.

FIX - reverses the effect of FLOAT. A floating point number found in FAC is converted to a fixed point number with the number left in FAC. The binary point will be implied between FACM and FACML and the sign bit is bit 19 of FACM.

Two significant differences occur between the pairs of subroutines. FIXFLO and FLOFIX move fractional numbers between the AC and the FAC. FLOAT and FIX require data to be in the FAC; however, data may be larger than 1.

Locations of the routines and registers are:

FACTEM:	7210	FIXFLO:	7303
TEMFAC:	7220	FLOFIX:	7311
TEM:	7564	FLOAT:	7323
		FIX:	7330

/INPUT-OUTPUT MODULE

/REQUIRES BASIC ARITHMETIC MODULE AND MOVE MODULE

*5350

```

5350      0   OUTFLO, 0 /OUTPUT FAC IN "E" FORMAT
5351 3111764 MEMA @ FACM /DESTROYS FAC AND FAR
5352      5104 SKIP AC19
5353      1357 JMP PMAN
5354 3065764 MNGM @ FACM
5355      110255 MEMA (255
5356      162000 ZERZ
5357      110253 PMAN, MEMA (253
5360 2001745 JMS PCHAR /PRINT SIGN
5361      110256 MEMA (256
5362 2001745 JMS PCHAR /PRINT PERIOD
5363 2165555 ZERM DEXP /BEGIN MATH
5364 3711763 FGO2, MMOA @ FACE
5365      5104 SKIP AC19
5366      1375 JMP FGO3 /FACE TOO LARGE
5367      510004 A+MA (4
5370      5104 SKIP AC19
5371      1400 JMP FGO4 /-1 <FACE<-4
5372 2001574 JMS MPYTEN /MULTIPLY BY TEN
5373 2705555 MMOM DEXP
5374      1364 JMP FGO2
5375 2001601 FGO3, JMS MTENTH
5376 2125555 MPOM DEXP
5377      1364 JMP FGO2
5400 3111764 FGO4, MEMA @ FACM /NORMALIZE
5401      5001 LASH 1
5402 3405764 ACCM @ FACM
5403 3705763 MMOM @ FACE
5404 3165765 ZERM @ FACML
5405      1417 JMP NORMA
5406 3111764 D2LOOP, MEMA @ FACM
5407      5061 RLSH 1
5410      5201 TACL
5411 2011447 ANDA K1777777
5412 3405764 ACCM @ FACM
5413 3111765 MEMA @ FACML
5414      5061 RLSH 1
5415      5202 TLAC
5416 3405765 ACCM @ FACML
5417 3127763 NORMA, MPOMZ @ FACE
5420      1406 JMP D2LOOP

5421 2165446 ZERM MANLZS
5422      70006 RESTART, MNGA (6 /OUTPUT MANTISSA
5423 2405557 ACCM CHRCNT
5424 2001450 OMLoop, JMS TTEN /MULTIPLY FAC BY TEN
5425 2111556 MEMA DDIGIT
5426 2511446 A+MA MANLZS
5427 2025446 ONEM MANLZS /FLAG SET
5430      402000 ACCZ

```

5431	1434	JMP CONT1
5432	2705555	MMOM DEXP
5433	1422	JMP RESTART
5434	2111556	CONT1, MEMA DDIGIT
5435	2001754	JMS PDECN
5436	2127557	MPOMZ CHRCNT
5437	1424	JMP OML00P
5440	110305	MEMA (305 /OUTPUT EXPONENT
5441	2001745	JMS PCHAR /PRINT "E"
5442	2025573	ONEM SUPFLA /SET ZERO SUPPRESS FLAG
5443	2111555	MEMA DEXP
5444	2001472	JMS OUTINT
5445	1001350	JMP @OUTFLO
5446	0	MANLZS, 0 /LEADING ZERO SUPPRESION IN MANTISSA
5447	1777777	K1777777, 1777777
5450	0	TEN, 0 /MULTIPLY FAC BY TEN, INTEGER
5451	2165556	ZERM DDIGIT /DDIGIT CONTAINS NUMBER OF OVERFLOWS
5452	3001771	JMS @ FACFAR
5453	70011	MNGA (11
5454	2405554	ACCM CNTR
5455	5210	MPTENL, CLL
5456	3111762	MEMA @ FARML
5457	3515765	A+MAM @ FACML
5460	3111761	MEMA @ FARM
5461	5141	EXCT L
5462	430000	APOA
5463	5210	CLL
5464	3515764	A+MAM @ FACM
5465	5141	EXCT L
5466	2125556	MPOM DDIGIT
5467	2127554	MPOMZ CNTR
5470	1455	JMP MPTENL
5471	1001450	JMP @TEN
5472	0	OUTINT, 0 /OUTPUT SIGNED DECIMAL INTEGER
5473	2405555	ACCM DEXP
5474	2165556	ZERM DDIGIT
5475	5104	SKIP AC19
5476	1502	JMP PEXP
5477	2065555	MNGM DEXP
5500	110255	MEMA (255
5501	162000	ZERZ
5502	110253	PEXP, MEMA (253
5503	2001745	JMS PCHAR /PRINT SIGN
5504	2111552	SUBHTH, MEMA DK100000
5505	2335555	M-AAM DEXP
5506	2125556	MPOM DDIGIT
5507	5104	SKIP AC19
5510	1504	JMP SUBHTH
5511	2111552	MEMA DK100000
5512	2001560	JMS PDECNE
5513	2111553	SUBTTH, MEMA DKK10000
5514	2335555	M-AAM DEXP
5515	2125556	MPOM DDIGIT
5516	5104	SKIP AC19

5517	1513	JMP SUBTTH
5520	2111553	MEMA DKK10000
5521	2001560	JMS PDECNE
5522	111750	SUBTH0, MEMA (1750
5523	2335555	M-AAM DEXP
5524	2125556	MPOM DDIGIT
5525	5104	SKIP AC19
5526	1522	JMP SUBTH0
5527	111750	MEMA (1750
5530	2001560	JMS PDECNE
5531	110144	SUBHUN, MEMA (144
5532	2335555	M-AAM DEXP
5533	2125556	MPOM DDIGIT
5534	5104	SKIP AC19
5535	1531	JMP SUBHUN
5536	110144	MEMA (144
5537	2001560	JMS PDECNE
5540	110012	SUBTEN, MEMA (12
5541	2335555	M-AAM DEXP
5542	2125556	MPOM DDIGIT
5543	5104	SKIP AC19
5544	1540	JMP SUBTEN
5545	110012	MEMA (12
5546	2001560	JMS PDECNE
5547	2111555	MEMA DEXP
5550	2001754	JMS PDECN
5551	1001472	JMP @OUTINT
5552	303240	DKK100000, 303240
5553	23420	DKK10000, 23420
5554	0	CNTR, 0
5555	0	DEXP,0 /DECIMAL EXPONENT
5556	0	DDIGIT,0 /DECIMAL DIGIT
5557	0	CHRCNT,0 /CHARACTER COUNTER
5560	0	PDECNE, 0 /SUPPRESS LEADING ZEROES
5561	2505555	A+MM DEXP
5562	2711556	MMOA DDIGIT
5563	2165556	ZERM DDIGIT
5564	402000	ACCZ
5565	1570	JMP PGO
5566	2103573	MEMZ SUPFLA
5567	1001560	JMP @PDECNE
5570	2165573	PGO, ZERM SUPFLA
5571	2001754	JMS PDECN
5572	1001560	JMP @PDECNE
5573	0	SUPFLA, 0
5574	0	MPYTEN, 0 /MULTIPLY BY TEN, FLOATING POINT
5575	111606	MEMA (KTEN
5576	3001772	JMS @ FGETAR
5577	3001766	JMS @ FMULT
5600	1001574	JMP @MPYTEN
5601	0	MTENTH, 0 /MULTIPLY BY .1, FLOATING POINT
5602	111611	MEMA (KTENTH
5603	3001772	JMS @ FGETAR

5604	3001766	JMS @ FMULT
5605	1001601	JMP @MTENTH
5606	4	KTEN, 4
5607	1200000	1200000
5610	0	0
5611	3777775	KTENTH, 3777775
5612	1463147	1463147
5613	0	0
5614	0	RSINT, 0 /INPUT SIGNED DECIMAL INTEGER
5615	2165557	ZERM CHRCNT
5616	3165763	ZERM @ FACE
5617	3165764	ZERM @ FACM
5620	3165765	ZERM @ FACML
5621	2165675	ZERM ERRORF
5622	2165676	ZERM ISIGNF
5623	2025672	ONEM PERSW
5624	2001737	FETCH1, JMS RCHAR /FETCH ASCII CHARACTER
5625	462305	A-MZ (305 /"E"?
5626	2167674	ZERMZ ESWIT
5627	2025674	ONEM ESWIT
5630	462377	A-MZ (377 /RUBOUT?
5631	162000	ZERZ
5632	2025675	ONEM ERRORF
5633	462255	A-MZ (255 /"-"?
5634	1637	JMP FETCP
5635	2025676	ONEM ISIGNF
5636	1624	JMP FETCH1
5637	462253	FETCP, A-MZ (253 /"+"
5640	1642	JMP FETCH2
5641	1624	JMP FETCH1
5642	462256	FETCH2, A-MZ (256 /"."?
5643	1646	JMP FETCH3
5644	2165672	ZERM PERSW
5645	1624	JMP FETCH1
5646	470260	FETCH3, A-MA (260 /IS IT A DECIMAL NUMBER?
5647	5144	EXCT AC19
5650	1667	JMP RSINT1 /NO
5651	470012	A-MA (12
5652	5104	SKIP AC19
5653	1667	JMP RSINT1 /NO
5654	510012	A+MA (12 /YES
5655	2405673	ACCM TEMP
5656	2703672	MMOZ PERSW
5657	2125557	MPOM CHRCNT
5660	2001450	JMS TTEN /MULTIPLY FAC BY TEN, INTEGER
5661	3111764	MEMA @FACM
5662	2003677	ANDZ MK31
5663	2025675	ONEM ERRORF
5664	2111673	MEMA TEMP
5665	3505765	A+MM @ FACML
5666	1624	JMP FETCH1
5667	2103676	RSINT1, MEMZ ISIGNF
5670	3001767	JMS @ FNEC
5671	1001614	JMP ORSINT
5672	0	PERSW, 0 /PERIOD SWITCH, COUNTS NUMBER OF DIGITS

/AFTER "."

```

5673      0   TEMP, 0
5674      0   ESWIT, 0 /"E" SWITCH
5675      0   ERRORF, 0 /ERROR FLAG
5676      0   ISIGNF, 0 /INTEGER SIGN FLAG
5677 31000000 MK31, 31000000

5700      0   FLOINP, 0 /INPUT A FLOATING NUMBER
5701 2001614  RSTART, JMS RSINT /DESTROYS TEMP, FAC AND FAR
5702 3001770  JMS @ FNOR
5703 110047   MEMA (47
5704 3505763  A+MM @ FACE
5705 3001773  JMS @ FACTEM /SAVE FAC
5706 2111557  MEMA CHRCNT
5707 2225736  ANGM TDEXP
5710 2703674  MMOZ ESWIT /GET EXPONENT?
5711      1715  JMP COMPEN
5712 2001614  JMS RSINT /YES
5713 3111765  MEMA @ FACML
5714 2505736  A+MM TDEXP
5715 3001774  COMPEN, JMS @ TEMFAC /COMPENSATE FOR DECIMAL EXPONENT
5716 2111736  COMPI, MEMA TDEXP
5717      5104  SKIP AC19
5720      1724  JMP COMP2
5721 2001601  JMS MTENTH
5722 2125736  MPOM TDEXP
5723      1716  JMP COMPI
5724 405160   COMP2, EXCT ZAC
5725      1731  JMP FIPEXT /EXIT
5726 2001574  JMS MPYTEN
5727 2705736  MMOM TDEXP
5730      1716  JMP COMPI
5731 2703675  FIPEXT, MMOZ ERRORF
5732 1001700  JMP @FLOINP
5733 110334   MEMA (334
5734 2001745  JMS PCHAR
5735      1701  JMP RSTART
5736      0   TDEXP, 0

5737      0   RCHAR, 0 /READ CHARACTER FROM TTY
5740      6454  RCHAR1, TTYRF
5741      1740  JMP RCHAR1
5742      44453  RDTTY
5743 2001745  JMS PCHAR
5744 1001737  JMP @RCHAR

5745      0   PCHAR, 0 /PRINT CHARACTER
5746      6444  WAIT, TTYPEF
5747      1746  JMP WAIT
5750      4443  PRTTY
5751 2125753  MPOM PCNTR
5752 1001745  JMP @PCHAR
5753      0   PCNTR, 0 /PRINT CHARACTER COUNTER

5754      0   PDECN, 0 /PRINT DECIMAL DIGIT
5755      510260 A+MA (260
5756 2001745  JMS PCHAR
5757 1001754  JMP @PDECN

```

5760	7570	FAFE, 7570	/FLOATING ARGUMENT
5761	7571	FARM, 7571	
5762	7572	FARML, 7572	
5763	7574	FACE, 7574	/FLOATING ACCUMULATOR
5764	7575	FACM, 7575	
5765	7576	FACML, 7576	
5766	7057	FMULT, 7057	
5767	7103	FNEG, 7103	
5770	7114	FNOR, 7114	
5771	7200	FACFAR, 7200	
5772	7246	FGETAR, 7246	
5773	7210	FACTEM, 7210	
5774	7220	TEMFAC, 7220	

*6000

6000	0	FEXPO, 0/ FLOATING EXPONENTIAL
6001	2164103	ZERM SIGN/ CLEAR SIGN
6002	2111575	MEMA FACM/ IS ARGUMENT NEGATIVE ?
6003	5144	EXCT AC19
6004	2144103	MONM SIGN/ YES SET SIGN
6005	5144	EXCT AC19/ NO-IS ARGUMENT TO BE NEGATED?
6006	2001103	JMS FNEG/ YES
6007	110127	MEMA (LOGE/ NO-MULTIPLY ARGUMENT BY LOG2E
6010	2001246	JMS FGETAR
6011	2001057	JMS FMULT
6012	110104	MEMA (NF/ STORE PRODUCT
6013	2001264	JMS FPUTAC
6014	2001330	JMS FIX/ GET FRACTIONAL PART OF PRODUCT
6015	2165576	ZERM FACML
6016	2111575	MEMA FACM
6017	2404107	ACCM N
6020	2001323	JMS FLOAT
6021	2001200	JMS FACFAR
6022	110104	MEMA (NF
6023	2001230	JMS FGETAC
6024	2000671	JMS FSUB
6025	110104	MEMA (NF/ STORE FRACTIONAL PART OF PRODUCT
6026	2001264	JMS FPUTAC
6027	2001200	JMS FACFAR/ SQUARE FRACTIONAL PART
6030	2001057	JMS FMULT
6031	110110	MEMA (F/ STORE SQUARED FRACTIONAL PART
6032	2001264	JMS FPUTAC
6033	110124	MEMA (D / ADD D
6034	2001246	JMS FGETAR
6035	2000610	JMS FADD
6036	2001200	JMS FACFAR
6037	110121	MEMA (C/ DIVIDE SUM INTO C
6040	2001230	JMS FGETAC
6041	2001020	JMS FDIV
6042	110113	MEMA (A/ ADD A
6043	2001246	JMS FGETAR
6044	2000610	JMS FADD
6045	110104	MEMA (NF/ SUBTRACT FRACTIONAL PART
6046	2001246	JMS FGETAR
6047	2000671	JMS FSUB
6050	2001210	JMS FACTEM/ STORE SUM
6051	110116	MEMA (B/ MULTIPLY B BY FRACTION SQUARED
6052	2001246	JMS FGETAR
6053	110110	MEMA (F
6054	2001230	JMS FGETAC
6055	2001057	JMS FMULT
6056	2001200	JMS FACFAR/ ADD PREVIOUS SUM
6057	2001220	JMS TEMFAC
6060	2000610	JMS FADD
6061	2001200	JMS FACFAR
6062	110104	MEMA (NF/ DIVIDE FRACTIONAL PART BY SUM
6063	2001230	JMS FGETAC
6064	2001020	JMS FDIV

6065	2001200	JMS FACFAR/ MULTIPLY BY TWO
6066	2000610	JMS FADD
6067	110132	MEMA CONE/ ADD ONE
6070	2001246	JMS FGETAR
6071	2000610	JMS FADD
6072	2110107	MEMA N/ MULTIPLY BY TWO RAISED TO INTEGER PART
6073	2505574	A+MM FACE
6074	2126103	MPOMZ SIGN
6075	1000000	JMP @FEXPO
6076	2001200	JMS FACFAR/ DIVIDE ONE BY ARGUMENT FOR MINUS EXPONENT
6077	110132	MEMA CONE
6100	2001230	JMS FGETAC
6101	2001020	JMS FDIV
6102	1000000	JMP @FEXPO
6103	0	SIGN, 0
6104	0	NF, 0
6105	0	0
6106	0	0
6107	0	N, 0
6110	0	F, 0
6111	0	0
6112	0	0
6113	4	A, 4
6114	1175060	1175060
6115	400000	0400000
6116	3777774	B, 3777774
6117	1067646	1067646
6120	3600000	3600000
6121	12	C, 12
6122	2626016	2626016
6123	600000	0600000
6124	7	D, 7
6125	1273256	1273256
6126	200000	0200000
6127	1	LOGE, 1
6130	1342521	1342521
6131	3200000	3200000
6132	1	ONE, 1
6133	1000000	1000000
6134	0	0
*7200		
7200	0	FACFAR, 0
*7210		
7210	0	FACTEM, 0
*7220		
7220	0	TEMFAC, 0
*7230		
7230	0	FGETAC, 0
*7246		
7246	0	FGETAR, 0
*7264		
7264	0	FPUTAC, 0
*6610		
6610	0	FADD, 0
*6671		
6671	0	FSUB, 0
*7020		
7020	0	FDIV, 0

*7057		
7057	0	FMULT, 0
*7103		
7103	0	FNEG, 0
*7330		
7330	0	FIX, 0
*7323		
7323	0	FLOAT, 0
*7574		
7574	0	FACE, 0
7575	0	FACM, 0
7576	0	FACML, 0

/FLOATING POINT ARCTAN
 /EXITS WITH RADIAN/(PI/2)

*6140

6140	0	FARCTN, 0
6141	110247	MEMA (A2
6142	2404507	ACCM TAD
6143	110002	MEMA (2
6144	2404510	ACCM NOM
6145	2111575	MEMA FACM
6146	5104	SKIP AC19
6147	152	JMP PLUS
6150	2001103	JMS FNEG /GET ABSOLUTE VALUE OF FAC
6151	2166104	ZERMZ TFLAG /SET SIGN FLAG
6152	2024104	PLUS, ONEM TFLAG
6153	110110	MEMA (X /STORE X
6154	2404511	ACCM PX
6155	2001264	JMS FPUTAC
6156	110132	MEMA (KK1
6157	2001246	JMS FGETAR
6160	2000671	JMS FSUB /X<= 1
6161	2111575	MEMA FACM
6162	5144	EXCT AC19
6163	175	JMP XEQY /YES
6164	110110	MEMA (X /X>1, INVERT
6165	2001246	JMS FGETAR
6166	110132	MEMA (KK1
6167	2001230	JMS FGETAC /1/X=Y
6170	2001020	JMS FDIV
6171	110110	MEMA (X
6172	2001264	JMS FPUTAC
6173	2164103	ZERM OFLAG /SET FLAG
6174	200	JMP SQ
6175	2024103	XEQY, ONEM OFLAG
6176	110110	MEMA (X
6177	2001230	JMS FGETAC
6200	2000567	SQ, JMS FSQAR /Z=X*2
6201	110105	MEMA (Z
6202	2404512	ACCM PZ
6203	2001264	JMS FPUTAC
6204	2000467	JMS POX /SOLVE NUMERATOR
6205	110241	MEMA (NUM
6206	2001264	JMS FPUTAC
6207	110260	MEMA (B2
6210	2404507	ACCM TAD
6211	110002	MEMA (2
6212	2404510	ACCM NOM /INITIALIZE POX TO SOLVE DENOMINATOR
6213	110132	MEMA (KK1
6214	2404511	ACCM PX /DEGENERATE USE OF X IN POX
6215	110105	MEMA (Z
6216	2001230	JMS FGETAC
6217	2000467	JMS POX
6220	2001200	JMS FACFAR
6221	110241	MEMA (NUM /MAKE DIVISION
6222	2001230	JMS FGETAC
6223	2001020	JMS FDIV

```

6224 2702103 TOUT, MMOZ OFLAG /X>1
6225 162000 ZERZ
6226 233 JMP LTO
6227 2001200 JMS FACFAR
6230 110244 MEMA (PI02
6231 2001230 JMS FGETAC
6232 2000671 JMS FSUB
6233 110244 LTO, MEMA (PI02 /OUTPUT IN RADIANS/(PI/2)
6234 2001246 JMS FGETAR
6235 2001020 JMS FDIV
6236 2702104 MMOZ TFLAG /NEG. INPUT?
6237 2001103 JMS FNEG
6240 1000140 JMP @ FARCTN

```

```

6241 0 NUM, 0
6242 0 0
6243 0 0
6244 1 PI02, 1
6245 1444177 1444177
6246 0 0
6247 3777773 A2, 3777773
6250 1543263 1543263
6251 0 0
6252 3777777 A1, 3777777
6253 1542222 1542222
6254 0 0
6255 0 A0, 0
6256 1217472 1217472
6257 0 0
6260 3777775 B2, 3777775
6261 1613743 1613743
6262 0 0
6263 0 B1, 0
6264 1213515 1213515
6265 0 0
6266 0 B0, 0
6267 1217472 1217472
6270 0 0

```

/FLOATING POINT SINE

/ARGUMENT IN RADIANS/(PI/2)

```

6271 0 FSIN, 0
6272 110332 MEMA (K9 /INITIALIZE POX
6273 2404507 ACCM TAD
6274 110004 MEMA (4
6275 2404510 ACCM NOM
6276 2164104 ZERM NFLAG
6277 2111575 MEMA FACM /MANTISSA OF FAC INTO AC, SINGLE PRECISSION
6300 5104 SKIP AC19 /CHECK IF NUMBER I NEG. SKIP IF NEG.
6301 304 JMP S / JMP TO S IF POS.
6302 2001103 JMS FNEG / MAKE C(FACM) POS
6303 2024104 ONEM NFLAG / 1 INTO NFLAG, IF FACM WAS NEG
6304 2001330 S, JMS FIX / CONVERT INTO FIX POINT
6305 2111575 MEMA FACM
6306 2002 ANDZ (2
6307 2124104 MPOM NFLAG
6310 5150 EXCT AC0 /EXECUTE IF 1ST OR 3RD QUADRANT

```

```

6311 2055576 MCPAM FACML
6312 2111576 EVEN, MEMA FACML
6313      5021 RASH 1
6314      5212 CLL TLAC
6315 2001303 JMS FIXFLO /FLOAT FRACTION (N) - PLACE IN FAC
6316 110105 MEMA (N
6317 2404511 ACCM PX
6320 2001264 JMS FPUTAC /C(AC) IN N
6321 2000567 JMS FSQAR
6322 110110 MEMA (Q
6323 2404512 ACCM PZ
6324 2001264 JMS FPUTAC /N SQUARED IN Q
6325 2000467 JMS POX
6326 2110104 MEMA NFLAG
6327      5150 EXCT AC0
6330 2001103 JMS FNEG
6331 1000271 JMP @ FSIN

```

```

6332 3777764 K9, 3777764
6333 1173275 1173275
6334      0 0
6335 3777771 K7, 3777771
6336 2633314 2633314
6337      0 0
6340 3777775 K5, 3777775
6341 1214642 1214642
6342      0 0
6343      0 K3, 0
6344 2552421 2552421
6345      0 0
6346      1 K1, 1
6347 1444176 1444176
6350      0 0

```

/FLOATING POINT COSINE
/ARGUMENT IN RADIANS/(PI/2)

```

6351      0 FCOS, 0
6352 110132 MEMA (KK1
6353 2001246 JMS FGETAR
6354 2000610 JMS FADD /ADD PI/2
6355 2000271 JMS FSIN
6356 1000351 JMP @ FCOS

```

/LOG SUBROUTINE

```

6357      0 FLOG, 0
6360 110453 MEMA (A7
6361 2404507 ACCM TAD
6362 110003 MEMA (3
6363 2404510 ACCM NOM
6364 2111575 MEMA FACM
6365      5144 EXCT AC19
6366 2001103 JMS FNEG /ABSOLUTE VALUE OF FAC IN FAC
6367 2111575 MEMA FACM
6370 2404111 ACCM XM/GET MANTISSA OF X
6371 2111574 MEMA FACE
6372 2405575 ACCM FACM
6373 2164110 ZERM X

```

6374	2164112	ZERM XML
6375	2165576	ZERM FACML
6376	2001323	JMS FLOAT/ N IN FAC
6377	110442	MEMA (C1
6400	2001246	JMS FGETAR
6401	2001057	JMS FMULT / N*C1
6402	110445	MEMA (C2
6403	2001246	JMS FGETAR
6404	2000671	JMS FSUB /N*C1 - C2
6405	110241	MEMA (M
6406	2001264	JMS FPUTAC /M IN M
6407	110110	MEMA (X / GET X
6410	2404512	ACCM PZ
6411	2001230	JMS FGETAC
6412	110450	MEMA (C3
6413	2001246	JMS FGETAR
6414	2000610	JMS FADD /X+C3
6415	110104	MEMA (U
6416	2404511	ACCM PX
6417	2001264	JMS FPUTAC
6420	110110	MEMA (X
6421	2001230	JMS FGETAC
6422	110450	MEMA (C3
6423	2001246	JMS FGETAR
6424	2000671	JMS FSUB /X-C3
6425	110104	MEMA (U
6426	2001246	JMS FGETAR
6427	2001020	JMS FDIV /X+C3:X-C3 = U
6430	110104	MEMA (U
6431	2001264	JMS FPUTAC
6432	2000567	JMS FSQAR
6433	110110	MEMA (X
6434	2001264	JMS FPUTAC / Z = U SQAR
6435	2000467	JMS POX
6436	110241	MEMA (M
6437	2001246	JMS FGETAR
6440	2000610	JMS FADD /N*C1 - C2 + P(U)
6441	1000357	JMP @FLOG
6442	0	C1, 0
6443	1305617	1305617
6444	0	0
6445	3777777	C2, 3777777
6446	1305617	1305617
6447	0	0
6450	0	C3, 0
6451	1324046	1324046
6452	0	0
6453	3777777	A7, 3777777
6454	1150313	1150313
6455	0	0
6456	3777777	A5, 3777777
6457	1462400	1462400
6460	0	0
6461	0	A3, 0
6462	1252526	1252526

6463	0	0
6464	1	A1, 1
6465	1777777	1777777
6466	0	0
6467	0	POX, 0 /SUMMATION ROUTINE
6470	2110507	MEMA TAD /ENTER WITH Z IN FAC
6471	2001246	POLY, JMS FGETAR
6472	2001057	JMS FMULT
6473	1100003	MEMA (3
6474	2514507	A+MAM TAD /GET NEXT CONSTANT
6475	2001246	JMS FGETAR
6476	20000610	JMS FADD
6477	2706510	MMOMZ NOM
6500	505	JMP IAL /SET UP FOR NEXT LOOP
6501	2110511	MEMA PX /FINISHED LOOPING?
6502	2001246	JMS FGETAR
6503	2001057	JMS FMULT /P(X) IN FAC
6504	1000467	JMP @ POX
6505	2110512	IAL, MEMA PZ
6506	471	JMP POLY
6507	0	TAD, 0 /POINTER TO "A" TABLE
6510	0	NOM, 0 /COUNTER FOR POX LOOPS
6511	0	PX, 0 /POINTER TO X
6512	0	PZ, 0 / " " Z
*6567		
6567	0	FSQAR, 0
*6104		
6104	0	U, 0
6105	0	0
6106	0	0
6107	0	0
6110	0	X, 0
6111	0	XM, 0
6112	0	XML, 0
*7323		
7323	0	FLOAT, 0
*7574		
7574	0	FACE, 0
*7210		
7210	0	FACTEM, 0
*7220		
7220	0	TEMFAC, 0
*7020		
7020	0	FDIV, 0
*7230		
7230	0	FGETAC, 0
*7200		
7200	0	FACFAR, 0
*6103		
6103	0	OFLAG, 0
6104	0	TFLAG, 0
6105	0	Z, 0
6106	0	0

6107	0 0
*6241	
6241	0 M, 0
*6104	
6104	0 NFLAG, 0
6105	0 N, 0
6106	0 0
6107	0 0
6110	0 Q, 0
*7330	
7330	0 FIX, 0
*7576	
7576	0 FACML, 0
*7303	
7303	0 FIXFLO, 0
*6132	
6132	1 KK1, 1
*7103	
7103	0 FNEG, 0
*7575	
7575	0 FACM, 0
*7057	
7057	0 FMULT, 0
*7264	
7264	0 FPUTAC, 0
*7246	
7246	0 FGETAR, 0
*6610	
6610	0 FADD, 0
*6671	
6671	0 FSUB, 0

*6520

/SINGLE PRECISION FLOATING POINT SQUARE ROOT
/DESTROYS FAR AND TEMP

6520	0	FSORT, 0
6521	2111574	MEMA FACE
6522	5110	SKIP AC0 /EXPONENT ODD?
6523	2166573	ZERMZ ODDEXF
6524	2024573	ONEM ODDEXF /NO
6525	5021	RASH 1
6526	2404574	ACCM SAVEX
6527	2111575	MEMA FACM
6530	405160	EXCT ZAC
6531	1000520	JMP @FSQRT
6532	2165574	ZERM FACE
6533	5104	SKIP AC19 /MANTISSA POSITIVE?
6534	537	JMP FSQRT1
6535	2001103	JMS FNEG /NO
6536	2025562	ONEM ERRSQT /NEGATE FAC AND SET ERROR FLAG
6537	2001210	FSQRT1, JMS FACTEM
6540	110576	MEMA (KFSQ1 /HERONS METHOD USED
6541	2001246	JMS FGETAR /MAKE A GUESS
6542	2001057	JMS FMULT
6543	110601	MEMA (KFSQ2
6544	2001246	JMS FGETAR
6545	2000610	JMS FADD
6546	110002	MEMA (2 /REFINE GUESS 3 TIMES
6547	2404575	ACCM CNTR
6550	2001200	FSQRT2, JMS FACFAR
6551	2001220	JMS TEMFAC
6552	2001020	JMS FDIV
6553	2000610	JMS FADD
6554	2705574	MMOM FACE
6555	2706575	MMOMZ CNTR
6556	550	JMP FSQRT2
6557	2702573	MMOZ ODDEXF
6560	564	JMP FSQRT3
6561	110604	MEMA (KFSQ3
6562	2001246	JMS FGETAR
6563	2001057	JMS FMULT
6564	2110574	FSQRT3, MEMA SAVEX
6565	2505574	A+MM FACE
6566	1000520	JMP @FSORT

/SINGLE PRECISION FLOATING SQUARE, DESTROYS FAR

6567	0	FSQAR, 0
6570	2001200	JMS FACFAR
6571	2001057	JMS FMULT
6572	1000567	JMP @FSQAR
6573	0	ODDEXF, 0
6574	0	SAVEX, 0
6575	0	CNTR, 0

6576	0	KFSQ1, 0	/ .590162
6577	1134250	1134250	
6600	0	0	
6601	3777777	KFSQ2, 3777777	/ .417307
6602	1524450	1524450	
6603	0	0	
6604	1	KFSQ3, 1	/ 2** .5
6605	1324050	1324050	
6606	0	0	
*7562			
7562	0	ERRSQT, 0	
*7020			
7020	0	FDIV, 0	
*6610			
6610	0	FADD, 0	
*7570			
7570	0	FARE, 0	/ FLOATING ARGUMENT
7571	0	FARM, 0	
7572	0	FARML, 0	
7573	0	0	
7574	0	FACE, 0	/ FLOATING ACCUMULATOR
7575	0	FACM, 0	
7576	0	FACML, 0	
7577	0	0	
*7057			
7057	0	FMULT, 0	
*7103			
7103	0	FNEG, 0	
*7200			
7200	0	FACFAR, 0	
*7246			
7246	0	FGETAR, 0	
*7210			
7210	0	FACTEM, 0	
*7220			
7220	0	TEMFAC, 0	

/BASIC ARITHMETIC MODULE
 /INDEPENDENT OF ALL OTHER PROGRAMS

*6610

```

6610      0    FADD, 0 /DOUBLE PRECISION FLOATING ADD, DESTROYS FAR
6611 2103572 MEMZ FARML
6612      616  JMP FADD8
6613 2103571 MEMZ FARM
6614      616  JMP FADD8
6615 1000610 JMP @FADD /EXIT IF FAR IS ZERO
6616 2111574 FADD8, MEMA FACE
6617 2471570 A-MA FARE
6620      5104 SKIP AC19 /WHICH IS BIGGER?
6621      627  JMP FADD2 /FACE IS
6622 2404670 ACCM FADCNT /FARE IS
6623 2000701 FADD1, JMS DBRSHF /RIGHT SHIFT FAC
6624 2126670 MPOMZ FADCNT
6625      623  JMP FADD1
6626      646  JMP FADD5

6627 405160 FADD2, EXCT ZAC
6630      646  JMP FADD5 /OPERANDS ALREADY ALIGNED
6631 2224670 ANGM FADCNT
6632      5210 FADD3, CLL /RIGHT SHIFT FAR
6633 2111571 MEMA FARM
6634      5150 EXCT AC0
6635      5204 STL
6636      5021 RASH 1
6637 2405571 ACCM FARM
6640 2111572 MEMA FARML
6641      5021 RASH 1
6642      5202 TLAC
6643 2405572 ACCM FARML
6644 2126670 MPOMZ FADCNT
6645      632  JMP FADD3
6646 2000716 FADD5, JMS DIFFER
6647      5210 CLL /ADDITION DONE HERE
6650 2111572 MEMA FARML
6651 2515576 A+MAM FACML
6652 2111571 MEMA FARM
6653 2505575 A+MM FACM
6654      5141 EXCT L
6655 2125575 MPOM FACM
6656 2102726 MEMZ SIGNF /RIGHT SHIFT FAC IF NEEDED
6657      666  JMP FADD7
6660 2000701 JMS DBRSHF
6661 2111571 MEMA FARM
6662      5201 TACL
6663 2111575 MEMA FACM
6664      5202 TLAC /RESTORE SIGN
6665 2405575 ACCM FACM
6666 2001114 FADD7, JMS FNOR
6667 1000610 JMP @FADD

```

6670	0	FADCNT, 0 /SHIFT COUNTER
6671	0	FSUB, 0 /DOUBLE PRECISION FLOATING SUBTRACT
6672	5210	CLL /DESTROYS FAR
6673	2075572	MNGAM FARML /NEGATE ARGUMENT
6674	2045571	MCPM FARM
6675	5141	EXCT L
6676	2125571	MPOM FARM
6677	2000610	JMS FADD
6700	1000671	JMP @FSUB
6701	0	DBRSHF, 0 /DOUBLE RIGHT SHIFT OF FAC
6702	5210	CLL
6703	2111575	MEMA FACM
6704	5150	EXCT AC0
6705	5204	STL
6706	5021	RASH 1
6707	2405575	ACCM FACM
6710	2111576	MEMA FACML
6711	5021	RASH 1
6712	5202	TLAC
6713	2405576	ACCM FACML
6714	2125574	MPOM FACE
6715	1000701	JMP @DBRSHF
6716	0	DIFFER, 0 /REMEMBER SIGN OF ANSWERS
6717	2050766	MCPA K1777777
6720	2011575	ANDA FACM
6721	2511571	A+MA FARM
6722	5104	SKIP AC19 /SIGNS OF FAC AND FAR DIFFERENT?
6723	2166726	ZERMZ SIGNF /NO
6724	2024726	ONEM SIGNF /YES
6725	1000716	JMP @DIFFER
6726	0	SIGNF, 0
6727	0	SPMPY, 0 /SINGLE PRECISION UN-SIGNED MULTIPLY SUBROUTINE
6730	110024	MEMA (24 /EXIT WITH LEAST SIGNIFICANT PART IN AC AND MP
6731	2404670	ACCM FADCNT /MOST SIGNIFICANT PART IN MP5
6732	2164765	ZERM MP5
6733	5210	CLL
6734	2110763	LOOP, MEMA MP1
6735	5061	RLSH 1
6736	5104	SKIP AC19
6737	757	JMP OUT
6740	5202	TLAC
6741	2404763	ACCM MP1
6742	2110765	MEMA MP5
6743	2510764	A+MA MP2
6744	5061	BACK, RLSH 1
6745	5201	TACL
6746	2010766	ANDA K1777777
6747	2404765	ACCM MP5
6750	2706670	MMOMZ FADCNT /DONE?
6751	734	JMP LOOP
6752	2110763	MEMA MP1 /YES
6753	5061	RLSH 1
6754	5202	TLAC
6755	2404763	ACCM MP1

```

6756 1000727 JMP @SPMPY
6757      5202 OUT, TLAC
6760 2404763 ACCM MP1
6761 2110765 MEMA MP5
6762      744 JMP BACK

6763      0 MP1, 0 /NUMERATOR IN PDIV
6764      0 MP2, 0 /DNOMINATOR IN PDIV
6765      0 MP5, 0 /QUOTIENT IN PDIV
6766 1777777 K1777777, 1777777

6767      0 PDIV, 0 /PECULIAR DIVIDE
6770 110024 MEMA C24 /REQUIRES NORMALIZED OPERANDS
6771 2404670 ACCM FADCNT
6772 2110763 PDIV1, MEMA MP1
6773 2470764 PDIV2, A-MA MP2
6774      5104 SKIP AC19
6775      1007 JMP PDIV3
6776 2110765 MEMA MP5
6777      5001 LASH 1
7000 2404765 ACCM MP5
7001 2110763 MEMA MP1
7002      5001 LASH 1
7003 2404763 ACCM MP1
7004 2706670 MMOMZ FADCNT
7005      773 JMP PDIV2
7006 1000767 JMP @PDIV
7007      5001 PDIV3, LASH 1
7010 2404763 ACCM MP1
7011 2110765 MEMA MP5
7012      5206 STL TLAC
7013      5041 LLSH 1
7014 2404765 ACCM MP5
7015 2706670 MMOMZ FADCNT
7016      772 JMP PDIV1
7017 1000767 JMP @PDIV

7020      0 FDIV, 0 /SINGLE PRECISION FLOATING POINT DIVIDE
7021 2001044 JMS PREPAR /ASSUMES NORMALIZED INPUT, PRESERVES FAR
7022 2103571 MEMZ FARM /TEST FOR DIVISION BY ZERO
7023      1026 JMP FDIV1
7024 2025563 ONEM ERRDIV
7025 1001020 JMP @FDIV
7026 2111570 FDIV1, MEMA FARE
7027 2325574 M-AM FACE
7030 2000767 JMS PDIV
7031 2110765 MEMA MP5
7032      5104 SKIP AC19
7033      1037 JMP FDIV2
7034      5061 RLSH 1
7035      5212 CLL TLAC
7036 2125574 MPOM FACE
7037 2405575 FDIV2, ACCM FACM
7040 2102726 MEMZ SIGNF
7041 2065575 MNGM FACM
7042 2165576 ZERM FACML
7043 1001020 JMP @FDIV

```

7044	0	PREPAR, 0 /PREPARES FOR MULTIPLICATION OR DIVISION
7045	2000716	JMS DIFFER /DETERMINE SIGN OF ANSWER
7046	2111575	MEMA FACM
7047	5144	EXCT AC19
7050	230000	ANGA
7051	2404763	ACCM MP1
7052	2111571	MEMA FARM
7053	5144	EXCT AC19
7054	230000	ANGA
7055	2404764	ACCM MP2
7056	1001044	JMP @PREPAR
7057	0	FMULT, 0 /SINGLE PRECISION FLOATING POINT MULTIPLY
7060	2111570	MEMA FARE /ASSUMES NORMALIZED INPUT
7061	2505574	A+MM FACE /PRESERVES FAR
7062	2001044	JMS PREPAR
7063	2000727	JMS SPMPY
7064	2011102	ANDA K3000000
7065	2510765	A+MA MP5
7066	5042	LLSH 2
7067	5104	SKIP AC19 /IS PRODUCT SMALLER THAN .5?
7070	1100	JMP OUTF
7071	5021	RASH 1 /NO
7072	5212	CLL TLAC
7073	2102726	BACKF, MEMZ SIGNF
7074	230000	ANGA
7075	2405575	ACCM FACM
7076	2165576	ZERM FACML
7077	1001057	JMP @FMULT
7100	2705574	OUTF, MMOM FACE
7101	1073	JMP BACKF
7102	3000000	K3000000, 3000000
7103	0	FNEG, 0 /NEGATE FAC
7104	2111576	MEMA FACML
7105	5210	CLL
7106	2235576	ANGAM FACML
7107	2051575	MCPA FACM
7110	5141	EXCT L
7111	430000	APOA
7112	2405575	ACCM FACM
7113	1001103	JMP @FNEG
7114	0	FNOR, 0 /NORMALIZE FAC
7115	2103576	MEMZ FACML
7116	1122	JMP FNOR1
7117	2103575	MEMZ FACM
7120	1122	JMP FNOR1
7121	1157	JMP FNOR5 /FAC CONTAINS 0
7122	2111575	FNOR1, MEMA FACM
7123	2165162	ZERM FNORF
7124	5104	SKIP AC19
7125	1137	JMP FNOR2
7126	2025162	ONEM FNORF
7127	2001103	JMS FNEG

7130	5104	SKIP AC19
7131	1137	JMP FNOR2
7132	5061	RLSH 1 /MANTISSA = 2000000, 0000000
7133	2405575	ACCM FACM /CHANGE TO LEGAL NUMBER
7134	2001103	JMS FNEG
7135	2125574	MPOM FACE
7136	1001114	JMP @FNOR
7137	5041	FNOR2, LLSH 1
7140	5144	EXCT AC19
7141	1154	JMP NEXIT
7142	2111576	MEMA FACML /DOUBLE LEFT SHIFT OF FAC
7143	5201	TACL
7144	5001	LASH 1
7145	2405576	ACCM FACML
7146	2111575	MEMA FACM
7147	5202	TLAC
7150	5041	LLSH 1
7151	2405575	ACCM FACM
7152	2705574	MMOM FACE
7153	1137	JMP FNOR2
7154	2103162	NEXIT, MEMZ FNORF
7155	2001103	JMS FNEG
7156	1001114	JMP @FNOR
7157	70047	FNOR5, MNCA (47
7160	2405574	ACCM FACE
7161	1001114	JMP @FNOR
7162	0	FNORF, 0

*7563

7563	0	ERRDIV, 0 /SET TO ONE IF DIVIDE BY 0 ATTEMPTED
7564	0	TEMPEX, 0 /TEMPORARY STORAGE
7565	0	TEMPM, 0
7566	0	TEMPML, 0
7567	0	0
7570	0	FARE, 0 /FLOATING ARGUMENT
7571	0	FARM, 0
7572	0	FARML, 0
7573	0	0
7574	0	FACE, 0 /FLOATING ACCUMULATOR
7575	0	FACM, 0
7576	0	FACML, 0
7577	0	0

/MOVE MODULE
/INDEPENDENT OF ALL OTHER PROGRAMS

*7200

/MOVE FAC TO FAR

7200	0	FACFAR, 0
7201	2111574	MEMA FACE
7202	2405570	ACCM FARE
7203	2111575	MEMA FACM
7204	2405571	ACCM FARM
7205	2111576	MEMA FACML
7206	2405572	ACCM FARML
7207	1001200	JMP @FACFAR

/MOVE FAC TO TEM

7210	0	FACTEM, 0
7211	2111574	MEMA FACE
7212	2405564	ACCM TEMPEX
7213	2111575	MEMA FACM
7214	2405565	ACCM TEMPM
7215	2111576	MEMA FACML
7216	2405566	ACCM TEMPML
7217	1001210	JMP @FACTEM

/MOVE TEM TO FAC

7220	0	TEMFAC, 0
7221	2111564	MEMA TEMPEX
7222	2405574	ACCM FACE
7223	2111565	MEMA TEMPM
7224	2405575	ACCM FACM
7225	2111566	MEMA TEMPML
7226	2405576	ACCM FACML
7227	1001220	JMP @TEMFAC

/MOVE 3 WORDS TO FAC

7230	0	FGETAC, 0
7231	2405200	ACCM POINT
7232	2111230	MEMA FGETAC
7233	2011302	ANDA MASK
7234	2505200	A+MM POINT
7235	3111200	MEMA @POINT
7236	2405574	ACCM FACE
7237	2125200	MPOM POINT
7240	3111200	MEMA @POINT
7241	2405575	ACCM FACM
7242	2125200	MPOM POINT
7243	3111200	MEMA @POINT
7244	2405576	ACCM FACML
7245	1001230	JMP @FGETAC

/MOVE 3 WORDS TO FAR

7246	0	FGETAR, 0
7247	2405200	ACCM POINT
7250	2111246	MEMA FGETAR
7251	2011302	ANDA MASK


```

7252 2505200 A+MM POINT
7253 3111200 MEMA @POINT
7254 2405570 ACCM FARE
7255 2125200 MPOM POINT
7256 3111200 MEMA @POINT
7257 2405571 ACCM FARM
7260 2125200 MPOM POINT
7261 3111200 MEMA @POINT
7262 2405572 ACCM FARML
7263 1001246 JMP @FGETAR

```

/MOVE 3 WORDS FROM FAC

```

7264      0 FPUTAC, 0
7265 2405200 ACCM POINT
7266 2111264 MEMA FPUTAC
7267 2011302 ANDA MASK
7270 2505200 A+MM POINT
7271 2111574 MEMA FACE
7272 3405200 ACCM @POINT
7273 2125200 MPOM POINT
7274 2111575 MEMA FACM
7275 3405200 ACCM @POINT
7276 2125200 MPOM POINT
7277 2111576 MEMA FACML
7300 3405200 ACCM @POINT
7301 1001264 JMP @FPUTAC

7302 3776000 MASK, 3776000

```

*7200

```

7200      0 POINT, 0

```

*7564

```

7564      0 TEMPEX, 0 /TEMPORARY STORAGE
7565      0 TEMPM, 0
7566      0 TEMPML, 0
7567      0 0

7570      0 FARE, 0 /FLOATING ARGUMENT
7571      0 FARM, 0
7572      0 FARML, 0
7573      0 0

7574      0 FACE, 0 /FLOATING ACCUMULATOR
7575      0 FACM, 0
7576      0 FACML, 0
7577      0 0

```

/FIXED POINT TO FLOATING POINT CONVERSION.
 /AC CONTAINS NUMBER TO BE CONVERTED, RESULT IN FAC.
 /BINARY POINT BETWEEN BIT 18 AND 19

*7303

7303	0	FIXFLO, 0
7304	2165574	ZERM FACE
7305	2165576	ZERM FACML
7306	2405575	ACCM FACM
7307	2001114	JMS FNOR
7310	1001303	JMP @FIXFLO

/FLOATING POINT TO FIXED POINT CONVERSION. REVERSES EFFECT OF FIXFLO.

7311	0	FLOFIX, 0
7312	2715574	MMOAM FACE
7313	5104	SKIP AC19 /NORMALIZABLE?
7314	2145574	MONM FACE /NO
7315	2111575	MEMA FACM
7316	2127574	FLO1, MPOMZ FACE /DONE?
7317	162000	ZERZ
7320	1001311	JMP @FLOFIX /YES
7321	5021	RASH 1
7322	1316	JMP FLO1

7323	0	FLOAT, 0
7324	110023	MEMA (23
7325	2405574	ACCM FACE
7326	2001114	JMS FNOR
7327	1001323	JMP @FLOAT
7330	0	FIX, 0
7331	2111574	MEMA FACE
7332	470023	A-MA (23
7333	2415574	ACCMA FACE
7334	5104	SKIP AC19
7335	2165574	ZERM FACE
7336	2103574	FLO2, MEMZ FACE
7337	162000	ZERZ
7340	1001330	JMP @FIX
7341	2000701	JMS DBRSHF
7342	1336	JMP FLO2

*7114

7114	0	FNOR, 0
------	---	---------

*6701

6701	0	DBRSHF, 0
------	---	-----------

*7574

7574	0	FACE, 0
7575	0	FACM, 0
7576	0	FACML, 0