```
                                ,TITLE 'FP15T2' TEST MODULE FOR SYSTEM EXERCISER
                                ,EBREL
                            /VERSION# A
                            /COPYRIGHT SEPT, 22, 1971
                            /DIGITAL EQUIPMENT CORPORATION, MAYNARD MASS, 01754
                            /PROGRAMMER: EARL L. BOUSE
                                    /
                                    /
                            /THIS PROGRAM IS DESIGNED AS A RANDOM FP15 INSTRUCTION EXERCISER
                            /FOR THE FP15 FLOATING POINT PROCESSOR,
                                    /
                                    /
00000 R 000000 A    USERSW  0               /LOAD AS A 'CP' MODULE,
00001 R 000000 A            0
00002 R 000000 A            0
00003 R 000000 A            0
00004 R 000062 R            ,DSA    SERVICE
00005 R 000032 R            ,DSA    INIT
00006 R 062061 A            ,SIXBT  'FP15T2'
00007 R 652462 A
00010 R 000200 A            200                     /DATA SWITCH 10 INHIBITS TEST,
00011 R        A            ,BLOCK 7
00020 R 000000 A    SYSERR  0                       /ERROR INDICATOR FOR MONITOR,
00021 R 000000 A            0
00022 R 000000 A    ERCODE  0
00023 R 000000 A            0
00024 R 000000 A            0
00025 R 000000 A            0
00026 R 000000 A            0
00027 R 000000 A            0
00030 R 000000 A            0
00031 R 000000 A            0
                            ,EJECT
```

```
                    /TO STORE ADDRESSES INTO EXECUTIVE AT LOAD TIME
                    /DEFINITIONS FOR EXERCISER
000010 A    UR=10                   /UNROUNDED
000020 A    UN=20                   /UNNORMALIZED
000030 A    UU=30                   /UNNORMALIZED/UNROUNDED
000040 A    FL=40                   /FLOATING
000100 A    DP=100                  /DOUBLE PRECISION
000200 A    IM=200                  /IMMEDIATE
000400 A    SUB=400                 /SUBTRACT
001000 A    RSUB=1000               /REVERSE SUBTRACT
003000 A    LD=3000                 /BASIC FP LOAD
003600 A    ST=IM!3400              /STORE IMMEDIATE
712000 A    DV=712000               /BASIC DIVIDE
715400 A    LJE=715400              /LOAD JUMP EXIT REGISTER
715600 A    SJE=715600              /STORE JUMP EXIT REGISTER
001400 A    MUL=1400                /BASIC MULTIPLY
002000 A    DIV=2000                /BASIC DIVIDE
002400 A    RDIV=2400               /BASIC REVERSE DIVIDE
006000 A    ADD=6000                /BASIC ADD
007400 A    OP6TO9=7400             /OPCODE 6 TO 9 MASK
710000 A    DRR=710000              /DIAGNOSTIC READ REGISTERS
710100 A    DSR=710100              /DIAGNOSTIC STEP AND READ REGISTERS
713250 A    FNM=713250              /NORMALIZE FMA (ASIGN) EPA
716601 A    BZA=716601              /BRANCH IF FMA ZERO
716602 A    BMA=716602              /BRANCH IF FMA NEGATIVE
716603 A    BLE=716603              /BRANCH IF FMA < OR = 0
716604 A    BPA=716604              /BRANCH IF FMA POSITIVE
716606 A    BRU=716606              /BRANCH UNCONDITIONAL
716610 A    BNA=716610              /BRANCH IF FMA NON-ZERO
716620 A    BAC=716620              /BRANCH IF GUARD BIT IS SET
006400 A    BR=6400                 /BASIC BRANCH
717300 A    DMN=717300              /DIAGNOSTIC MAINTENANCE MODE ON
717200 A    DMF=717200              /DIAGNOSTIC MAINTENANCE MODE OFF
710314 A    FPT=710314              /FLOATING POINT TEST
                    .EJECT
```

```
                        /INITIALIZE THE 'FP15T2' TEST,
                        /
00032 R 000000 A    INIT    0
00033 R 707764 A            EBA
00034 R 200032 R            LAC     INIT
00035 R 040062 R            DAC     SERVICE
00036 R 140020 R            DZM     SYSERR          /CLEAR THE ERROR LOCATIONS,
00037 R 140021 R            DZM     SYSERR+1
00040 R 140022 R            DZM     ERCODE
00041 R 777000 A            LAW     -1000
00042 R 040545 R            DAC     K1000
00043 R 776000 A            LAW     -2000
00044 R 040546 R            DAC     K12000
00045 R 777400 A            LAW     -400
00046 R 044561 R            DAC     RATIO
00047 R 100656 R            JMS     FPRAN           /INITIALIZE OP-CODE BUFFER,
00050 R 104235 R            JMS     HOLDSW          /CHECK FOR THE HOLD SWITCH,
00051 R 710314 A            FPT                     /TEST FOR THE FP15
00052 R 604266 R            JMP     ERROR1          /FP15 IS NOT OUT THERE,
00053 R 204572 R            LAC     (FPEXER
00054 R 040073 R            DAC     DSTSW0          /SET UP DISTRIBUTION SWITCH,
00055 R 777730 A            LAW     -50
00056 R 044545 R            DAC     PASCNT
00057 R 204573 R            LAC     (720000
00060 R 044543 R            DAC     PGMLOP
00061 R 604256 R            JMP     EXITM5          /START THE TEST,
                        /
                        /
                        /
                        /
                        /
                        /
                        /SERVICE ENTRANCE FOR THE FP15T2,
                        /
00062 R 000000 A    SERVICE 0
00063 R 707764 A            EBA
00064 R 750004 A            LAS
00065 R 500010 R            AND     USERSW+10
00066 R 740200 A            SZA                     /IS THE HOLD SWITCH SET?
00067 R 604256 R            JMP     EXITM5          /YES, EXIT
00070 R 715400 A            LJE                     /LOAD THE 'JEA' REGISTER,
00071 R 004574 R            (XJEA)
00072 R 620073 R            JMP*    DSTSW0
00073 R 000000 A    DSTSW0  0                       /PROGRAM DISTRIBUTION LOCATION,
                        /
                        ,EJECT
```

```
                        /FPU EXERCISER RANDOM HARDWARE OP-CODE EXECUTIONER
00074 R 140540 R    FPEXER  DZM OVRFLR
00075 R 140536 R            DZM OVRFL1
00076 R 140541 R            DZM UNDFLN
00077 R 140537 R            DZM UNDFL1
00100 R 140542 R            DZM DVBY00
00101 R 140543 R            DZM OVRFLX
00102 R 140544 R            DZM IDVBY0
00103 R 224547 R            LAC* AUTORG     /RANDOM OPERATION
00104 R 044552 R            DAC ADRESS#
00105 R 444547 R            ISZ AUTORG
00106 R 224552 R            LAC* ADRESS
00107 R 040527 R            DAC OPCODE
00110 R 100567 R            JMS RANDAT      /GENERATE RANDOM NUMBERS,
00111 R 101556 R            JMS IFREST      /INTEGER FLOATING RESTRICTIONS FOR RANDEX
00112 R 200527 R            LAC OPCODE
00113 R 504575 R            AND (OP6TO9)
00114 R 544576 R            SAD (ADD)
00115 R 601033 R            JMP FPPADD      /SOFTWARE ,,,,ADDITION
00116 R 544577 R            SAD (SUB)
00117 R 601040 R            JMP FPPSUB      /,,,,SUBTRACTION
00120 R 544600 R            SAD (MUL)
00121 R 601045 R            JMP FPPMUL      /,,,,MULTIPLICATION
00122 R 544601 R            SAD (DIV)
00123 R 601052 R            JMP FPPDIV      /,,,,DIVISION
00124 R 544602 R            SAD (RSUB)
00125 R 601057 R            JMP FPPRSB      /,,,,REVERSE SUBTRACTION
00126 R 601064 R            JMP FPPRDV      /,,,,REVERSE DIVISION
                        /SOFTWARE COMPULATIONS COMPLETE***************
                        /*********************************************
                        /INITIALIZE FPU HARDWARE OPERATIONAL BOOKEEPING
00127 R 200527 R    FPUXA1  LAC OPCODE
00130 R 504603 R            AND (FL)
00131 R 544604 R            SAD (CAL)
00132 R 600142 R            JMP FPUXA2      /INTEGER OPERATINOS
00133 R 204605 R            LAC (%ANSR1)
00134 R 040352 R            DAC FPANSR
00135 R 204606 R            LAC (%XFAC1)
00136 R 040225 R            DAC FPFMA       /3 OR 2 WORD XFER FLOATING FMA
00137 R 204607 R            LAC (%XHAC1)
00140 R 040350 R            DAC FPFMB       /AND FLOATING FMB
00141 R 600163 R            JMP FPUXA3
00142 R 200527 R    FPUXA2  LAC OPCODE
00143 R 504610 R            AND (DP)
00144 R 544604 R            SAD (CAL)
00145 R 600155 R            JMP FPUX2A
00146 R 204611 R            LAC (%ANSR2)
00147 R 040352 R            DAC FPANSR
00150 R 204612 R            LAC (%XFAC2)
00151 R 040225 R            DAC FPFMA       /2 OR 1 WORD XFER INTEGER FMA
00152 R 204613 R            LAC (%XHAC2)
00153 R 040350 R            DAC FPFMB       /AND INTEGER FMB
00154 R 600163 R            JMP FPUXA3
```

```
00155 R 204614 R    FPUX2A  LAC (%ANSR3)
00156 R 040352 R            DAC FPANSR
00157 R 204615 R            LAC (%XFAC3)
00160 R 040225 R            DAC FPFMA
00161 R 204616 R            LAC (%XHAC3)
00162 R 040350 R            DAC FPFMB
00163 R 200527 R    FPUXA3  LAC OPCODE
00164 R 504617 R            AND (770367)
00165 R 344620 R            TAD (UR)
00166 R 344621 R            TAD (LD)        /3000
00167 R 040224 R            DAC FPULD
00170 R 200527 R            LAC OPCODE
00171 R 504622 R            AND (770177)
00172 R 344623 R            TAD (ST)        /3600
00173 R 040351 R            DAC FPUST       /ALWAYS IMMEDIATE
00174 R 504624 R            AND (-UU-1)
00175 R 344625 R            TAD (UR!UN)
00176 R 040270 R            DAC CFPUST      /ALWAYS UR!UN
00177 R 101556 R            JMS IFREST
00200 R 101573 R            JMS DPFRES
                    /SOFTWARE OPERATIONS COMPLETED AND ADDRESSES RESERVED
                    /
                    /VIA HARDWARE COMPUTE THE FPU FUNCTION IN "OPCODE"
                    /
                    /INITIALIZE HARDWARE TO SOFTWARE EXECUTION RATIO 400 TO 1
00201 R 204561 R            LAC RATIO#
00202 R 740100 A            SMA
00203 R 777400 A            LAW -400
00204 R 044561 R            DAC RATIO
00205 R 104235 R    FPLOOP  JMS HOLDSW              /CHECK FOR THE HOLD SWITCH,
00206 R 444545 R            ISZ PASCNT
00207 R 600216 R            JMP .+7
00210 R 104252 R            JMS PCNEXT              /EXIT '-5',
00211 R 777730 A            LAW -50
00212 R 044545 R            DAC PASCNT
00213 R 444543 R            ISZ PGMLOP
00214 R 741000 A            SKP
00215 R 604261 R            JMP EXITM4              /TYPE 'DONE'
00216 R 141432 R            DZM XJEA
00217 R 141434 R            DZM XJEA+2
00220 R 141436 R            DZM XJEA+4
00221 R 141440 R            DZM XJEA+6
00222 R 200224 R            LAC FPULD
00223 R 044555 R            DAC FFAIL#
00224 R 000000 A    FPULD   CAL             /MODIFIED TO CONTAIN HARDWARE LOAD UR+OPCODE,,,,ETC.
00225 R 000000 A    FPFMA   CAL             /MODIFIED TO CONTAIN ADDRESS %XFAC1 OR %XFAC2
                            .EJECT
```

```
                        /IF DPI OR DPF AND MANTISS = 400000 000000
                        /HARDWARE WILL STORE AS 000000 000000
00226 R 200527 R                LAC OPCODE
00227 R 504610 R                AND (DP)
00230 R 544610 R                SAD (DP)
00231 R 741000 A                SKP
00232 R 600247 R                JMP CFPU        /SINGLE PRECISION
00233 R 200515 R                LAC %XFAC3
00234 R 740200 A                SZA
00235 R 600247 R                JMP CFPU        /FMA LOW NOT = 000000
00236 R 200514 R                LAC %XFAC2
00237 R 544626 R                SAD (400000)
00240 R 140514 R                DZM %XFAC2      /FMA HIGH = 400000 AND FMA LOW = 000000
00241 R 200520 R                LAC %XHAC3
00242 R 740200 A                SZA
00243 R 600247 R                JMP CFPU
00244 R 200517 R                LAC %XHAC2
00245 R 544626 R                SAD (400000)
00246 R 140517 R                DZM %XHAC2
                        /IF FLOATING AND MANTISSAS = 0 CLEAR EXPONENT %XFAC1 FOR COMPARISON
                        /IN CTEST1 BUDDY BROTHER
00247 R 200527 R        CFPU    LAC OPCODE
00250 R 504627 R                AND (DP!FL)
00251 R 544603 R                SAD (FL)
00252 R 741000 A                SKP             /SPF
00253 R 600263 R                JMP .+10        /DPF
00254 R 200513 R                LAC %XFAC1
00255 R 504630 R                AND (777000)
00256 R 540514 R                SAD %XFAC2
00257 R 740200 A                SZA
00260 R 600262 R                JMP .+2
00261 R 140513 R                DZM %XFAC1      /MANTISSA = 0
00262 R 600270 R                JMP .+6
00263 R 200514 R                LAC %XFAC2
00264 R 540515 R                SAD %XFAC3
00265 R 740200 A                SZA
00266 R 741000 A                SKP
00267 R 140513 R                DZM %XFAC1
                                .EJECT
```

```
                        /CONFIRM LOAD/STORE,,,UR,,,FUNCTIONS PROPERLY
00270 R 000000 A    CFPUST  CAL             /STORE UR!UN
00271 R 400352 R            FPANSR+400000   /INDIRECT FPANSR
                        /NO JEA EXCEPTIONS SHOULD HAVE OCCURRED
                        /
                        /MIGHT LOOK LIKE REDUNDANT CHECKING BUT IT ISN'T
                        /SOFTWARE MIGHT EXPECT A JEA FUNCTION BUT THAT FUNCTION SHOULD NOT OCCUR HERE
                        /
                        /EXCEPT IF SINGLE PRECISION INTEGER AND FMA = 400000 THEN EXPECT OVERFLOW #30
00272 R 777777 A            LAW -1
00273 R 044554 R            DAC COMPLD#
00274 R 200527 R            LAC OPCODE
00275 R 504627 R            AND (DP!FL)
00276 R 544604 R            SAD (CAL)
00277 R 741000 A            SKP
00300 R 600304 R            JMP CTEST0
00301 R 200515 R            LAC %XFAC3
00302 R 544626 R            SAD (400000)
00303 R 600353 R            JMP TPRT        /EXPECT OVERFLOW #30
00304 R 201432 R    CTEST0  LAC XJEA
00305 R 740200 A            SZA
00306 R 601502 R            JMP XFLOW       /UNEXPECTED OVERFLOW
00307 R 201434 R            LAC XJEA+2
00310 R 740200 A            SZA
00311 R 601502 R            JMP XFLOW       /UNEXPECTED UNDERFLOW
00312 R 201436 R            LAC XJEA+4
00313 R 740200 A            SZA
00314 R 601502 R            JMP XFLOW       /UNEXPECTED ABNORMAL DIVISION
00315 R 201440 R            LAC XJEA+6
00316 R 740200 A            SZA
00317 R 601502 R            JMP XFLOW       /UNEXPECTED ILLEGAL MEM REFERENCE
00320 R 200534 R            LAC %NA1
00321 R 544631 R            SAD (-1)
00322 R 741000 A            SKP
00323 R 604370 R            JMP ERROR5
00324 R 200535 R            LAC %NA2
00325 R 544631 R            SAD (-1)
00326 R 741000 A            SKP
00327 R 604370 R            JMP ERROR5
00330 R 200513 R    CTEST1  LAC %XFAC1
00331 R 540531 R            SAD %ANSR1
00332 R 741000 A            SKP
00333 R 604370 R            JMP ERROR5      /EXPONENT FAILURE LOAD/STORE
00334 R 200515 R    CTEST3  LAC %XFAC3
00335 R 540533 R            SAD %ANSR3
00336 R 741000 A            SKP
00337 R 604370 R            JMP ERROR5      /FMA FAILURE LOAD/STORE
00340 R 200514 R    CTEST2  LAC %XFAC2
00341 R 540532 R            SAD %ANSR2
00342 R 741000 A            SKP
00343 R 604370 R            JMP ERROR5      /FMA FAILURE LOAD/STORE
00344 R 144554 R            DZM COMPLD
                            ,EJECT
```

```
00345 R 200527 R            LAC OPCODE
00346 R 040347 R            DAC .+1
00347 R 740040 A            XX
00350 R 000000 A    FPFMB   CAL             /MODIFIED TO CONTAIN ADDRESS %XHAC1 OR %XHAC2
                    /ALL OPERATIONS COMPLETED AND SAVED IN 2'S COMP COMPARE RESULTS
00351 R 000000 A    FPUST   CAL             /MODIFIED TO CONTAIN HARDWARE STORE+...ETC,
00352 R 000000 A    FPANSR  CAL             /MODIFIED TO CONTAIN ADDRESS %ANSR1 OR %ANSR2
                    /HARDWARE OPERATIONS COMPLETE-CHECK FOR UNOCCURED BUT EXPECTED JEA EXCEPTIONS
00353 R 201432 R    TPRT    LAC XJEA
00354 R 740200 A            SZA
00355 R 600367 R            JMP TPRT1
00356 R 200540 R            LAC OVRFLR
00357 R 740200 A            SZA
00360 R 601540 R            JMP EINT11      /OVERFLOW EXPECTED...
00361 R 200536 R            LAC OVRFL1
00362 R 740200 A            SZA
00363 R 601542 R            JMP EINT12      /...BUT NONE OCCURED
00364 R 200543 R            LAC OVRFLX
00365 R 740200 A            SZA
00366 R 601544 R            JMP EINT30      /...ETC
00367 R 201434 R    TPRT1   LAC XJEA+2
00370 R 740200 A            SZA
00371 R 600400 R            JMP TPRT2
00372 R 200541 R            LAC UNDFLN
00373 R 740200 A            SZA
00374 R 601546 R            JMP EINT13      /UNDERFLOW EXPECTED...
00375 R 200537 R            LAC UNDFL1
00376 R 740200 A            SZA
00377 R 601550 R            JMP EINT14      /...BUT NONE OCCURED
00400 R 201436 R    TPRT2   LAC XJEA+4
00401 R 740200 A            SZA
00402 R 600411 R            JMP TEST1
00403 R 200542 R            LAC DVBY00
00404 R 740200 A            SZA
00405 R 601552 R            JMP EINT15      /...ABNORMAL DIVISION EXPECTED BUT NONE OCCURED
00406 R 200544 R            LAC IDVBY0
00407 R 740200 A            SZA
00410 R 601554 R            JMP EINT35      /...INTEGER DIVISION BY ZERO EXPECTED BUT NONE OCCURED
                            .EJECT
```

```
00411 R 200566 R     TEST1   LAC OVRFL1
00412 R 340540 R             TAD OVRFLR
00413 R 340541 R             TAD UNDFLN
00414 R 340537 R             TAD UNDFL1
00415 R 340543 R             TAD OVRFLX
00416 R 340542 R             TAD DVBY00
00417 R 340544 R             TAD IDVBY0
00420 R 740200 A             SZA
00421 R 600443 R             JMP FPALLOK+2   /EXPECTED INTERRUPT EXCEPTION OCCURED - IGNORE RESULT
00422 R 203704 R             LAC %FAC1       /SOFTWARE RESULT
00423 R 540531 R             SAD %ANSR1      /HARDWARE RESULT
00424 R 741000 A             SKP
00425 R 100436 R             JMS FPERROR     /EPA RESULT WRONG
00426 R 203706 R     TEST3   LAC %FAC3       /SOFTWARE RESULT
00427 R 540533 R             SAD %ANSR3      /HARDWARE RESULT
00430 R 741000 A             SKP
00431 R 100436 R             JMS FPERROR     /FMA (LOW ORDER) RESULT WRONG
00432 R 203705 R     TEST2   LAC %FAC2       /SOFTWARE RESULT
00433 R 540532 R             SAD %ANSR2      /HARDWARE RESULT
00434 R 600441 R             JMP FPALLOK
00435 R 100436 R             JMS FPERROR     /FMA (HIGH ORDER) RESULT WRONG
00436 R 000000 A     FPERROR CAL             /JMS IS FOR DIAGNOSTIC AID, BOY,
00437 R 104332 R             JMS ERROR2
00440 R 600444 R             JMP FPABORT
00441 R 444561 R     FPALLOK ISZ RATIO       /NORMAL RETURN,,,UNTIL RATIO=0
00442 R 600205 R             JMP FPLOOP
00443 R 101635 R             JMS TFBRA
                     /RATIO OPERATIONS COMPLETE
00444 R 104235 R     FPABORT JMS HOLDSW              /CHECK FOR HOLD SWITCH,
00445 R 204547 R             LAC AUTORG
00446 R 544632 R             SAD (AEOPCDB    /END OF OP-CODE BUFFER?
00447 R 741000 A             SKP             /YES, RESET TABLE
00450 R 600074 R             JMP FPEXER
                             ,EJECT
```

```
00451 R 440546 R            ISZ KT2000
00452 R 600455 R            JMP .+3
00453 R 101776 R            JMS INITADR
00454 R 600505 R            JMP FEXIT
00455 R 440545 R            ISZ K1000
00456 R 600505 R            JMP FEXIT
00457 R 777000 A            LAW -1000
00460 R 040545 R            DAC K1000
00461 R 204633 R            LAC (AOPCODBF)
00462 R 044547 R            DAC AUTORG
00463 R 777716 A            LAW -62
00464 R 044551 R            DAC TEMP2
00465 R 104515 R    LOOP1   JMS RANGEN
00466 R 504634 R            AND (37777)     /32K
00467 R 040530 R            DAC TEMP
00470 R 344635 R            TAD (-EOPCODB-1)
00471 R 740100 A            SMA
00472 R 600465 R            JMP LOOP1
00473 R 204636 R            LAC (-OPCODBF-1)
00474 R 340530 R            TAD TEMP
00475 R 741100 A            SPA
00476 R 600465 R            JMP LOOP1
00477 R 200530 R            LAC TEMP
00500 R 064547 R            DAC* AUTORG
00501 R 444547 R            ISZ AUTORG
00502 R 740000 A            NOP
00503 R 444551 R            ISZ TEMP2
00504 R 600465 R            JMP LOOP1
00505 R 204633 R    FEXIT   LAC (AOPCODBF)
00506 R 044547 R            DAC AUTORG
00507 R 600074 R            JMP FPEXER
                            .EJECT
```

```
00510 R 000000 A    FACLOC  0
00511 R 000000 A    ESGLOC  0
00512 R 000000 A    EXPLOC  0
00513 R 000000 A    %XFAC1  0               /TEMPORARY
00514 R 000000 A    %XFAC2  0               /SOFTWARE
00515 R 000000 A    %XFAC3  0               /FMA
00516 R 000000 A    %XHAC1  0               /TEMPORARY
00517 R 000000 A    %XHAC2  0               /SOFTWARE
00520 R 000000 A    %XHAC3  0               /FMB
00521 R 000000 A    AC1LOC  0               /%XFAC1, %XHAC1
00522 R 000000 A    AC2LOC  0               /%XFAC2, %XHAC2
00523 R 000000 A    AC3LOC  0               /%XFAC3, %XHAC3
00524 R 000000 A    DIVSW   0
00525 R 000000 A    RSW     0
00526 R 000000 A    FPACTV  0
00527 R 000000 A    OPCODE  0
00530 R 000530 R    TEMP    .DSA TEMP       /TEMPORARY STORAGE
                    /RESULTANT HARDWARE CALCULATIONS
00531 R 000000 A    %ANSR1  0               /EXPONENT ,,, WORD 1 DPF AND SPF
00532 R 000000 A    %ANSR2  0               /WORD 2 DPF AND SPF WORD 1 DPI
00533 R 000000 A    %ANSR3  0               /WORD 3 DPF WORD 2 DPI WORD 1 SPI
00534 R 777777 A    %NA1    LAW -1          /ERROR IF: WORD 3 WHEN DPI OR WORD 2 WHEN SPI
00535 R 777777 A    %NA2    LAW -1          /ERROR IF: WORD 3 WHEN SPI
00536 R 000000 A    OVRFL1  0               /EXPONENT OVERFLOW
00537 R 000000 A    UNDFL1  0               /EXPONEND UNDERFLOW
00540 R 000000 A    OVRFLR  0               /OVERFLOW VIA ROUNDING
00541 R 000000 A    UNDFLN  0               /UNDERFLOW VIA NORMALIZING
00542 R 000000 A    DVBY00  0               /ABNORMAL DIVIDE
00543 R 000000 A    OVRFLX  0               /INTEGER OVERFLOW
00544 R 000000 A    IDVBY0  0               /INTEGER DIVISION BY ZERO
00545 R 777000 A    K1000   LAW -1000
00546 R 776000 A    KT2000  LAW -2000
                    .EJECT
```

```
                        /DIAGNOSTIC REGISTER BUFFER FILLED FROM "DRR"
00547 R 004604 R    REGADR  ,DSA (CAL)      /BMB 00-17
00550 R 004604 R            ,DSA (CAL)      /BMB 18-35
00551 R 004604 R            ,DSA (CAL)      /SC 12-17 AND IR 06-17
00552 R 004604 R            ,DSA (CAL)      /EPA 00-17
00553 R 004604 R            ,DSA (CAL)      /A SIGN AND FMA 01-17
00554 R 004604 R            ,DSA (CAL)      /FMA 18-35
00555 R 004604 R            ,DSA (CAL)      /EPB 00-17
00556 R 004604 R            ,DSA (CAL)      /B SIGN AND FMB 01-17
00557 R 004604 R            ,DSA (CAL)      /FMB 18-35
00560 R 004604 R            ,DSA (CAL)      /B SIGN AND FMQ 1-17
00561 R 004604 R            ,DSA (CAL)      /FMQ 18-35
00562 R 004604 R            ,DSA (CAL)      /ADDER 00-17
00563 R 004604 R            ,DSA (CAL)      /ADDER 18-35
00564 R 004604 R            ,DSA (CAL)      /JEA 00-17
00565 R 004604 R            ,DSA (CAL)      /STA 00-17 (SEE BELOW)
00566 R 004604 R            ,DSA (CAL)      /AR 00-17
                        /STATUS REGISTER 00-17 BIT EXPLANATION
                        /STA00                  /FP15 BUSY
                        /STA01                  /FETCH CYCLE
                        /STA02                  /OPAND CYCLE
                        /STA03                  /EXP CYCLE
                        /STA04                  /FUN CYCLE
                        /STA05                  /NOR CYCLE
                        /STA06                  /WRITE CYCLE
                        /STA07                  /INT1
                        /STA08                  /INT2
                        /STA09                  /TIME STATE 1
                        /STA10                  /TIME STATE 2
                        /STA11                  /TIME STATE 3
                        /STA12-17               /DIAG INST REG 12-17
00567 R 000567 R    RANDAT  ,DSA RANDAT
00570 R 104515 R            JMS RANGEN
00571 R 040513 R            DAC %XFAC1
00572 R 104515 R            JMS RANGEN
00573 R 040514 R            DAC %XFAC2
00574 R 104515 R            JMS RANGEN
00575 R 040515 R            DAC %XFAC3
00576 R 104515 R            JMS RANGEN
00577 R 040516 R            DAC %XHAC1
00600 R 104515 R            JMS RANGEN
00601 R 040517 R            DAC %XHAC2
00602 R 104515 R            JMS RANGEN
00603 R 040520 R            DAC %XHAC3
00604 R 200527 R            LAC OPCODE
00605 R 504575 R            AND (OP6TO9)
00606 R 544600 R            SAD (MUL)       /MULTIPLY MASK
00607 R 600630 R            JMP RANDT1
00610 R 544601 R            SAD (DIV)
00611 R 620567 R            JMP* RANDAT     /DIVISION = EXIT
00612 R 544637 R            SAD (RDIV)
00613 R 620567 R            JMP* RANDAT     /DIVISION = EXIT
                            ,EJECT
```

```
                        /ADDITION AND SUBTRACTION MASKS
00614 R 200516 R                LAC %XHAC1
00615 R 504640 R                AND (37)        /DOUBLE PRECISION
00616 R 340513 R                TAD %XFAC1
00617 R 040516 R                DAC %XHAC1
00620 R 200527 R                LAC OPCODE
00621 R 504610 R                AND (OP)
00622 R 544610 R                SAD (OP)
00623 R 620567 R                JMP* RANDAT
00624 R 200516 R                LAC %XHAC1
00625 R 504641 R                AND (27)        /SINGLE PRECISION
00626 R 040516 R                DAC %XHAC1
00627 R 620567 R                JMP* RANDAT
                        /TO MASK WHEN OPCODE IS INTEGER MULTIPLY  TO LESSEN OVERFLOWS
                        /AND INCREASE INTEGER OPERATION
00630 R 200527 R    RANDT1      LAC OPCODE
00631 R 504627 R                AND (OP!FL)
00632 R 544604 R                SAD (CAL)
00633 R 741000 A                SKP
00634 R 600644 R                JMP RANDT2
00635 R 200515 R                LAC %XFAC3      /SINGLE PRECISION INTEGER
00636 R 504642 R                AND (1777)
00637 R 040515 R                DAC %XFAC3
00640 R 200520 R                LAC %XHAC3
00641 R 504642 R                AND (1777)
00642 R 040520 R                DAC %XHAC3
00643 R 620567 R                JMP* RANDAT
00644 R 544610 R    RANDT2      SAD (OP)
00645 R 741000 A                SKP
00646 R 620567 R                JMP* RANDAT
00647 R 200514 R                LAC %XFAC2      /DOUBLE PRECISION INTEGER
00650 R 504642 R                AND (1777)
00651 R 040514 R                DAC %XFAC2
00652 R 200517 R                LAC %XHAC2
00653 R 504642 R                AND (1777)
00654 R 040517 R                DAC %XHAC2
00655 R 620567 R                JMP* RANDAT
                                .EJECT
```

```
                        /FLOATING POINT RANDOM OP-CODE BUFFER SETUP AND RANDOM "R" PRIMER
00656 R 000000 A        FPRAN   0
00657 R 204633 R                LAC (AOPCDBF)
00660 R 044547 R                DAC AUTORG      /BEGINNING OF BUFFER
00661 R 620656 R                JMP*    FPRAN
                        /LEGAL FLOATING POINLT OPERATIONS TO DATE
                        /
                        /,,,,SUBTRACT,,,,
                        /
00662 R 710400 A        OPCDBF  710400          /ISB    /SPI
00663 R 710500 A                710500          /ESB    /DPI
00664 R 710440 A                710440          /FSB    /SPF
00665 R 710450 A                710450          /URFSB  SPF UR
00666 R 710460 A                710460          /UNFSB  /SPF UN
00667 R 710470 A                710470          /UUFSB  /SPF UR UN
00670 R 710540 A                710540          /DSB    /DPF
00671 R 710550 A                710550          /URDSB  /DPF UR
00672 R 710560 A                710560          /UNDSB  /DPF UN
00673 R 710570 A                710570          /UUDSB  /DPF UP UN
                        /
                        /,,,,REVERSE SUBTRACT,,,,
                        /
00674 R 711000 A                711000          /IRS    /SPI
00675 R 711100 A                711100          /ERS    /DPI
00676 R 711040 A                711040          /FRS    /SPF
00677 R 711050 A                711050          /URFRS  /SPF UR
00700 R 711060 A                711060          /UNFRS  /SPF UN
00701 R 711070 A                711070          /UUFRS  /SPF UR UN
00702 R 711140 A                711140          /DRS    /DPF
00703 R 711150 A                711150          /URDRS  /DPF UR
00704 R 711160 A                711160          /UNDRS  /DPF UN
00705 R 711170 A                711170          /UUDRS  /DPF UR UN
                        /
                        /,,,,MULTIPLY,,,,
                        /
00706 R 711400 A                711400          /IMP    /SPI
00707 R 711500 A                711500          /EMP    /DPI
00710 R 711440 A                711440          /FMP    /SPF
00711 R 711450 A                711450          /URFMP  /SPF UR
00712 R 711460 A                711460          /UNFMP  /SPF UN
00713 R 711470 A                711470          /UUFMP  /SPF UR UN
00714 R 711540 A                711540          /DMP    /DPF
00715 R 711550 A                711550          /URDMP  /DPF UR
00716 R 711560 A                711560          /UNDMP  /DPF UN
00717 R 711570 A                711570          /UUDMP  /DPF UR UN
                                ,EJECT
```

```
                        /....DIVIDE....
                        /
00720 R 712000 A                712000          /IDV    /SPI
00721 R 712100 A                712100          /EDV    /DPI
00722 R 712040 A                712040          /FDV    /SPF
00723 R 712050 A                712050          /URFDV  /SPF UR
00724 R 712140 A                712140          /DDV    /DPF
00725 R 712150 A                712150          /URDDV  /DPF UR
                        /
                        /....REVERSE DIVIDE....
                        /
00726 R 712400 A                712400          /IRD    /SPI
00727 R 712500 A                712500          /ERD    /DPI
00730 R 712440 A                712440          /FRD    /SPF
00731 R 712450 A                712450          /URFRD  /SPF UR
00732 R 712540 A                712540          /DRD    /DPF
00733 R 712550 A                712550          /URDRD  /DPF UR
                        /
                        /....ADDITION....
                        /
00734 R 716000 A                716000          /IAD    /SPI
00735 R 716100 A                716100          /EAD    /DPI
00736 R 716040 A                716040          /FAD    /SPF
00737 R 716050 A                716050          /URFAD  /SPF UR
00740 R 716060 A                716060          /UNFAD  /SPF UN
00741 R 716070 A                716070          /UUFAD  /SPF UR UN
00742 R 716140 A                716140          /DAD    /DPF
00743 R 716150 A                716150          /URDAD  /DPF UR
00744 R 716160 A                716160          /UNDAD  /DPF UN
00745 R 716170 A        EOPCDB  716170          /UUDAD  /DPF UR UN
00746 R 000064 A                EOPCDB-OPCDBF+1
      000062 A        AXKX=EOPCDB-OPCDBF-1
                        /FROM THIS TABLE OF ADDRESSES THE FPU OPERATIONS WILL BE BLED
                        /THIS TABLE WILL BE SCRAMBLED EVERY 1000 PASSES
                        /AND RESET TO THE LISTING EVERY 2000 PASSES
                                ,EJECT
```

```
00747 R 000662 R    AOPCDBF OPCDBF
                            ,REPT AXKX,1
00750 R 000663 R            OPCDBF*1
00751 R 000664 R *R
00752 R 000665 R *R
00753 R 000666 R *R
00754 R 000667 R *R
00755 R 000670 R *R
00756 R 000671 R *R
00757 R 000672 R *R
00760 R 000673 R *R
00761 R 000674 R *R
00762 R 000675 R *R
00763 R 000676 R *R
00764 R 000677 R *R
00765 R 000700 R *R
00766 R 000701 R *R
00767 R 000702 R *R
00770 R 000703 R *R
00771 R 000704 R *R
00772 R 000705 R *R
00773 R 000706 R *R
00774 R 000707 R *R
00775 R 000710 R *R
00776 R 000711 R *R
00777 R 000712 R *R
01000 R 000713 R *R
01001 R 000714 R *R
01002 R 000715 R *R
01003 R 000716 R *R
01004 R 000717 R *R
01005 R 000720 R *R
01006 R 000721 R *R
01007 R 000722 R *R
01010 R 000723 R *R
01011 R 000724 R *R
01012 R 000725 R *R
01013 R 000726 R *R
01014 R 000727 R *R
01015 R 000730 R *R
01016 R 000731 R *R
01017 R 000732 R *R
01020 R 000733 R *R
01021 R 000734 R *R
01022 R 000735 R *R
01023 R 000736 R *R
01024 R 000737 R *R
01025 R 000740 R *R
01026 R 000741 R *R
01027 R 000742 R *R
01030 R 000743 R *R
01031 R 000744 R *R
01032 R 000745 R     AEOPCDB EOPCDB
```

```
                        /*****ADDITION*****
                        /
01033 R 200527 R    FPPADD  LAC OPCODE
01034 R 504610 R            AND (DP)
01035 R 544610 R            SAD (DP)
01036 R 601071 R            JMP DPFLAD      /DOUBLE PRECISION FLOATING AND INGEGER ADD
01037 R 601105 R            JMP SPFLAD      /SING AND INTEGERLE PRECISION FLOATING ADD
                        /
                        /*****SUBTRACTION*****
                        /
01040 R 200527 R    FPPSUB  LAC OPCODE
01041 R 504610 R            AND (DP)
01042 R 544610 R            SAD (DP)
01043 R 601073 R            JMP DPFLSB      /DOUBLE PRECISION FLOATING AND INTEGER SUBTRACT
01044 R 601107 R            JMP SPFLSB      /SING AND INTEGERLE PRECISION FLOATING SUBTRACT
                        /
                        /*****MULTIPLICATION*****
                        /
01045 R 200527 R    FPPMUL  LAC OPCODE
01046 R 504610 R            AND (DP)
01047 R 544610 R            SAD (DP)
01050 R 601075 R            JMP DPFLML      /DOUBLE PRECISION FLOATING AND INTEGER MULTIPLY
01051 R 601111 R            JMP SPFLML      /SING AND INTEGERLE PRECISION FLOATING MULTIPLY
                        /*****DIVISION*****
                        /
01052 R 200527 R    FPPDIV  LAC OPCODE
01053 R 504610 R            AND (DP)
01054 R 544610 R            SAD (DP)
01055 R 601077 R            JMP DPFLDV      /DOUBLE PRECISION FLOATING AND INTEGER DIVIDE
01056 R 601113 R            JMP SPFLDV      /SING AND INTEGERLE PRECISION FLOATING DIVIDE
                        /
                        /*****REVERSE SUBTRACTION*****
                        /
01057 R 200527 R    FPPRSB  LAC OPCODE
01060 R 504610 R            AND (DP)
01061 R 544610 R            SAD (DP)
01062 R 601101 R            JMP DPFLRS      /DOUBLE PRECISION FLOATING AND INTEGER REVERSE SUBTRACTION
01063 R 601115 R            JMP SPFLRS      /SING AND INTEGERLE PRECISION FLOATING REVERSE SUBTRACTION
                        /
                        /*****REVERSE DIVISION*****
                        /
01064 R 200527 R    FPPRDV  LAC OPCODE
01065 R 504610 R            AND (DP)
01066 R 544610 R            SAD (DP)
01067 R 601103 R            JMP DPFLRD      /DOUBLE PRECISION FLOATING AND INTEGER REVERSE DIVISION
01070 R 601117 R            JMP SPFLRD      /SING AND INTEGERLE PRECISION FLOATING REVERSE DIVISION
                            ,EJECT
```

```
                        /SOFTWARE DIRECTOR OF DOUBLE PRECISION FLOATING AND INTEGER,,,
                        /
                        /SOFTWARE MANTISSA PREVIOUSLY LOADED BY "DLAC"
                        /
                        /,,,ADDITION
                        /
01071 R 750000 A    DPFLAD  CLA
01072 R 601214 R            JMP DPFLEX
                        /
                        /,,,SUBTRACTION
                        /
01073 R 204643 R    DPFLSB  LAC (1)
01074 R 601214 R            JMP DPFLEX
                        /
                        /,,,MULTIPLICATION
                        /
01075 R 204644 R    DPFLML  LAC (2)
01076 R 601214 R            JMP DPFLEX
                        /
                        /,,,DIVISION
                        /
01077 R 204645 R    DPFLDV  LAC (3)
01100 R 601214 R            JMP DPFLEX
                        /
                        /,,,REVERSE SUBTRACTION
                        /
01101 R 204646 R    DPFLRS  LAC (4)
01102 R 601214 R            JMP DPFLEX
                        /
                        /,,,REVERSE DIVISION
                        /
01103 R 204647 R    DPFLRD  LAC (5)
01104 R 601214 R            JMP DPFLEX
                        /
                        /
                        /
                        /
                        /
                        /
                            ,EJECT
```

```
                        /SOFTWARE DIRECTOR OF SING AND INTEGERLE PRECISION FLOATING,,,
                        /
                        /SOFTWARE MANTISSA PREVIOUSLY LOADED BY "FLAC"
                        /
                        /,,,ADDITION
                        /
01105 R 204650 R    SPFLAD  LAC (6)
01106 R 601244 R            JMP SPFLEX
                        /
                        /,,,SUBTRACTION
                        /
01107 R 204651 R    SPFLSB  LAC (7)
01110 R 601244 R            JMP SPFLEX
                        /
                        /,,,MULTIPLICATION
                        /
01111 R 204620 R    SPFLML  LAC (10)
01112 R 601244 R            JMP SPFLEX
                        /
                        /,,,DIVISION
                        /
01113 R 204652 R    SPFLDV  LAC (11)
01114 R 601244 R            JMP SPFLEX
                        /
                        /,,,REVERSE SUBTRACTION
                        /
01115 R 204653 R    SPFLRS  LAC (12)
01116 R 601244 R            JMP SPFLEX
                        /
                        /,,,REVERSE DIVISION
                        /
01117 R 204654 R    SPFLRD  LAC (13)
01120 R 601244 R            JMP SPFLEX
                        /
                        /
                        /
                        /
                        /
                        /
                        /
                            ,EJECT
```

```
                        /SINGLE AND DOUBLE PRECISION INTEGER SOFTWARE EXECUTIONER
01121 R 204563 R    DPIEX   LAC SOFTEX
01122 R 041125 R            DAC DOIEX
01123 R 103722 R            JMS DLAC        /LOAD 3 RANDOM WORDS
01124 R 000513 R            ,DSA %XFAC1     /AND FLOAT IF INTEGER
01125 R 000000 A    DOIEX   CAL
01126 R 000516 R            ,DSA %XHAC1
01127 R 200527 R            LAC OPCODE
01130 R 504575 R            AND (OP6TO9)
01131 R 544600 R            SAD (MUL)
01132 R 741000 A            SKP
01133 R 601157 R            JMP DPIEX1
                        /OPERATION IS INTEGER MULTIPLY
                        /MUST SWAP AND SHIFT ONE RIGHT BECAUSE OF SOFTWARE FINISH POINT
                        /MY FMQ = %FAC2 AND %FAC3 WITH THE HIGH ORDER BITS
                        /IF %FAC2 OR %FAC3 NOT = 0, +1 TO OVRFLX
                        /IF = 0, SWAP %FAC2, 3 WITH %FAC4, 4A
01134 R 200527 R            LAC OPCODE
01135 R 504610 R            AND (DP)
01136 R 544610 R            SAD (DP)
01137 R 601143 R            JMP ,+4
01140 R 203707 R            LAC %FAC4
01141 R 740200 A            SZA
01142 R 440543 R            ISZ OVRFLX
01143 R 203705 R            LAC %FAC2
01144 R 740200 A            SZA
01145 R 440543 R            ISZ OVRFLX
01146 R 203706 R            LAC %FAC3
01147 R 740200 A            SZA
01150 R 440543 R            ISZ OVRFLX
01151 R 203707 R            LAC %FAC4
01152 R 744020 A            RCR
01153 R 043705 R            DAC %FAC2
01154 R 204565 R            LAC %FAC4A
01155 R 740020 A            RAR
01156 R 043706 R            DAC %FAC3
                            ,EJECT
```

```
                        /IF RESULT IS ZERO - CLEAR EXPONENT
01157 R 203705 R    DPIEX1  LAC %FAC2
01160 R 543706 R            SAD %FAC3
01161 R 740200 A            SZA
01162 R 601200 R            JMP DPIEX2
01163 R 143704 R            DZM %FAC1
01164 R 143707 R            DZM %FAC4
01165 R 143721 R            DZM %FSIGN
                        /IF INTEGER DIVISION BY ZERO IS EXPECTED AND THE FMA IS NOT 000000
                        /THEN EXPECT OVERFLOW #30 ALSO WHEN SPI STORE
01166 R 200544 R            LAC IDVBY0
01167 R 741200 A            SNA
01170 R 600127 R            JMP FPUXA1
01171 R 200515 R            LAC %XFAC3
01172 R 740200 A            SZA
01173 R 440543 R            ISZ OVRFLX      /BECAUSE %XFAC3 IS NORMALIZED IN FPU HARDWARE
01174 R 200520 R            LAC %XHAC3
01175 R 740200 A            SZA
01176 R 440543 R            ISZ     OVRFLX
01177 R 600127 R            JMP FPUXA1
                        /COMBINE SIGN BIT WITH AC
01200 R 102102 R    DPIEX2  JMS %FIX,       /FIX BACK TO INTEGER
01201 R 203721 R            LAC %FSIGN
01202 R 744100 A            SMA!CLL
01203 R 600127 R            JMP FPUXA1
01204 R 203706 R            LAC %FAC3
01205 R 740031 A            TCA
01206 R 043706 R            DAC %FAC3
01207 R 203705 R            LAC %FAC2
01210 R 741401 A            CMA!SZL
01211 R 344643 R            TAD (1)
01212 R 043705 R            DAC %FAC2
01213 R 600127 R            JMP FPUXA1
                            ,EJECT
```

```
                        /DOUBLE PRECISION FLOATING SOFTWARE EXECUTIONER
01214 R 341415 R    DPFLEX  TAD SOFCOD
01215 R 041226 R            DAC DOFEX
01216 R 044563 R            DAC SOFTEX#
01217 R 200527 R            LAC OPCODE
01220 R 504603 R            AND (FL)
01221 R 544604 R            SAD (CAL)
01222 R 601121 R            JMP DPIEX
01223 R 101310 R            JMS SOFNR
01224 R 103722 R            JMS     DLAC
01225 R 000513 R            ,DSA    %XFAC1
01226 R 000000 A    DOFEX   CAL             /SOFTWARE CALCULATIONS
01227 R 000516 R            ,DSA %XHAC1
                        /IF RESULT IS ZERO - CLEAR EXPONENT
01230 R 203705 R            LAC %FAC2
01231 R 543706 R            SAD %FAC3
01232 R 740200 A            SZA
01233 R 601240 R            JMP ,+5
01234 R 143704 R            DZM %FAC1
01235 R 143707 R            DZM %FAC4
01236 R 143721 R            DZM %FSIGN
01237 R 600127 R            JMP FPUXA1
                        /COMBINE SIGN BIT WITH AC
01240 R 203721 R            LAC %FSIGN
01241 R 343705 R            TAD %FAC2
01242 R 043705 R            DAC %FAC2
01243 R 600127 R            JMP FPUXA1      /SOFTWARE RESULT LEFT IN %FAC1 TO %FAC3
                        ,EJECT
```

```
                        /
                        /SINGLE PRECISION FLOATING SOFTWARE EXECUTIONER
                        /
01244 R 341415 R    SPFLEX  TAD SOFCOD
01245 R 041262 R            DAC SOFEX
01246 R 044563 R            DAC SOFTEX
01247 R 200527 R            LAC OPCODE
01250 R 504603 R            AND (FL)
01251 R 544603 R            SAD (FL)
01252 R 601257 R            JMP .+5
01253 R 777772 A            LAW -6          /BACK UP
01254 R 344563 R            TAD SOFTEX      /POINTER FOR
01255 R 044563 R            DAC SOFTEX      /DP SOFTWARE
01256 R 601121 R            JMP DPIEX       /AND GO TO INTEGER
01257 R 101310 R            JMS SOFNR
01260 R 104071 R            JMS     FLAC
01261 R 000513 R            .DSA    %XFAC1
01262 R 000000 A    SOFEX   CAL             /SOFTWARE CALCULATIONS
01263 R 000516 R            .DSA %XHAC1
                        /IF RESULT IS ZERO - CLEAR EXPONENT
01264 R 203705 R            LAC %FAC2
01265 R 543706 R            SAD %FAC3
01266 R 740200 A            SZA
01267 R 601274 R            JMP .+5
01270 R 143704 R            DZM %FAC1
01271 R 143707 R            DZM %FAC4
01272 R 143721 R            DZM %FSIGN
01273 R 600127 R            JMP FPUXA1
                        /COMBINE SIGN BIT WITH AC
01274 R 203721 R            LAC %FSIGN
01275 R 343705 R            TAD %FAC2
01276 R 043705 R            DAC %FAC2
01277 R 203704 R            LAC %FAC1
01300 R 504655 R            AND (777)
01301 R 043704 R            DAC %FAC1
01302 R 203706 R            LAC %FAC3
01303 R 504630 R            AND (777000)
01304 R 343704 R            TAD %FAC1
01305 R 043704 R            DAC %FAC1
01306 R 143706 R            DZM %FAC3
01307 R 600127 R            JMP FPUXA1      /SOFTWARE RESULT LEFT IN %FAC1 TO %FAC3
                            .EJECT
```

```
                        /SOFTWARE NORMILIZER FOR LOAD FUNCTION CONFIRMATION
01310 R 001310 R    SOFNR   ,DSA ,
01311 R 200527 R            LAC OPCODE
01312 R 504575 R            AND (OP6TO9)
01313 R 544601 R            SAD (DIV)
01314 R 741000 A            SKP
01315 R 544637 R            SAD (RDIV)
01316 R 601326 R            JMP SOF1
01317 R 200527 R            LAC OPCODE
01320 R 504656 R            AND (UN!FL)
01321 R 544604 R            SAD (CAL)
01322 R 621310 R            JMP* SOFNR
01323 R 504657 R            AND (UN)
01324 R 544657 R            SAD (UN)
01325 R 621310 R            JMP* SOFNR
01326 R 200527 R    SOF1    LAC OPCODE
01327 R 504610 R            AND (DP)
01330 R 544610 R            SAD (DP)
01331 R 601335 R            JMP SOF2
01332 R 200513 R            LAC %XFAC1
01333 R 504630 R            AND (777000)
01334 R 040515 R            DAC %XFAC3
01335 R 200514 R    SOF2    LAC %XFAC2
01336 R 504660 R            AND (377777)
01337 R 540515 R            SAD %XFAC3
01340 R 740200 A            SZA
01341 R 741000 A            SKP
01342 R 621310 R            JMP* SOFNR
01343 R 200514 R            LAC %XFAC2
01344 R 504626 R            AND (400000)
01345 R 043721 R            DAC %FSIGN
01346 R 200514 R            LAC %XFAC2
01347 R 504660 R            AND (377777)
01350 R 040514 R            DAC %XFAC2
01351 R 200514 R    FNXSA   LAC %XFAC2
01352 R 504661 R            AND (LAC)
01353 R 740200 A            SZA
01354 R 601405 R            JMP SOFNRX
01355 R 200515 R            LAC %XFAC3
01356 R 744010 A            RCL
01357 R 040515 R            DAC %XFAC3
01360 R 200514 R            LAC %XFAC2
01361 R 740010 A            RAL
01362 R 040514 R            DAC %XFAC2
01363 R 200527 R            LAC OPCODE
01364 R 504610 R            AND (DP)
01365 R 544610 R            SAD (DP)
01366 R 601401 R            JMP SOF3
                            ,EJECT
```

```
01367 R 200513 R            LAC %XFAC1
01370 R 504630 R            AND (777000)
01371 R 040530 R            DAC TEMP
01372 R 200513 R            LAC %XFAC1
01373 R 504655 R            AND (777)
01374 R 344631 R            TAD (-1)
01375 R 504655 R            AND (777)
01376 R 340530 R            TAD TEMP
01377 R 040513 R            DAC %XFAC1
01400 R 601351 R            JMP FNXSA
01401 R 777777 A    SOF3    LAW -1
01402 R 340513 R            TAD %XFAC1
01403 R 040513 R            DAC %XFAC1
01404 R 601351 R            JMP FNXSA
01405 R 200514 R    SOFNRX  LAC %XFAC2
01406 R 343721 R            TAD %FSIGN
01407 R 040514 R            DAC %XFAC2
01410 R 200527 R            LAC OPCODE
01411 R 504610 R            AND (DP)
01412 R 544604 R            SAD (CAL)
01413 R 140515 R            DZM %XFAC3
01414 R 621310 R            JMP* SOFNR
                    /SOFTWARE CALCULATION SUBROUTINES
                    /
01415 R 121416 R    SOFCOD  JMS* .+1
                    /
                    /DOUBLE PRECISION FLOATING AND SP AND DP INTEGER
                    /
01416 R 004055 R            .DSA DFAD       /ADDITION
01417 R 003771 R            .DSA DSUB       /SUBTRACTION
01420 R 004000 R            .DSA DMPY       /MULTIPLICATION
01421 R 004025 R            .DSA DDVD       /DIVISION
01422 R 004042 R            .DSA DSBR       /REVERSE SUBTRACTION
01423 R 004013 R            .DSA DDVR       /REVERSE DIVISION
                    /
                    /SINGLE PRECISION FLOATING
                    /
01424 R 004135 R            .DSA FAD        /ADDITION
01425 R 004151 R            .DSA FSUB       /SUBTRACTION
01426 R 004160 R            .DSA FMPY       /MULTIPLICATION
01427 R 004173 R            .DSA FDVD       /DIVISION
01430 R 004210 R            .DSA FSBR       /REVERSE SUBTRACTION
01431 R 004223 R            .DSA FDVR       /REVERSE DIVISION
                            .EJECT
```

```
                        /INTERRUPT EXCEPTION EXIT ADDRESS
01432 R 000000 A    XJEA    0
01433 R 601442 R            JMP OVFLO       /OVERFLOW
01434 R 000000 A            0
01435 R 601454 R            JMP UNFLO       /UNDERFLOW
01436 R 000000 A            0
01437 R 601465 R            JMP DVBY0       /ABNORMAL DIVISION BY ZERO
01440 R 000000 A            0
01441 R 601476 R            JMP NEXMEM      /ILLEGAL MEMORY REFERENCE
                        /
                        /OVERFLOW
                        /
01442 R 777777 A    OVFLO   LAW -1
01443 R 341432 R            TAD XJEA        /DECREMENT RETURN
01444 R 041432 R            DAC XJEA        /PC BY ONE
01445 R 044560 R            DAC JEADRS#
01446 R 200540 R            LAC OVRFLR      /FROM ROUNDING
01447 R 340536 R            TAD OVRFL1
01450 R 340543 R            TAD OVRFLX
01451 R 750200 A            SZA!CLA
01452 R 621432 R            JMP* XJEA
                        /
                        /UNEXPECTED HARDWARE OVERFLOW:
                        /
                        / 1, FROM ROUNDING
                        / 2, FROM EXPONENT VIA MULTIPLY
                        /
01453 R 601502 R            JMP XFLOW
                        /
                        /UNDERFLOW
                        /
01454 R 777777 A    UNFLO   LAW -1
01455 R 341434 R            TAD XJEA+2      /DITTO
01456 R 041434 R            DAC XJEA+2
01457 R 044560 R            DAC JEADRS
01460 R 200541 R            LAC UNDFLN      /FROM NORMALIZATIN
01461 R 340537 R            TAD UNDFL1
01462 R 750200 A            SZA!CLA
01463 R 621434 R            JMP* XJEA+2     /UNDERFLOW EXPECTED
                        /
                        /UNEXPECTED HARDWARE UNDERFLOW
                        /
                        / 3, FROM NORMALIZATION
                        / 4, FROM EXPONENT VIA MULTIPLY
                        /
01464 R 601502 R            JMP XFLOW
                            .EJECT
```

```
                        /DIVISION BY ZERO
                        /
01465 R 777777 A    DVBY0   LAW -1
01466 R 341436 R            TAD XJEA+4
01467 R 041436 R            DAC XJEA+4
01470 R 044560 R            DAC JEADRS
01471 R 200542 R            LAC DVBY00
01472 R 340544 R            TAD IDVBY0
01473 R 750200 A            SZA!CLA
01474 R 621436 R            JMP* XJEA+4     /ABNORMAL DIVISION EXPECTED
                        /
                        /UNEXPECTED HARDWARE ABNORMAL DIVISION
                        /
01475 R 601502 R            JMP XFLOW
                        /
                        /ILLEGAL MEMORY REFERENCED
                        /
01476 R 740040 A    NEXMEM  HLT
01477 R 601476 R            JMP .-1
01500 R 740040 A    TNEXMM  HLT
01501 R 601500 R            JMP .-1
01502 R 200540 R    XFLOW   LAC OVRFLR
01503 R 544604 R            SAD (CAL)
01504 R 601507 R            JMP .+3
01505 R 204643 R            LAC (1)         / 1, OVERFLOW FROM ROUNDING
01506 R 604344 R            JMP ERROR3      /UNEXPECTED JEA BREAK
01507 R 200536 R            LAC OVRFL1
01510 R 544604 R            SAD (CAL)
01511 R 601514 R            JMP .+3
01512 R 204644 R            LAC (2)         / 2, EXPONENT OVERFLOW
01513 R 604344 R            JMP ERROR3
01514 R 200541 R            LAC UNDFLN
01515 R 544604 R            SAD (CAL)
01516 R 601521 R            JMP .+3
01517 R 204645 R            LAC (3)         / 3, UNDERFLOW FROM NORMILIZATION
01520 R 604344 R            JMP ERROR3
01521 R 200537 R            LAC UNDFL1
01522 R 544604 R            SAD (CAL)
01523 R 601526 R            JMP .+3
01524 R 204646 R            LAC (4)         / 4, EXPONENT UNDERFLOW
01525 R 604344 R            JMP ERROR3
01526 R 200542 R            LAC DVBY00
01527 R 544604 R            SAD (CAL)
01530 R 601533 R            JMP .+3
01531 R 204647 R            LAC (5)         / 5, FLOATING ABNORMAL DIVIDE
01532 R 604344 R            JMP ERROR3
01533 R 200544 R            LAC IDVBY0
01534 R 544604 R            SAD (CAL)
01535 R 604344 R            JMP ERROR3
01536 R 204662 R            LAC (25)        / 25, INTEGER DIVISION BY ZERO
01537 R 604344 R            JMP ERROR3
                        .EJECT
```

```
                        /EXPECTED INTERRUPTS FROM,,,BUT NONE OCCURED
01540 R 204652 R    EINT11  LAC (11)          / 11, OVERFLOW FROM ROUNDING
01541 R 741000 A            SKP
01542 R 204653 R    EINT12  LAC (12)          / 12, EXPONENT OVERFLOW
01543 R 741000 A            SKP
01544 R 204625 R    EINT30  LAC (30)          / 30, INTEGER OVERFLOW
01545 R 604356 R            JMP ERROR4        /EXPECTED JEA BREAK BUT NONE OCCURRED
01546 R 204654 R    EINT13  LAC (13)          / 13, UNDERFLOW FROM NORNALIZATION
01547 R 741000 A            SKP
01550 R 204663 R    EINT14  LAC (14)          / 14, EXP UNDFLOW
01551 R 604356 R            JMP ERROR4        /EXPECTED JEA BREAK BUT NONE OCCURRED
01552 R 204664 R    EINT15  LAC (15)          / 15, FLOATING ABNORMAL DIVISION
01553 R 741000 A            SKP
01554 R 204665 R    EINT35  LAC (35)          / 35, INTEGER DIVISION BY ZERO
01555 R 604356 R            JMP ERROR4        /EXPECTED JEA BREAK BUT NONE OCCURRED
                        /SINGLE PRECISION FLOATING RESTRICTIONS FOR "RANDEX"
01556 R 001556 R    IFREST  ,DSA ,
01557 R 777777 A    SPFRES  LAW -1
01560 R 040534 R            DAC %NA1          /PREP FOR POSSIBLE
01561 R 040535 R            DAC %NA2          /DATA FORMAT ERROR
01562 R 200527 R            LAC OPCODE
01563 R 504627 R            AND (DP!FL)
01564 R 544603 R            SAD (FL)
01565 R 741000 A            SKP
01566 R 601606 R            JMP SPIRES
01567 R 140515 R            DZM %XFAC3
01570 R 140533 R            DZM %ANSR3
01571 R 140520 R            DZM %XHAC3
01572 R 621556 R            JMP* IFREST
                        /DOUBLE PRECISION FLOATING RESTRICTIONS FOR "RANDEX"
01573 R 000000 A    DPFRES  0
01574 R 200527 R            LAC OPCODE
01575 R 504627 R            AND (DP!FL)
01576 R 544627 R            SAD (DP!FL)
01577 R 741000 A            SKP
01600 R 621573 R            JMP* DPFRES
01601 R 200351 R            LAC FPUST
01602 R 504666 R            AND (-UR-1)       /ALWAYS STORE UR FOR DP FL
01603 R 344620 R            TAD (UR)
01604 R 040351 R            DAC FPUST
01605 R 621573 R            JMP* DPFRES
                        /SINGLE PRECISION INTEGER RESTRICTINS FOR "RANDEX"
01606 R 544604 R    SPIRES  SAD (CAL)
01607 R 741000 A            SKP
01610 R 601625 R            JMP DPIRES
01611 R 140514 R            DZM %XFAC2
01612 R 140532 R            DZM %ANSR2
01613 R 143705 R            DZM %FAC2
01614 R 140517 R            DZM %XHAC2
01615 R 140513 R            DZM %XFAC1
01616 R 140531 R            DZM %ANSR1
01617 R 140516 R            DZM %XHAC1
01620 R 143704 R            DZM %FAC1
```

```
                        /IF MANTISSA = 400000, +1 TO OVRFLX
                        /BECAUSE BOUNDRY LIMIT EXCEEDED AND OVERFLOW WILL ON CFPUST
01621 R 200515 R                LAC %XFAC3
01622 R 544626 R                SAD (400000)
01623 R 440543 R                ISZ OVRFLX
01624 R 621556 R                JMP* IFREST
                        /DOUBLE PRECISION INTEGER RESTRICTIOS FOR "RANDEX"
01625 R 544610 R        DPIRES  SAD (DP)
01626 R 741000 A                SKP
01627 R 621556 R                JMP* IFREST
01630 R 140513 R                DZM %XFAC1
01631 R 140531 R                DZM %ANSR1
01632 R 140516 R                DZM %XHAC1
01633 R 143704 R                DZM %FAC1
01634 R 621556 R                JMP* IFREST
                        /IF ABNORMAL DIVISION DO NOT BRANCH
01635 R 000000 A        TFBRA   0
01636 R 200542 R                LAC DVBY00
01637 R 340544 R                TAD IDVBY0
01640 R 740200 A                SZA
01641 R 621635 R                JMP* TFBRA      /EXIT = NO TEST
                        /IF INTEGER = EXIT = NO BRANCHING HERE
01642 R 200527 R                LAC OPCODE
01643 R 504603 R                AND (FL)
01644 R 544604 R                SAD (CAL)
01645 R 621635 R                JMP* TFBRA
                        /IF SPF-UR EXIT
01646 R 200527 R                LAC OPCODE
01647 R 504667 R                AND (DP!FL!UR!UN)
01650 R 544670 R                SAD (FL!UR)
01651 R 621635 R                JMP* TFBRA      /EXIT
                        /IF SPF-UR EXIT
01652 R 200527 R                LAC OPCODE
01653 R 504667 R                AND (DP!FL!UR!UN)
01654 R 544670 R                SAD (FL!UR)
01655 R 621635 R                JMP* TFBRA              /EXIT
                        /ERRORS OCCURING NOW ARE BRANCH FUNCTION FAILURES
01656 R 777777 A                LAW -1
01657 R 044554 R                DAC COMPLD
                        /BRANCH IF ZERO FMA
01660 R 144553 R        TBZA    DZM BRANCH#
01661 R 200527 R                LAC OPCODE
01662 R 504610 R                AND (DP)
01663 R 544610 R                SAD (DP)
01664 R 601670 R                JMP TBZA1
01665 R 203704 R                LAC %FAC1
01666 R 504630 R                AND (777000)
01667 R 741000 A                SKP
                                ,EJECT
```

```
01670 R 203706 R    TBZA1   LAC %FAC3
01671 R 040530 R            DAC TEMP
01672 R 203705 R            LAC %FAC2
01673 R 504660 R            AND (377777)
01674 R 540530 R            SAD TEMP
01675 R 741000 A            SKP
01676 R 601701 R            JMP .+3
01677 R 741200 A            SNA
01700 R 444553 R            ISZ BRANCH
01701 R 204671 R            LAC (BZA)
01702 R 101752 R            JMS SBRAN
                    /BRANCH IF NEGATIVE FMA
01703 R 144553 R    TBMA    DZM BRANCH
01704 R 203721 R            LAC %FSIGN
01705 R 741100 A            SPA
01706 R 444553 R            ISZ BRANCH
01707 R 204672 R            LAC (BMA)
01710 R 101752 R            JMS SBRAN
                    /BRANCH IF FMA < OR = 0
01711 R 144553 R    TBLE    DZM BRANCH
01712 R 203705 R            LAC %FAC2
01713 R 741100 A            SPA
01714 R 444553 R            ISZ BRANCH
01715 R 504660 R            AND (377777)
01716 R 540530 R            SAD TEMP
01717 R 741000 A            SKP
01720 R 601723 R            JMP .+3
01721 R 741200 A            SNA
01722 R 444553 R            ISZ BRANCH
01723 R 204673 R            LAC (BLE)
01724 R 101752 R            JMS SBRAN
                    /BRANCH IF POSITIVE FMA
01725 R 144553 R    TBPA    DZM BRANCH
01726 R 203721 R            LAC %FSIGN
01727 R 740100 A            SMA
01730 R 444553 R            ISZ BRANCH
01731 R 204674 R            LAC (BPA)
01732 R 101752 R            JMS SBRAN
                    /BRANCH UNCONDITINAL
01733 R 444553 R    TBRU    ISZ BRANCH
01734 R 204675 R            LAC (BRU)
01735 R 101752 R            JMS SBRAN
                            .EJECT
```

```
                        /BRANCH IF FMA NON-ZERO
01736 R 144553 R    TBNA    DZM BRANCH
01737 R 203705 R            LAC %FAC2
01740 R 504660 R            AND (377777)
01741 R 540530 R            SAD TEMP
01742 R 740200 A            SZA
01743 R 444553 R            ISZ BRANCH
01744 R 204676 R            LAC (BNA)
01745 R 101752 R            JMS SBRAN
                        /BRANCH IF GUARD BIT IS SET
01746 R 144553 R    TBAC    DZM BRANCH
01747 R 204677 R            LAC (BAC)       /GUARD BIT SHOULD ALWAYS BE CLEARED FROM STORE INSTRUCTION
01750 R 101752 R            JMS SBRAN               /THEREFORE BNA SHOULD NEVER BRANCH
01751 R 621635 R            JMP* TFBRA      /EXIT BRANCH TESTING
01752 R 000000 A    SBRAN   0
01753 R 041772 R            DAC YBRANCH
01754 R 041764 R            DAC NBRANCH
01755 R 204553 R            LAC BRANCH      /IF = 1 BRANCH
01756 R 741200 A            SNA             /IF = 0 NO BRANCH
01757 R 101762 R            JMS NOBRAN      /BRANCH = 0; NO BRANCG
01760 R 101771 R            JMS BRAN        /BRANCH = 1; BRANCH
01761 R 621752 R            JMP* SBRAN      /EXIT BRANCHES
01762 R 000000 A    NOBRAN  0
01763 R 441762 R            ISZ NOBRAN      /+1 TO RETURN
01764 R 000000 A    NBRANCH CAL             /OCTAL EQUIVALENT OF BRANCH INSTRUCTION
01765 R 001767 R            .+2
01766 R 621762 R            JMP* NOBRAN     /EXIT; OK
01767 R 604401 R            JMP ERROR6      /A BRANCH OCCURRED WHEN UNEXPECTED
01770 R 621762 R            JMP* NOBRAN
01771 R 000000 A    BRAN    0
01772 R 000000 A    YBRANCH CAL             /OCTAL EQUIV
01773 R 001775 R            .+2             /EXIT; OK
01774 R 604414 R            JMP ERROR7      /NO BRANCH OCCURRED WHEN EXPECTED
01775 R 621771 R            JMP* BRAN
                        /RESET ADDRESS BUFFER TO ORIGINAL SETTING
01776 R 001776 R    INITADR .DSA .
01777 R 776000 A            LAW -2000
02000 R 040546 R            DAC KT2000
02001 R 204633 R            LAC (AOPCDBF)
02002 R 044547 R            DAC AUTORG
02003 R 204700 R            LAC (OPCDBF)
02004 R 741000 A            SKP
02005 R 344643 R            TAD (1)
02006 R 064547 R            DAC* AUTORG
02007 R 444547 R            ISZ AUTORG
02010 R 541032 R            SAD AEOPCDB
02011 R 621776 R            JMP* INITADR
02012 R 602005 R            JMP .-5
                            .EJECT
```

```
                        ,TITLE SUBTITLE FPOINT-NON
                    /GENERAL FLOATING POINT ARITHMETIC
                    /
                    /       FLOATING POINT MATH (NON-EAE)
                    /
                    /
                    /       %FLOT,          FLOAT INTEGER TO FLOATING ACCUMULATOR,
                    /       %FUNF,          UNFLOAT SECOND WORD OF FLOATING ACCUMULATOR TO A-REG,
                    /       %FNEG,          NEGATE THE FLOATING ACCUMULATOR
                    /       %FG,            SHORT GET ARGUMENT (ADDRESS)
                    /       %FNOR,          NORMALIZE THE FLOATING ACCUMULATOR
                    /       %FH,            HOLD THE FLOATING ACCUMULATOR
                    /       %FIR,           ROUND RESULT AND INSERT SIGN
                    /       %FS,            SIGN CONTROL
                    /       %ANEG,          NEGATE A-REG,
                    /       %SWICH          SWITCH THE FLOATING AND HELD ACC'S,
                    /
                    /       %FA,            GENERAL FLOATING ADD
                    /       %FM,            GENERAL FLOATING MULTIPLY
                    /       %FD,            GENERAL FLOATING DIVIDE
                    /
                    /
                    /TERMS:
                    /       %FAC1           EXPONENT-FLOATING POINT ACCUMULATOR
                    /       %FAC2           SIGN+HIGH-ORDER MANTISSA-FLT ACC
                    /       %FAC3           LOW-ORDER MANTISSA-FLT ACC
                    /
                    /       %HAC1           EXPONENT-HELD ACCUMULATOR
                    /       %HAC2           SIGN+HIGH-ORDER MANTISSA-HELD ACC
                    /       %HAC3           LOW-ORDER MANTISSA
                    /       A-REG           HARDWARE ACCUMULATOR(AC)
                    /
                    /
300000 A            ADD15=300000
                            ,EJECT
```

```
                        /--------------------
                        /
                        /NEGATE THE A-REG
                        /
                        /
02013 R 000000 A    %ANEG,  0
02014 R 744001 A            CMA!CLL
02015 R 344643 R            TAD     (1
02016 R 622013 R            JMP*    %ANEG,
                        /NEGATE THE FSIGN
02017 R 000000 A    %SNEG,  0
02020 R 203721 R            LAC %FSIGN
02021 R 244626 R            XOR (400000)
02022 R 043721 R            DAC %FSIGN
02023 R 622017 R            JMP* %SNEG,
                        /FLOAT INTEGER TO FLOATING ACCUMULATOR (%FLOT,)
                        /CALLING SEQUENCE:
                        /       JMS     %FLOT,  /SUBR CALL(INTEGER IN A-REG)
                        /       NEXT INSTRUCTION        /RETURN(INTEGER NORMALIZED IN FLT ACC)
                        /                       /A-REG IS UNCHANGED
02024 R 000000 A    %FLOT,  0
02025 R 200527 R            LAC OPCODE
02026 R 504603 R            AND (FL)
02027 R 544603 R            SAD (FL)
02030 R 622024 R            JMP* %FLOT,
02031 R 143704 R            DZM %FAC1
02032 R 200527 R            LAC OPCODE
02033 R 504610 R            AND (DP)
02034 R 544610 R            SAD (DP)
02035 R 602040 R            JMP .+3
02036 R 203706 R            LAC %FAC3
02037 R 602043 R            JMP .+4
02040 R 203705 R            LAC %FAC2
02041 R 504660 R            AND (377777)
02042 R 543706 R            SAD %FAC3
02043 R 740200 A            SZA
02044 R 602050 R            JMP .+4
02045 R 143705 R            DZM %FAC2       /ZERO SIGN
02046 R 143706 R            DZM %FAC3
02047 R 622024 R            JMP* %FLOT,
02050 R 200527 R            LAC OPCODE
02051 R 504610 R            AND (DP)
02052 R 544610 R            SAD (DP)
02053 R 602065 R            JMP %FLOT1
02054 R 203706 R            LAC %FAC3
02055 R 504626 R            AND (400000)
02056 R 043705 R            DAC %FAC2
02057 R 740100 A            SMA
02060 R 622024 R            JMP* %FLOT,
02061 R 203706 R            LAC %FAC3
02062 R 740031 A            TCA
02063 R 043706 R            DAC %FAC3
02064 R 622024 R            JMP* %FLOT,
```

```
02065 R 203705 R    %FLOT1  LAC %FAC2
02066 R 744100 A            SMA!CLL
02067 R 622024 R            JMP* %FLOT,
02070 R 203706 R    %FLOT2  LAC %FAC3
02071 R 740031 A            TCA
02072 R 043706 R            DAC %FAC3
02073 R 203705 R            LAC %FAC2
02074 R 741401 A            CMA!SZL
02075 R 344643 R            TAD (1)
02076 R 504660 R            AND (377777)
02077 R 344626 R            TAD (400000)
02100 R 043705 R            DAC %FAC2
02101 R 622024 R            JMP* %FLOT,
                    /FIX THE FLOATING AC
02102 R 000000 A    %FIX,   0
02103 R 203704 R            LAC %FAC1
02104 R 741200 A            SNA
02105 R 602121 R            JMP %FIXB       /NO REALIGNMENT NECESSARY
02106 R 203707 R            LAC %FAC4
02107 R 744010 A            RCL
02110 R 043707 R            DAC %FAC4
02111 R 044557 R            DAC GUARD
02112 R 203706 R            LAC %FAC3
02113 R 740010 A            RAL
02114 R 043706 R            DAC %FAC3
02115 R 203705 R            LAC %FAC2
02116 R 740010 A            RAL
02117 R 043705 R            DAC %FAC2
02120 R 143704 R            DZM %FAC1
02121 R 200527 R    %FIXB   LAC OPCODE
02122 R 504575 R            AND (OP6TO9)
02123 R 544637 R            SAD (RDIV)
02124 R 622102 R            JMP* %FIX,
02125 R 544601 R            SAD (DIV)
02126 R 622102 R            JMP* %FIX,
02127 R 200527 R            LAC OPCODE
02130 R 504610 R            AND (DP)
02131 R 544610 R            SAD (DP)
02132 R 602135 R            JMP .+3
02133 R 203706 R            LAC %FAC3
02134 R 741000 A            SKP
02135 R 203705 R            LAC %FAC2
02136 R 741100 A            SPA
02137 R 440543 R            ISZ OVRFLX      /OVERFLOW OCCURED
02140 R 504660 R            AND (377777)
02141 R 622102 R            JMP* %FIX,
                    /NEGATE THE FLOATING ACCUMULATOR (%FNEG,)
                    /CALLING SEQUENCE:
                    /       JMS     %FNEG   /SUBR CALL (CHANGE SIGN OF NORMALIZED FLT ACC,)
                    /       NEXT INSTRUCTION        /RETURN WITH SIGN OF FLT CC CHARGED,
02142 R 000000 A    %FNEG,  0               /ENTRY-EXIT
02143 R 203705 R            LAC     %FAC2   /SIGN WORD OF FLOATING ACCUMULATOR
02144 R 543706 R            SAD %FAC3
```

```
02145 R 740200 A            SZA             /IF ABSOLUTE ZERO DO NOT NEGATE
02146 R 244626 R            XOR     (400000 /CHANGE SIGN
02147 R 043705 R            DAC     %FAC2   /RESTORE
02150 R 622142 R            JMP*    %FNEG,  /EXIT
                      /
                      /--------------------
                      /
                      /SHORT GET ARGUMENT ROUTINE (%FG,)
                      /CALLING SEQUENCE;
                      /               (PC-1)  0               /ADDRESS OF ARG IN USERS PROG,
                      /               (PC)    JMS     %FG,    /SUBR CALL
                      /               (PC+1)  0               /(STORE ADDRESS HERE)
                      /               (PC+2)  NEXT INSTRUCTION        /RETURN HERE
                      /
                      /SUBROUTINE %FG, GOES BACK TWO LEVELS (THREE IF INDIRECT)
                      /TO GET THE ADDRESS OF AN ARGUMENT AND STORES IT AT PC+1
                      /OF THE CALLING PROGRAM,
                      /
                      /
02151 R 000000 A    %FG,    0               /ENTRY-EXIT
02152 R 202151 R            LAC     %FG,    /MASK OF POSSIBLE
02153 R 504701 R            AND     (77777  /SIGN BIT SET FROM
02154 R 042151 R            DAC     %FG,    /LINK DURING JMS
02155 R 777776 A            LAW     -2      /GET CALLERS EXIT
02156 R 342151 R            TAD     %FG,    /AC=(PC-1) FROM CALLING PROGRAM
02157 R 043720 R            DAC     %J5     /SAVE FIRST LEVEL
                      /SECOND LEVEL
02160 R 223720 R            LAC*    %J5     /GET (ARG ADDR) ADDR
02161 R 463720 R            ISZ*    %J5     /BUMP CALLERS EXIT
02162 R 043720 R            DAC     %J5     /SAVE ARG ADDR
02163 R 223720 R            LAC*    %J5     /GET ARG ADDR
02164 R 043720 R            DAC     %J5     /END OF 2ND LEVEL
02165 R 741100 A            SPA             /IF 0 BIT IS ON ADDRESS IS INDIRECT
02166 R 223720 R            LAC*    %J5     /ACCESS AGAIN
02167 R 062151 R            DAC*    %FG,    /STORE ARG ADDR IN CALLING POINT
02170 R 442151 R    %FGEX,  ISZ     %FG,    /BUMP EXIT
02171 R 622151 R            JMP*    %FG,    /EXIT
                      /
                      /---------------------
                      /
                      /NORMALIZE FLOATING ACCUMULATOR (%FNOR,)
                      /CALLING SEQUENCE;
                      /       JMS     %FNOR   /SUBR CALL
                      /       NEXT INSTRUCTION        /SUBR RETURN
                      /
                      /
                      /THE FLOATING POINT ACCUMULATOR IS SHIFTED LEFT UNTIL BIT 1 IS
                      /ON, BEFORE ANY SHIFTING THE NUMBER IS CHECKED FOR
                      /ZERO, IF IT IS ZERO THE EXPONENT IS CLEARED AND THE
                      /SIGN WORD SET OFF,
                      /
02172 R 000000 A    %FNOR,  0
02173 R 140541 R            DZM     UNDFLN
```

```
02174 R 200527 R            LAC     OPCODE
02175 R 504656 R            AND     (UN!FL)
02176 R 544604 R            SAD     (CAL)
02177 R 622172 R            JMP*    %FNOR,
02200 R 504657 R            AND (UN)
02201 R 544657 R            SAD (UN)
02202 R 622172 R            JMP* %FNOR,
02203 R 203705 R            LAC     %FAC2   /IF %FAC2 = %FAC3
02204 R 504660 R            AND     (377777 /STRIP SIGN IN CASE OF
02205 R 043705 R            DAC     %FAC2   /MINUS ZERO,
02206 R 543706 R            SAD     %FAC3   /AND AC=0 CLEAR
02207 R 740200 A            SZA             /THE EXPONENT AND SIGN THE EXIT
02210 R 602223 R            JMP %FNORA-1    /WITH THE FLT ACC IN ABSOLUTE 0 STATE,
02211 R 103652 R            JMS OCHECK
02212 R 602217 R            JMP ,+5
02213 R 203707 R            LAC %FAC4
02214 R 544565 R            SAD %FAC4A
02215 R 740200 A            SZA
02216 R 602223 R            JMP %FNORA-1
02217 R 143704 R            DZM     %FAC1   /CLEAR EXPONT
02220 R 143721 R            DZM     %FSIGN  /CLEAR SIGN
02221 R 143717 R            DZM     %J4
02222 R 622172 R            JMP*    %FNOR,  /EXIT
02223 R 203705 R            LAC %FAC2
02224 R 744010 A    %FNORA  RCL             /IF BIT 0 IS ON EIT
02225 R 741100 A            SPA             /IF NOT KEEP ROTATING
02226 R 602267 R            JMP %FNORB
                    /
                    /DECREMENT EXPONENT,
                    /
02227 R 777777 A            LAW     -1
02230 R 343704 R            TAD     %FAC1
02231 R 043704 R            DAC     %FAC1
02232 R 200527 R            LAC OPCODE
02233 R 504610 R            AND (DP)
02234 R 544610 R            SAD (DP)
02235 R 602243 R            JMP ,+6
02236 R 203704 R            LAC %FAC1
02237 R 544702 R            SAD (777377)
02240 R 602245 R            JMP ,+5
02241 R 544703 R            SAD (377)       /RANDEX SAD
02242 R 602245 R            JMP ,+3
02243 R 203704 R            LAC %FAC1
02244 R 544660 R            SAD     (377777)
02245 R 440541 R            ISZ     UNDFLN  /EXPONENT UNDERFLOW
                    /
                    /ROTATE
                    /
02246 R 744000 A            CLL
02247 R 103652 R            JMS OCHECK
02250 R 602260 R            JMP %FN,
02251 R 204565 R            LAC %FAC4A
02252 R 744010 A            RCL
```

```
02253 R 044565 R                DAC %FAC4A
02254 R 203707 R                LAC %FAC4
02255 R 740010 A                RAL
02256 R 043707 R                DAC %FAC4       /USED AS FMQ
02257 R 044557 R                DAC GUARD
02260 R 203706 R    %FN,        LAC     %FAC3
02261 R 740010 A                RAL
02262 R 043706 R                DAC     %FAC3
02263 R 203705 R                LAC     %FAC2
02264 R 740010 A                RAL
02265 R 043705 R                DAC     %FAC2
02266 R 602224 R                JMP     %FNORA
02267 R 200536 R    %FNORB      LAC OVRFL1
02270 R 741200 A                SNA
02271 R 622172 R                JMP* %FNOR,
02272 R 200541 R                LAC UNDFLN
02273 R 741200 A                SNA
02274 R 622172 R                JMP* %FNOR,
02275 R 140536 R                DZM OVRFL1
02276 R 140541 R                DZM UNDFLN
02277 R 622172 R                JMP* %FNOR,
                        /
                        /-------------------
                        /
                        /HOLD THE FLOATING ACCUMULATOR (%FH,)
                        /CALLING SEQUENCE:
                        /       JMS     %FH             /CALL (VALUE IN FLOATING ACC)
                        /       NEXT INSTRUCTION                /RETURN (VALUE IN FLOATING ACC AND HELD AC
                        /
                        /
                        /
02300 R 000000 A    %FH,    0
02301 R 203704 R            LAC     %FAC1   /EXPONENT
02302 R 043710 R            DAC     %HAC1
02303 R 203705 R            LAC     %FAC2   /HIGH-ORDER MANTISSA
02304 R 043711 R            DAC     %HAC2
02305 R 203706 R            LAC     %FAC3   /LOW-ORDER MANTISSA
02306 R 043712 R            DAC     %HAC3
02307 R 622300 R            JMP*    %FH,    /EXIT
                        /
                        /--------------------
                        /
                        /ROUND AND INSERT SIGN (%FIR,)
                        /CALLING SEQUENCE:
                        /       JMS     %FIR,           /SUBR CALL
                        /               400/1           /ROUNDOFF BIT
                        /               777000/777776   /EXTRACT MASK
                        /       NEXT INSTRUCTION        /SUBR RETURN
                        /
                        /
                        /THE RESULT IN THE FLOATING ACCUMULATION IS ROUNDED ACCORDING
                        /TO THE MODE (SINGLE/DOUBLE PRECISION)
                        /
```

```
                        /
02310 R 000000 A    %FIR,   0
02311 R 140540 R            DZM     OVRFLR
02312 R 200527 R            LAC     OPCODE
02313 R 504670 R            AND     (UR!FL)
02314 R 544604 R            SAD     (CAL)
02315 R 602331 R            JMP     %FIREX
02316 R 504620 R            AND (UR)
02317 R 544620 R            SAD (UR)
02320 R 602331 R            JMP %FIREX
                        /+1 TO FMA IF %FAC4 IS NEGATIVE
02321 R 204557 R            LAC GUARD#
02322 R 745102 A            SPA!STL
02323 R 602334 R            JMP %FIR,A      /GUARD BIT SET FOR DP
02324 R 744000 A            CLL
02325 R 200527 R            LAC OPCODE
02326 R 504610 R            AND (DP)
02327 R 544604 R            SAD (CAL)
02330 R 602334 R            JMP %FIR,A
02331 R 442310 R    %FIREX  ISZ     %FIR,
02332 R 442310 R            ISZ     %FIR,
02333 R 622310 R            JMP*    %FIR,
02334 R 750010 A    %FIR,A  GLK
02335 R 362310 R            TAD*    %FIR,   /GET ROUNDOFF BIT,
02336 R 442310 R            ISZ     %FIR,   /POINT TO MASK,
02337 R 343706 R            TAD     %FAC3   /ADD TO LOW-ORDER RESULT TO PERFORM ROUNDOFF,
02340 R 522310 R            AND*    %FIR,   /MASK OF INSIGNIFICANT PORTION,
02341 R 043706 R            DAC     %FAC3   /KEEP LOW-ORDER RESULT,
02342 R 750010 A            GLK             /GET OVERFLOW BIT,
02343 R 343705 R            TAD     %FAC2   /ADD TO HIGH-ORDER RESULT,
02344 R 043705 R            DAC %FAC2
02345 R 740100 A            SMA             /OVER FLOW INTO SIGN BIT?
02346 R 602356 R            JMP     %FIRA   /NO
                        /
                        /ROUND-OFF CAUSED NUMBER TO OVERFLOW INTO SIGN BIT,
                        /ROTATE %FAC2,%FAC3 RIGHT ONE BIT AND INCREMENT EXPONENT,
                        /
02347 R 744020 A            RCR
02350 R 043705 R            DAC     %FAC2
02351 R 203706 R            LAC     %FAC3
02352 R 740020 A            RAR
02353 R 522310 R            AND*    %FIR,   /TRUNCATE
02354 R 043706 R            DAC     %FAC3
02355 R 103664 R            JMS INCEXP
02356 R 442310 R    %FIRA   ISZ     %FIR,   /POINT TO RETURN ADDRESS
02357 R 622310 R            JMP*    %FIR,   /EXIT
                            ,EJECT
```

```
                        /SIGN CONTROL (STRIP SIGNS) (%FS,)
                        /CALLING SEQUENCE:
                        /       JMS     %FS,    /CALL
                        /       NEXT INSTRUCTION        /RETURN
                        /
                        /THE SIGN BIT OF THE FLOATING ACC IS STORED IN BIT 0 OF
                        /STORAGE WORD %J4, THE SIGN TBIT OF THE HELD ACC IS
                        /EXCLUSIVE OR'ED WITH IT AND THAT VALUE IS STORED IN BIT 0 OF
                        /%FSIGN, RETURN IS MADE WITH BIT 0 OF BOTH THE
                        /HELD ACC AND THE FLOATING ACC SET TO ZERO,
                        /
                        /
02360 R 000000 A    %FS,    0
02361 R 203705 R            LAC     %FAC2   /SIGN WORD OF FLOATING ACCUMULATOR
02362 R 504626 R            AND     (400000 /KEEP ONLY THE SIGN
02363 R 043717 R            DAC     %J4     /STORE IT
02364 R 203705 R            LAC     %FAC2   /GET SIGN
02365 R 504660 R            AND     (377777 /WORD AGAIN AND KEEP
02366 R 043705 R            DAC     %FAC2   /ONLY THE ABSOLUTE VALUE
02367 R 203711 R            LAC     %HAC2   /SIGN WORD OF HELD ACCUMULATOR
02370 R 243717 R            XOR     %J4     /LIKE SIGNS=0, UNLIKE SIGN=1
02371 R 504626 R            AND     (400000 /KEEP ONLY THE SIGN AND
02372 R 043721 R            DAC     %FSIGN  /SAVE IT
02373 R 203711 R            LAC     %HAC2   /GET THE SIGN WORD,
02374 R 504660 R            AND     (377777 /OF THE HEALD ACC AND,
02375 R 043711 R            DAC     %HAC2   /STRIP OFF THE SIGN,
02376 R 622360 R            JMP*    %FS,    /EXIT
                        /CHECK THE FLOATING ACS TO EQUAL ZERO
02377 R 000000 A    CAC0    0
02400 R 203705 R            LAC %FAC2
02401 R 543706 R            SAD %FAC3
02402 R 740200 A            SZA
02403 R 741000 A            SKP
02404 R 602414 R            JMP CAC0EX+2
02405 R 203711 R            LAC %HAC2
02406 R 543712 R            SAD %HAC3
02407 R 740200 A            SZA
02410 R 602412 R            JMP CAC0EX
02411 R 602414 R            JMP .+3
02412 R 442377 R    CAC0EX  ISZ CAC0
02413 R 622377 R            JMP* CAC0
02414 R 143704 R            DZM %FAC1
02415 R 143705 R            DZM %FAC2
02416 R 143706 R            DZM %FAC3
02417 R 143707 R            DZM %FAC4
02420 R 144557 R            DZM GUARD
02421 R 144565 R            DZM %FAC4A
02422 R 622377 R            JMP* CAC0
                        /--------------------
                        /
                        /SWITCH THE FLOATING POINT ACCUMULATOR
                        /AND THE HELD ACCUMULATOR,
                        /CALLING SEQUENCE:
```

```
                        /       JMS     %SWICH  /SUBR CALL
                        /       NEXT INSTRUCTION        /RETURN
                        /
                        /THE ACCUMULATOR AND LINK ARE CLOBBERED BY
                        /THIS SUBROUTINE, TEMPORARY STORAGE WORD %J5 IS ALSO
                        /CLOBBERED.
                        /
02423 R 000000 A        %SWICH  0
02424 R 203704 R                LAC     %FAC1   /EXPONENTS,
02425 R 042360 R                DAC     %FS,    /USE TEMPORARILY FREE
02426 R 203710 R                LAC     %HAC1   /STORAGE FOR
02427 R 043704 R                DAC     %FAC1
02430 R 202360 R                LAC     %FS,    /WORD MANIPULATION
02431 R 043710 R                DAC     %HAC1
                        /                       /HIGH-ORDER MANTISSAE,
02432 R 203705 R                LAC     %FAC2
02433 R 042360 R                DAC     %FS,
02434 R 203711 R                LAC     %HAC2
02435 R 043705 R                DAC     %FAC2
02436 R 202360 R                LAC     %FS,
02437 R 043711 R                DAC     %HAC2
                        /                       /LOW-ORDER MANTISSAE
02440 R 203706 R                LAC     %FAC3
02441 R 042360 R                DAC     %FS,
02442 R 203712 R                LAC     %HAC3
02443 R 043706 R                DAC     %FAC3
02444 R 202360 R                LAC     %FS,
02445 R 043712 R                DAC     %HAC3
02446 R 622423 R                JMP*    %SWICH
                        /
                        /--------------------
                        /
                        /GENERAL FLOATING ADD
                        /CALLING SEQUENCE;
                        /       JMS     %FA,    /CALL(AUGEND INFLOAT ACC, ADDEN IN HELD ACC)
                        /       32/42           /MAXIMUM        SHIFT (26 S.P., 34 D.P.)
                        /       NEXT INSTRUCTION        /SUBR RETURN
                        /
                        /
                        /
02447 R 000000 A        %FA,    0
02450 R 143713 R                DZM %HAC4
02451 R 143707 R                DZM %FAC4
02452 R 144571 R                DZM %MQ#
02453 R 144557 R                DZM GUARD
02454 R 144564 R                DZM SWITCH#
02455 R 203705 R                LAC %FAC2
02456 R 504626 R                AND (400000)
02457 R 044570 R                DAC %J4%#       /SIGN OF FMB
02460 R 203711 R                LAC %HAC2
02461 R 504626 R                AND (400000)
02462 R 044566 R                DAC %F%#        /SIGN OF FMA - THE COMPLIMENT OF ORIGINAL XFAC IF FUNCTION IS NOT ADD
02463 R 102360 R                JMS %FS,        /STRIP SIGNS AND GET ABSOLUTE
```

```
                        /IF FMB=0 - EXIT - AND KEEP ORIGINAL FMA,,,+1 TO SWITCH
                        /IF FMA=0 - EXIT AND FMB TO FMA'S
02464 R 203705 R                LAC %FAC2       /FMB (HIGH)
02465 R 543706 R                SAD %FAC3       /FMB (LOW)
02466 R 740200 A                SZA
02467 R 602500 R                JMP ,+11
02470 R 203710 R                LAC %HAC1
02471 R 043704 R                DAC %FAC1
02472 R 203711 R                LAC %HAC2       /FMA
02473 R 043705 R                DAC %FAC2       /TO FMA (HIGH)
02474 R 203712 R                LAC %HAC3       /FMA
02475 R 043706 R                DAC %FAC3       /TO NEW FMA (LOW)
02476 R 444564 R                ISZ SWITCH
02477 R 603017 R                JMP %FAEX       /FMB = 0
                        /FMB NOT = 0
02500 R 203711 R                LAC %HAC2       /FMA
02501 R 543712 R                SAD %HAC3
02502 R 740200 A                SZA
02503 R 741000 A                SKP
02504 R 603017 R                JMP %FAEX       /FMA = 0
02505 R 203704 R    %FA,T       LAC     %FAC1   /DETERMINE EXPONENT DIFFERENCE
02506 R 740001 A                CMA             /(ADDEND-AUGEND-1)
02507 R 343710 R                TAD     %HAC1
02510 R 043720 R                DAC     %J5
02511 R 741100 A                SPA             /IF MINUS DO NOT SWITCH %FAC1 GREATER THAN %HAC1 (FMB > FMA)
02512 R 602516 R                JMP %FAT,B      /,,,DO NOT SWITCH
02513 R 102423 R                JMS     %SWICH
02514 R 444564 R                ISZ SWITCH      /HARDWARE FMA > FMB BEFORE ALIGNMENT
02515 R 602505 R                JMP     %FA,T   /ALSO
02516 R 203704 R    %FAT,B      LAC %FAC1
02517 R 243710 R                XOR %HAC1
02520 R 504626 R                AND (400000)
02521 R 741200 A                SNA
02522 R 602541 R                JMP %FAA        /EXPONENT SIGNS SIMILAR
                        /EXPONENT SIGNS UNLIKE
02523 R 203704 R                LAC %FAC1
02524 R 741100 A                SPA
02525 R 602532 R                JMP ,+5
02526 R 203710 R                LAC %HAC1
02527 R 740001 A                CMA
02530 R 343704 R                TAD %FAC1
02531 R 602534 R                JMP ,+3
02532 R 740001 A                CMA
02533 R 343710 R                TAD %HAC1
02534 R 741101 A                SPA!CMA
02535 R 603000 R                JMP %FAE,X      /OUT OF BOUNDS
02536 R 344704 R                TAD (43)
02537 R 741100 A                SPA
02540 R 603000 R                JMP %FAE,X      /OUT OF BOUNDS
02541 R 443720 R    %FAA        ISZ %J5
02542 R 741000 A                SKP
02543 R 602600 R                JMP %FAB,B      /EPA=EPB
02544 R 203720 R                LAC %J5
```

```
02545 R 362447 R            TAD* %FA,
02546 R 751100 A            SPA!CLA
02547 R 603000 R            JMP %FAE,X      /EXPONENTS GREATER THAN MAX ALLOWED 32/42
                      /
                      /SHI%FT THE ADDEND RIGHT DELTA TIMES,
                      /
02550 R 203711 R    %FAB    LAC     %HAC2
02551 R 744020 A            RCR
02552 R 043711 R            DAC     %HAC2
02553 R 203712 R            LAC     %HAC3
02554 R 740020 A            RAR
02555 R 043712 R            DAC     %HAC3
02556 R 203713 R            LAC %HAC4
02557 R 740020 A            RAR
02560 R 043713 R            DAC %HAC4
02561 R 443720 R            ISZ     %J5     /FINISHED SHIFTING?
02562 R 602550 R            JMP     %FAB    /NO,,,CYCLE AGAIN
                      /IF ROUNDING IS REQUESTED - AND BIT 0 OF %HAC4 IS SET = ROUND
                      /
02563 R 203713 R            LAC %HAC4
02564 R 744100 A            SMA!CLL
02565 R 602600 R            JMP %FAB,B      /DO NOT ROUND
02566 R 200527 R            LAC OPCODE
02567 R 504620 R            AND (UR)
02570 R 544620 R            SAD (UR)
02571 R 602600 R            JMP %FAB,B      /DO NOT ROUND
02572 R 203712 R    %RL     LAC %HAC3
02573 R 344643 R            TAD (1)
02574 R 043712 R            DAC %HAC3
02575 R 750010 A            GLK
02576 R 343711 R            TAD %HAC2
02577 R 043711 R            DAC %HAC2
                      /SWITCH ONLY ACCUMULATORS NOT EXP TO GET %HAC2 GREATER
02600 R 203713 R    %FAB,B  LAC %HAC4
02601 R 044571 R            DAC %MQ         /SAVE FOR DEBUGGING
02602 R 143713 R            DZM %HAC4
02603 R 143707 R            DZM %FAC4
02604 R 204564 R            LAC SWITCH
02605 R 144564 R            DZM SWITCH      /START A NEW SWITCH MONITOR
02606 R 741200 A            SNA
02607 R 602624 R            JMP %FABB1
02610 R 203711 R            LAC %HAC2
02611 R 543705 R            SAD %FAC2
02612 R 741000 A            SKP
02613 R 602620 R            JMP ,+5
02614 R 203712 R            LAC %HAC3
02615 R 744001 A            CMA!CLL
02616 R 343706 R            TAD %FAC3
02617 R 602644 R            JMP %FABB3
02620 R 203711 R            LAC %HAC2       /FMB
02621 R 740001 A            CMA
02622 R 343705 R            TAD %FAC2       /FMA
02623 R 602637 R            JMP %FABB2
```

```
02624 R 203705 R    %FABB1  LAC %FAC2
02625 R 543711 R            SAD %HAC2
02626 R 741000 A            SKP
02627 R 602634 R            JMP .+5
02630 R 203706 R            LAC %FAC3       /FMB
02631 R 744001 A            CMA!CLL
02632 R 343712 R            TAD %HAC3       /FMA
02633 R 602644 R            JMP %FABB3
02634 R 203705 R            LAC %FAC2       /FMB
02635 R 740001 A            CMA
02636 R 343711 R            TAD %HAC2       /FMA
                    /SWITCH ONLY %FAC2, AND 3 WITH CORRESPONDING %HAC2, AND 3
02637 R 740100 A    %FABB2  SMA
02640 R 602647 R            JMP %FAB,C-1
02641 R 204705 R            LAC (%FAB,C)
02642 R 042423 R            DAC %SWICH      /FMA<FMB
02643 R 602432 R            JMP %SWICH+7
02644 R 741400 A    %FABB3  SZL
02645 R 602647 R            JMP %FAB,C-1
02646 R 602641 R            JMP %FABB2+2
                    /NOW SHIFT ADDEND RIGHT ONE MORE TIME
02647 R 444564 R            ISZ SWITCH      /FMB<FMA
02650 R 203711 R    %FAB,C  LAC %HAC2
02651 R 744020 A            RCR
02652 R 043711 R            DAC %HAC2
02653 R 203712 R            LAC %HAC3
02654 R 740020 A            RAR
02655 R 043712 R            DAC %HAC3
02656 R 203713 R            LAC %HAC4
02657 R 740020 A            RAR
02660 R 043713 R            DAC %HAC4
                    /THE EXPONENTS ARE NOW EQUAL, IF THE SIGNS ARE UNLIKE
                    /NEGATE THE ADDEND BEFORE ADDING, AFTER ADDING IF THE
                    /SUM IS NEGATIVE THE ADDEND MUST BE LARGER, THE SIGN OF
                    /THE SUM WOULD THEN BE THAT OF THE ADDEND
02661 R 200527 R            LAC OPCODE
02662 R 504575 R            AND (OP6TO9)
02663 R 544576 R            SAD (ADD)
02664 R 602774 R            JMP %FANG1
                    /FSIGN = 400000 /LIKE SIGNS
                    /FSIGN = 000000 /UNLIKE SIGNS
02665 R 203721 R            LAC     %FSIGN
02666 R 740100 A            SMA             /NEGATE ADDEND
02667 R 602705 R            JMP     %FAC    /SUBTRACT UNLIKE ,,, REALLY ADD
02670 R 203713 R    %PFAC   LAC %HAC4       /SUBTRACT LIKE
02671 R 102013 R            JMS %ANEG,
02672 R 043713 R            DAC %HAC4
02673 R 203712 R            LAC     %HAC3
02674 R 740401 A            CMA!SNL
02675 R 602700 R            JMP .+3
02676 R 744000 A            CLL
02677 R 344643 R            TAD (1)
02700 R 043712 R            DAC     %HAC3   /NEGATED LOW-ORDER ADDEN
```

```
02701 R 203711 R            LAC     %HAC2   /HIGH-ORDER ADDEND
02702 R 741401 A            CMA!SZL         /IF OVER FLOW FROM
02703 R 344643 R            TAD     (1      /NEGATING LOW-ORDER ADDEND
02704 R 043711 R            DAC     %HAC2   /END OF NEGATION
                    /WHEN COMPUTING THE EXPONENT THE ONE'S COMPLEMENT WAS USED
                    /IOSTEAD OF THE TWO'S COMPLEMENT, THIS WAS TO FORCE A RIGHT SHIFT
                    /OF THE AUGEND SO THAT OVERFLOW WOULD NOT INTERFERE WITH SUM
                    /SIGN DETERMINATION
                    /
                    /
02705 R 203705 R    %FAC    LAC     %FAC2           /HIGH-ORDER AUGEND
02706 R 744020 A            RCR                     /1-BIT RIGHT
02707 R 043705 R            DAC     %FAC2
02710 R 203706 R            LAC     %FAC3
02711 R 740020 A            RAR                     /LOW-ORDER AUGEND
02712 R 043706 R            DAC %FAC3
02713 R 203707 R            LAC %FAC4
02714 R 740020 A            RAR
02715 R 744000 A            CLL                     /RIGHTMOST BIT OF %FAC4 DROPS OFF, (WHO CARES BABY)
                    /
                    /BEGIN THE ADDITION,
                    /
02716 R 343713 R            TAD %HAC4
02717 R 043707 R            DAC %FAC4
02720 R 750010 A            GLK
02721 R 343706 R            TAD %FAC3
02722 R 343712 R            TAD     %HAC3
02723 R 043706 R            DAC     %FAC3           /LOW-ORDER SUM
02724 R 750010 A            GLK                     /IF OVERFLOW FROM ADDITION
02725 R 343705 R            TAD     %FAC2
02726 R 343711 R            TAD     %HAC2
02727 R 043705 R            DAC     %FAC2           /HIGH-ORDER SUM
                    /
                    /IF SUM IS NEGATIVE, COMPLEMENT THE ANSWER AND THE SIGN OF THE SUM
02730 R 203705 R            LAC %FAC2
02731 R 740100 A            SMA
02732 R 602750 R            JMP     %FAD
02733 R 203707 R            LAC %FAC4
02734 R 102013 R            JMS %ANEG,
02735 R 043707 R            DAC %FAC4
02736 R 203706 R            LAC     %FAC3
02737 R 740401 A            CMA!SNL
02740 R 602743 R            JMP ,+3
02741 R 744000 A            CLL
02742 R 344643 R            TAD (1)
02743 R 043706 R            DAC     %FAC3
02744 R 203705 R            LAC     %FAC2
02745 R 741401 A            CMA!SZL
02746 R 344643 R            TAD     (1
02747 R 043705 R            DAC     %FAC2
                    /
                    /A FORCED SHIFT OF %FAB AND %FAC WAS PREVIOUSLY INITIATED
                    /SHIFT BACK IF BIT 1 OF %FAC2 IS = 0
```

```
                        /IF BIT1=1 INCREMENT EXPONENT BECAUSE OF THAT FORCED SHIFT
                        /AND SET GUARD BIT FOR ROUNDING
02750 R 203705 R    %FAD    LAC     %FAC2
02751 R 504706 R            AND (ADD15)
02752 R 741200 A            SNA
02753 R 602762 R            JMP     %FAD,D
02754 R 544707 R            SAD (JMS)
02755 R 602762 R            JMP %FAD,D
02756 R 203707 R            LAC %FAC4
02757 R 044557 R            DAC GUARD
02760 R 103664 R            JMS INCEXP
02761 R 603017 R            JMP     %FAEX   /DETERMINE ANSWER SIGN
                        /SHIFT BACK HERE
02762 R 203707 R    %FAD,D  LAC %FAC4
02763 R 744010 A            RCL
02764 R 043707 R            DAC %FAC4
02765 R 203706 R            LAC     %FAC3           /SHIFT BACK
02766 R 740010 A            RAL
02767 R 043706 R            DAC     %FAC3
02770 R 203705 R            LAC     %FAC2
02771 R 740010 A            RAL
02772 R 043705 R            DAC     %FAC2
02773 R 603017 R            JMP %FAEX
                        /IF FSIGN = 000000      /LIKE SIGNS
                        /IF FSIGN = 400000      /UNLIKE SIGNS
02774 R 203721 R    %FANG1  LAC %FSIGN
02775 R 741100 A            SPA
02776 R 602670 R            JMP %PFAC       /ADD UNLIKE - REALLY SUBTRACT
02777 R 602705 R            JMP %FAC        /ADD LIKE
                    /EXPONENT DIFFERENCE TO GREAT TO COMPLETE OPERATION
                    /EXPONENT REAVULATION
03000 R 203704 R    %FAE,X  LAC %FAC1
03001 R 243710 R            XOR %HAC1
03002 R 504626 R            AND (400000)
03003 R 741200 A            SNA
03004 R 603017 R            JMP %FAEX       /EXP SIGNS EQUAL
03005 R 203704 R            LAC %FAC1
03006 R 740100 A            SMA
03007 R 603012 R            JMP %FAEXX
03010 R 102423 R    %FE     JMS %SWICH      /SWITCH TO GET %FAC LARGER (MORE POSITIVE)
03011 R 444564 R            ISZ SWITCH
                    /FAC REGISTERS MORE +
                    /IF FAC2 AND FAC3 = 0, HAC REGISTERS MORE +
                    /SO THEREFORE SWITCH REGISTERS AND KEEP SIGN
03012 R 203705 R    %FAEXX  LAC %FAC2
03013 R 543706 R            SAD %FAC3
03014 R 740200 A            SZA
03015 R 741000 A            SKP
03016 R 603010 R            JMP %FE
03017 R 200527 R    %FAEX   LAC OPCODE
03020 R 504575 R            AND (OP6T09)
03021 R 544576 R            SAD (ADD)
03022 R 603042 R            JMP ADSIGN
```

```
03023 R 203721 R            LAC %FSIGN
03024 R 741200 A            SNA                 /SKP TO SUB LIKE SIGNS
03025 R 603035 R            JMP GOTOGO          /SUB UNLIKE SIGNS
                      /%FSIGN = 400000 TO SUBTRACT LIKE SIGNS
03026 R 204570 R            LAC %J4%
03027 R 043721 R            DAC %FSIGN          /BECOMES "A" SIGN
03030 R 204564 R            LAC SWITCH
03031 R 504643 R            AND (1)
03032 R 741200 A            SNA
                      /COMP ASIGN
03033 R 102017 R            JMS %SNEG,
03034 R 603056 R            JMP %FAEXT
03035 R 204570 R    GOTOGO  LAC %J4%
03036 R 750100 A            SMA!CLA
03037 R 204626 R            LAC (400000)
03040 R 043721 R            DAC %FSIGN          /BECOMES "A" SIGN
03041 R 603056 R            JMP %FAEXT
03042 R 203721 R    ADSIGN  LAC %FSIGN
03043 R 741200 A            SNA
03044 R 603054 R            JMP GOGOGO          /ADD LIKE SIGNS
                      /ADD UNLIKE SIGNS %FSIGN = 400000
03045 R 204566 R            LAC %F%             /SIGN OF FMA
03046 R 043721 R            DAC %FSIGN          /BECOMES "A" SIGN
03047 R 204564 R            LAC SWITCH
03050 R 504643 R            AND (1)
03051 R 741200 A            SNA
03052 R 102017 R            JMS %SNEG,
03053 R 603056 R            JMP %FAEXT
03054 R 204570 R    GOGOGO  LAC %J4%
03055 R 043721 R            DAC %FSIGN
03056 R 102172 R    %FAEXT  JMS     %FNOR,
03057 R 442447 R            ISZ     %FA,    /BUMP FOR EXIT
03060 R 622447 R            JMP*    %FA,            /EXIT
                      /
                      /WHEN SWITCH = 0, HARDWARE FMB > FMA ,,, THEREFORE COMP ASIGN
                            ,EJECT
```

```
                        /--------------------
                        /
                        /GENERAL FLOATING MULTIPLY (%FM,) NON-EAE
                        /CALLING SEQUENCE:
                        /       JMS     %FM,            /CALL
                        /       NEXT INSTRUCTION                /RETURN
                        /
                        /
                        /THE ARGUMENTS ARE IN THE FLOATING (%FAC1-3) AND
                        /HELD ACCUMULATORS (%HAC1-3), THE FLOATING ACCUMULATOR IS
                        /THE MULTIPLICAND AND THE HELD ACCUMULATOR THE MULTIPLIER,
                        /IF EITHER ARE ZERO, RETURN IS MADE WITH 0 IN THE FLOATING
                        /ACC, THE EXPONENT IS COMPUTED FIRST,
                        /
03061 R 000000 A    %FM,    0                       /ENTRY EXIT
03062 R 744000 A            CLL
03063 R 143707 R            DZM %FAC4
03064 R 144557 R            DZM GUARD
03065 R 203704 R            LAC     %FAC1           /EXPONENT OF MULTIPLICAND
03066 R 343710 R            TAD     %HAC1           /ADD EXPONENT OF MULTIPLIER
03067 R 043704 R            DAC     %FAC1           /SAVE EXPONENT OF PRODUCT,
03070 R 200527 R            LAC OPCODE
03071 R 504610 R            AND (DP)
03072 R 544610 R            SAD (DP)
03073 R 603112 R            JMP %FM2
03074 R 203704 R            LAC %FAC1
03075 R 504655 R            AND (777)
03076 R 746010 A            CLL!RTL
03077 R 742010 A            RTL
03100 R 742010 A            RTL
03101 R 742010 A            RTL
03102 R 740010 A            RAL
03103 R 040530 R            DAC TEMP
03104 R 200513 R            LAC %XFAC1
03105 R 240516 R            XOR %XHAC1
03106 R 504577 R            AND (400)
03107 R 740200 A            SZA
03110 R 603132 R            JMP %FM1        /UNLIKE SIGNS
03111 R 603120 R            JMP %FM2A
03112 R 203704 R    %FM2    LAC %FAC1
03113 R 040530 R            DAC TEMP
03114 R 200513 R            LAC %XFAC1
03115 R 240516 R            XOR %XHAC1
03116 R 741100 A            SPA
03117 R 603132 R            JMP %FM1        /UNLIKE SIGNS
03120 R 203710 R    %FM2A   LAC %HAC1
03121 R 741100 A            SPA
03122 R 603127 R            JMP %FM3        /LIKE SIGNS WERE NEGATIVE
03123 R 200530 R            LAC TEMP
03124 R 741100 A            SPA
03125 R 440536 R            ISZ OVRFL1
03126 R 603132 R            JMP .+4
03127 R 200530 R    %FM3    LAC TEMP
```

```
03130 R 740100 A            SMA
03131 R 440537 R            ISZ UNDFL1
                    /(+EPA)+(+EPA) STAYED POSITIVE
03132 R 102360 R    %FM1    JMS     %FS,            /GET SIGN AND ABSOLUTIZE,
03133 R 102377 R            JMS CAC0
03134 R 623061 R            JMP* %FM,
                    /
                    /CHECK MULTIPICAND FOR ZEROAND SAVE IN WORK AREA,
                    /
03135 R 203706 R            LAC     %FAC3           /LOW-ORDER MULTIPLICAND
03136 R 143706 R            DZM     %FAC3           /CLEAR FOR SHIFTING IN PRODUCT
03137 R 043716 R            DAC     %J3             /SAVE IT,
03140 R 203705 R            LAC     %FAC2           /HIGH-ORDER MULTIPLICAND
03141 R 143705 R            DZM     %FAC2           /CLEAR PRODUCT AREA FOR SHIFTING,
03142 R 043715 R            DAC     %J2             /NOT 0,,, SAVE IT
03143 R 203715 R            LAC %J2
03144 R 543716 R            SAD %J3
03145 R 740200 A            SZA
03146 R 741000 A            SKP
03147 R 603225 R            JMP %FMC
03150 R 143714 R            DZM     %J1             /INITIALIZE CARRY EXTENSIONS
03151 R 143707 R            DZM     %FAC4           /
03152 R 144567 R            DZM %J1A#
03153 R 144565 R            DZM %FAC4A#
                    /
                    /THE MULTIPLICATION IS PERFORMED BY A SERIES OF ADDS
                    /THE MULTIPLICAND IS ADDED TO ITSELF AND SHIFTED FOR EVERY
                    /1-BIT OF THE MULTIPLIER, THE ADDITION STOPS WHEN THE
                    /2-WORD MULTIPLIER IS 0,
                    /
03154 R 203715 R    %FMA    LAC     %J2             /SHIF THREE
03155 R 744020 A            RCR                     /WORD MULTIPLICAND
03156 R 043715 R            DAC     %J2             /ONE
03157 R 203716 R            LAC     %J3             /BIT
03160 R 740020 A            RAR                     /RIGHT
03161 R 043716 R            DAC     %J3
03162 R 203714 R            LAC     %J1             /3RD WORD EXTENSION OF MULTIPLICAND
03163 R 740020 A            RAR
03164 R 043714 R            DAC     %J1
03165 R 204567 R            LAC %J1A
03166 R 740020 A            RAR
03167 R 044567 R            DAC %J1A        /4TH WORD EXTENSION
                    /
03170 R 203712 R            LAC     %HAC3           /SHIFT TWO WORD MULTIPLIER
03171 R 744010 A            RCL                     /ONE BIT LEFT
03172 R 043712 R            DAC     %HAC3           /IF HIGH-ORDER WORD IS ZERO,
03173 R 203711 R            LAC     %HAC2           /THE MULTIPLIER MIGHT BE ZERO
03174 R 740010 A            RAL                     /
03175 R 043711 R            DAC     %HAC2           /
03176 R 741200 A            SNA                     /IF MULTIPLIER 0?
03177 R 603222 R            JMP     %FMB            /MAYBE
03200 R 744100 A            SMA!CLL                 /NO,,,IS MULTIPLIER BIT 0 SET ?
03201 R 603154 R            JMP     %FMA            /NO,,,CYCLE AGAIN
```

```
                        /
                        /ADD THE THREE WORD (INCLUDING EXTENSION) MULTIPLICAND TO THE
                        /BUILDING PRODUCT IN THE FLOATING ACC,
                        /
03202 R 204565 R                LAC %FAC4A
03203 R 344567 R                TAD %J1A
03204 R 044565 R                DAC %FAC4A
03205 R 750010 A                GLK
03206 R 343707 R                TAD     %FAC4           /YES,,,ADD MULTIPLICAND TO PRODUCT
03207 R 343714 R                TAD     %J1             /EXTENSION OF MULTIPLICAND (3RD WORD)
03210 R 043707 R                DAC     %FAC4           /3RD WORD OF BUILDING PRODUCT
03211 R 750010 A                GLK                     /ADD OVERFLOW FROM LINK TO 2ND WORD
03212 R 343706 R                TAD     %FAC3           /2ND WORD OF BUILDING PRODUCT
03213 R 343716 R                TAD     %J3
03214 R 043706 R                DAC     %FAC3
03215 R 750010 A                GLK                     /ADD OVERFLOW FROM LINK TO FST WORD
03216 R 343715 R                TAD     %J2             /OF PRODUCT
03217 R 343705 R                TAD     %FAC2
03220 R 043705 R                DAC     %FAC2
03221 R 603154 R                JMP     %FMA            /CYCLE AGAIN
                        /AT % FMB THE HIGH-ORDER MULTIPLIER WORD IS ZERO,
                        /TEST LOW-ORDER WORD, IF IT IS ZERO ALSO, EXIT,
                        /
03222 R 203712 R    %FMB    LAC     %HAC3           /IS MULTIPLIER Ø ?
03223 R 740200 A                SZA                     /YES
03224 R 603154 R                JMP     %FMA            /NO,,,CYCLE AGAIN
                        /
                        /EXIT ROUTINE,
03225 R 203707 R    %FMC    LAC %FAC4               /REDUNDANT IN CASE UNNORMALIZED
03226 R 044557 R            DAC GUARD
03227 R 102172 R            JMS     %FNOR,          /NORMALIZE
03230 R 623061 R            JMP*    %FM,            /RETURN
                            ,EJECT
```

```
                        /GENERAL FLOATING DIVIDE (%FD,) NON-EAE
                        /CALLING SEQUENCE;
                        /
                        /       JMS     %FD,            /CALL
                        /       -34/-44                 /NO OF BITS TO GENERATE
                        /       400/1                   /LEAST SIGNIFICANT QUOTIENT BIT
                        /       NEXT INSTRUCTION                /RETURN HERE
                        /
                        /UPON ENTRY TO THE SUBROUTINE THE ARGUMENTS WERE STORED IN THE
                        /FLOATING POINT REGISTER (%FAC1-3), AND THE HELD ACCUMULATOR (%HAC1-3),
                        /RETURN IS MADE TO THE CALLING PROGRAM WITH THE QUOTIENT IN THE
                        /FLOATING POINT REGISTER,  THE DIVISOR IS IN THE FLOATION AC,
                        /
03231 R 000000 A        %FD,    0                       /ENTRY-EXIT
03232 R 223231 R                LAC*    %FD,            /GET NO OF BITS (MUST BE NEGATIVE)
03233 R 043714 R                DAC     %J1             /SAVE NO OF BITS AS COUNTER,
03234 R 443231 R                ISZ     %FD,
03235 R 102660 R                JMS     %FS,            /SIGN CONTROL
                        /ABNORMAL DIVISION IF FMB = 0
03236 R 203705 R                LAC %FAC2               /FMB HIGH
03237 R 543706 R                SAD %FAC3               /FMB LOW
03240 R 740200 A                SZA
03241 R 741000 A                SKP
03242 R 603563 R                JMP %FDC                /ABNORMAL DIVISION
03243 R 200527 R                LAC OPCODE
03244 R 504603 R                AND (FL)
03245 R 544604 R                SAD (CAL)
03246 R 603405 R                JMP IDIVIDE             /INTEGER DIVISION
                        /ABNORMAL DIVISION IF FMB IS UNNORMALIZED
03247 R 200527 R                LAC OPCODE
03250 R 504575 R                AND (OP6TO9)
03251 R 544637 R                SAD (RDIV)
03252 R 603257 R                JMP .+5         /EXCEPT FOR REVERSE DIVISION
03253 R 200517 R                LAC %XHAC2      /FMB HIGH
03254 R 744010 A                RCL
03255 R 740100 A                SMA
03256 R 603275 R                JMP %FD0        /CALCULATE EXPONENT BEFORE FLAGGING ABNORMAL DIVISION
                        /NO ABNORMAL DIVISION IF FMA = 0
03257 R 203711 R                LAC %HAC2
03260 R 543712 R                SAD %HAC3
03261 R 740200 A                SZA
03262 R 741000 A                SKP
03263 R 603445 R                JMP IDEXIT              /THE QUOTENT WILL = 0
                        /NEGATE THE DIVISOR IN THE FLOATING ACCUMULATOR,
03264 R 203706 R                LAC     %FAC3           /LOW-ORDER DIVISION
03265 R 143706 R                DZM     %FAC3           /CLEAR FOR QUOTIENT
03266 R 102013 R                JMS     %ANEG,          /NEGATE AND
03267 R 043716 R                DAC     %J3             /SAVE LOW-ORDER DIVISION
03270 R 203705 R                LAC     %FAC2           /HIGH-ORDER DIVISION
03271 R 741401 A                CMA!SZL                 /SKIP IF LINK ON FROM NEGATING LOW-ORDER
03272 R 344643 R                TAD     (1              /DIVISOR
03273 R 143705 R                DZM     %FAC2           /CLEAR FOR QUOTIENT
03274 R 043715 R                DAC     %J2
```

```
                        /THE EXPONENT OF THE DIVIDEND MINUS THE EXPONENT OF THE
                        /DIVISOR EQUALS THE EXPONENT QUOTIENT,
03275 R 203704 R    %FD0    LAC     %FAC1           /DIVISOR EXPONENT
03276 R 102013 R            JMS     %ANEG,
03277 R 343710 R            TAD     %HAC1           /DIVIDEND EXPONENT
03300 R 043704 R            DAC     %FAC1           /EXPONENT BEFORE NORMALIZATION
03301 R 200527 R            LAC OPCODE
03302 R 504610 R            AND (DP)
03303 R 544610 R            SAD (DP)
03304 R 603323 R            JMP %FD2        /DOUBLE PRECISION
03305 R 203704 R            LAC %FAC1
03306 R 504655 R            AND (777)
03307 R 744010 A            CLL!RAL
03310 R 742010 A            RTL
03311 R 742010 A            RTL
03312 R 742010 A            RTL
03313 R 742010 A            RTL
03314 R 040530 R            DAC TEMP        /TEMP FOR QUOTENT EXPONENT
03315 R 200513 R            LAC %XFAC1      /EPA
03316 R 240516 R            XOR %XHAC1      /EPB
03317 R 504577 R            AND (400)
03320 R 741200 A            SNA
03321 R 603343 R            JMP %FD1        /EXPONENTS WERE; LIKE SIGNS
03322 R 603331 R            JMP %FD4
03323 R 203704 R    %FD2    LAC %FAC1       /,,,UNLIKE SIGNS
03324 R 040530 R            DAC TEMP        /TEMP FOR QUOTENT EXPONENT
03325 R 200513 R            LAC %XFAC1
03326 R 240516 R            XOR %XHAC1
03327 R 740100 A            SMA
03330 R 603343 R            JMP %FD1
                        /EXPONENTS WERE OF UNLIKE SIGNS
03331 R 203710 R    %FD4    LAC %HAC1       /EPA
03332 R 741100 A            SPA
03333 R 603340 R            JMP %FD3        /EPA WAS NEGATIVE
03334 R 200530 R            LAC TEMP
03335 R 741100 A            SPA
03336 R 440536 R            ISZ OVRFL1      /FROM SUBTRACTION OF EXPONENTS
03337 R 603343 R            JMP ,+4
03340 R 200530 R    %FD3    LAC TEMP
03341 R 740100 A            SMA
03342 R 440537 R            ISZ UNDFL1      /FROM SUBTRACTION OF EXPONENTS
                        /ABNORMAL DIVISION NOW IF FMB IS UNNORMALIZED
03343 R 200527 R    %FD1    LAC OPCODE
03344 R 504575 R            AND (OP6TO9)
03345 R 544637 R            SAD (RDIV)
03346 R 603353 R            JMP ,+5         /EXCEPT FOR REVERSE DIVISION
03347 R 200517 R            LAC %XHAC2      /FMB HIGH
03350 R 744010 A            RCL
03351 R 740100 A            SMA
03352 R 603563 R            JMP %FDC        /ABNORMAL DIVISION
                        /NORMALIZE MANTISSA BECAUSE HARDWARE NORMALIZES IT SONNY
                        /THE PROGRAM WILL NEVER EXPECT A FLOATING ABNORMAL DIVISION ON RD
03353 R 203711 R            LAC %HAC2       /FMA HIGH
```

```
03354 R 504661 R            AND (LAC)
03355 R 740200 A            SZA
03356 R 603527 R            JMP %FDA-2
03357 R 203712 R            LAC %HAC3       /FMA LOW
03360 R 744010 A            RCL
03361 R 043712 R            DAC %HAC3
03362 R 203711 R            LAC %HAC2
03363 R 740010 A            RAL
03364 R 043711 R            DAC %HAC2
03365 R 777777 A            LAW -1
03366 R 343704 R            TAD %FAC1       /QUOTENT
03367 R 043704 R            DAC %FAC1       /EXPONENT
03370 R 200527 R            LAC OPCODE
03371 R 504610 R            AND (DP)
03372 R 544610 R            SAD (DP)
03373 R 603401 R            JMP .+6
03374 R 203704 R            LAC %FAC1
03375 R 544702 R            SAD (777377)
03376 R 603403 R            JMP .+5
03377 R 544703 R            SAD (377)
03400 R 603403 R            JMP .+3
03401 R 203704 R            LAC %FAC1
03402 R 544660 R            SAD (377777)
03403 R 440541 R            ISZ UNDFLN
03404 R 603543 R            JMP %FD1
                    /INTEGER DIVISION
                    /IF DIVIDEND < DIVISOR, THE QUOTENT WILL = 0
                    /(FMA<FMB)
                    /AND IF FMB = 0, +1 TO INTEGER DIVISION BY ZERO
03405 R 203711 R    IDIVIDE LAC %HAC2       /FMA
03406 R 543705 R            SAD %FAC2       /FMB
03407 R 741000 A            SKP             /FMA = FMB HIGH CHECK LOW
03410 R 603440 R            JMP IDV2
03411 R 203712 R            LAC %HAC3       /FMA
03412 R 745100 A            SPA!CLL
03413 R 603422 R            JMP .+7
03414 R 740031 A            TCA
03415 R 343706 R            TAD %FAC3       /FMB
03416 R 741400 A            SZL
03417 R 741200 A            SNA
03420 R 603454 R            JMP IDIFR       /FMA > FMB
03421 R 603445 R            JMP IDEXIT      /FMA < FMB
03422 R 203706 R            LAC %FAC3       /FMB LOW
03423 R 745100 A            SPA!CLL
03424 R 603432 R            JMP .+6
03425 R 740031 A            TCA
03426 R 343712 R            TAD %HAC3       /FMA LOW
03427 R 740400 A            SNL
03430 R 603445 R            JMP IDEXIT      /FMA < FMB
03431 R 603454 R            JMP IDIFR       /FMA > FMB
03432 R 740031 A            TCA
03433 R 343712 R            TAD %HAC3       /FMA LOW
03434 R 741400 A            SZL
```

```
03435 R 741200 A            SNA
03436 R 603445 R            JMP IDEXIT      /FMA < FMB
03437 R 603454 R            JMP IDIFR       /FMA > FMB
03440 R 740001 A    IDV2    CMA
03441 R 343705 R            TAD %FAC2       /FMB HIGH
03442 R 740100 A            SMA
03443 R 603445 R            JMP IDEXIT      /FMA < FMB
03444 R 603454 R            JMP IDIFR       /FMA > FMB
03445 R 143704 R    IDEXIT  DZM %FAC1       /QUOTENT WILL = 0
03446 R 143705 R            DZM %FAC2
03447 R 143706 R            DZM %FAC3
03450 R 140536 R            DZM OVRFL1
03451 R 140537 R            DZM UNDFL1
03452 R 140541 R            DZM UNDFLN
03453 R 603600 R            JMP %FDEXIT
                    /DETERMINE THE DIFFERENCE TWX THE FMA/FMB AND DO THAT MANY SUBTRACTIONS
03454 R 143714 R    IDIFR   DZM %J1         /DIVISION COUNTER
03455 R 203711 R    IDVC    LAC %HAC2
03456 R 504661 R            AND (LAC)
03457 R 740200 A            SZA
03460 R 603476 R            JMP IDVC3
03461 R 203712 R            LAC %HAC3
03462 R 744010 A            RCL             /FMA LOW
03463 R 043712 R            DAC %HAC3
03464 R 203711 R            LAC %HAC2
03465 R 740010 A            RAL             /FMA HIGH
03466 R 043711 R            DAC %HAC2
03467 R 203706 R            LAC %FAC3
03470 R 744010 A            RCL             /FMB LOW
03471 R 043706 R            DAC %FAC3
03472 R 203705 R            LAC %FAC2
03473 R 740010 A            RAL             /FMB HIGH
03474 R 043705 R            DAC %FAC2
03475 R 603455 R            JMP IDVC
03476 R 443714 R    IDVC3   ISZ %J1         /+1 TO DIVISION COUNTER
03477 R 203705 R            LAC %FAC2
03500 R 504661 R            AND (LAC)
03501 R 740200 A            SZA
03502 R 603512 R            JMP IDVC4
03503 R 203706 R            LAC %FAC3
03504 R 744010 A            RCL
03505 R 043706 R            DAC %FAC3
03506 R 203705 R            LAC %FAC2
03507 R 740010 A            RAL
03510 R 043705 R            DAC %FAC2
03511 R 603476 R            JMP IDVC3       /FMB 00 NOT = 1
                    /FMB00 = 1
03512 R 203714 R    IDVC4   LAC %J1
03513 R 740031 A            TCA
03514 R 043714 R            DAC %J1         /NEGETED DIFFERENCE TWX FMA/FMB
                    /THE DIFFERENCE TWX THE FMA/FMB HAS BEED DETERMINED
                    /PRIME MANTISSAS FOR SUBTRACTIONS
03515 R 744000 A            CLL
```

```
03516 R 203706 R            LAC %FAC3
03517 R 740031 A            TCA
03520 R 043716 R            DAC %J3
03521 R 203705 R            LAC %FAC2
03522 R 741401 A            CMA!SZL
03523 R 344643 R            TAD (1)
03524 R 043715 R            DAC %J2
03525 R 143706 R            DZM %FAC3
03526 R 143705 R            DZM %FAC2
                    /THE DIVISION IS DONE BY A SERIES OF SUBTRACTIONS, THE DIVISOR WILL BE
                    /SUBTRACTED FROM THE DIVIDEND, IF THE RESULT IS MINUS THE DIVISOR IS
                    /LARGER THAN THE DIVIDED AND A BIT IS NOT ADDED TO THE SHIFTED EXPONENT
03527 R 777777 A            LAW -1
03530 R 044556 R            DAC FSTSUB#
03531 R 203706 R    %FDA    LAC     %FAC3           /SHIFT BUILDING QUOTIENT 1 BIT LEFT
03532 R 744010 A            RCL
03533 R 043706 R            DAC     %FAC3
03534 R 203705 R            LAC     %FAC2           /IF NEGATIVE, THE LARGEST POSSIBLE
03535 R 740010 A            RAL                     /QUOTIENT HAS BEEN OBTAINED,
03536 R 741100 A            SPA
03537 R 603616 R            JMP     %FDE
03540 R 043705 R            DAC     %FAC2           /SUBTRACT LOW DIVISOR FROM
03541 R 203716 R            LAC     %J3             /LOW DIVIDEND
03542 R 343712 R            TAD     %HAC3
03543 R 043710 R            DAC     %HAC1           /SAVE TEMPORARILY,,,MAYBE NEW LOW-ORDER DIVIDEND
03544 R 750010 A            GLK
03545 R 343711 R            TAD     %HAC2           /IF OVERFLOW ADD 1 TO HIGH ORDER DIVIDEND
03546 R 343715 R            TAD     %J2             /SUBJECT HIGH-ORDER DIVISOR
03547 R 740100 A            SMA
03550 R 603602 R            JMP     %FDD            /IF NEGATIVE DO NOT GENERATE A QUOTIENT BIT
03551 R 203712 R            LAC     %HAC3           /CONTINUE WITH OLD DIVIDEND INSTEAD OF REMAINDER
03552 R 744010 A    %FDB    RCL                     /
03553 R 043712 R            DAC     %HAC3           /
03554 R 203711 R            LAC     %HAC2           /SHIFT DIVIDEND
03555 R 740010 A            RAL                     /
03556 R 043711 R            DAC     %HAC2           /
03557 R 144556 R            DZM FSTSUB
03560 R 443714 R            ISZ     %J1             /HAVE ALL BITS BEEN SHIFTED
03561 R 603531 R            JMP     %FDA            /NO,,,CYCLE AGAIN
03562 R 603622 R            JMP %FDF                /INGEGER JUMP
                    /ABNORMAL DIVISION EXPECTED
03563 R 200527 R    %FDC    LAC OPCODE
03564 R 504603 R            AND (FL)
03565 R 544604 R            SAD (CAL)
03566 R 440544 R            ISZ IDVBY0              /INTEGER DIVISION BY ZERO
03567 R 544603 R            SAD (FL)
03570 R 440542 R            ISZ DVBY00              /FLOATING ABNORMAL DIVISION
03571 R 200527 R            LAC OPCODE
03572 R 504610 R            AND (DP)
03573 R 544604 R            SAD (CAL)
03574 R 603600 R            JMP .+4
03575 R 140536 R            DZM OVRFL1              /ABNORMAL DIVISION TAKES PRECIDENCE
03576 R 140537 R            DZM UNDFL1              /OVER DP JEA FUNCTIONS
```

```
03577 R 140541 R            DZM UNDFLN
03600 R 443231 R    %FDEXIT ISZ     %FD,            /BUMP FOR EXIT
03601 R 623231 R            JMP*    %FD,            /EXIT
                    /AT%FDD THE DIVISOR IS LESS THAN THE DIVIDEND AND A QUOTIENT BIT
                    /IS GENERATED, THE QUOTIENT BIT IS PICKED UP FROM THE CALLING SEQUENCE;
                    /IT IS THE LEAST SIGNIFICANT BIT AND IS ORED INTO THE BUILDING QUOTIENT.
                    /AN ADJUSTMENT IS NECESSARY TO PICK-UP THE NEW DIVIDEND (REMAINDER).
                    /
03602 R 043711 R    %FDD    DAC     %HAC2           /NEW-HIGH-ORDER DIVIDEND
03603 R 740400 A            SNL
03604 R 603610 R            JMP ,+4
03605 R 777777 A            LAW -1
03606 R 544556 R            SAD FSTSUB
03607 R 103664 R            JMS INCEXP
03610 R 144556 R            DZM FSTSUB
03611 R 223231 R            LAC*    %FD,            /GET LEAST SIGNIFICANT BIT, OR IT
03612 R 243706 R            XOR     %FAC3           /INTP THE LOW-ORDER QUOTIENT WORD
03613 R 043706 R            DAC     %FAC3           /SAVE NEW QUOTIENT
03614 R 203710 R            LAC     %HAC1           /GET LOW-ORDER REMAINDER
03615 R 603552 R            JMP     %FDB            /GO BACK AND SHIFT QUOTIENT
                    /
                    /AT %FDE THE LARGEST POSSIBLE QUOTIENT HAS BEEN OBTAINED, THE
                    /QUOTIENT WAS SHIFTED LEFT ONE BIT TOO MANY, SHIFT IT BACK 1 BIT
                    /AND INCREMENT THE EXPONENT OF THE QUOTIENT
                    /
03616 R 744020 A    %FDE    RCR                     /HIGH-ORDER QUOTIENT
03617 R 203706 R            LAC     %FAC3           /GET LOW-ORDER QUOTIENT AND
03620 R 740020 A            RAR                     /REPLACE BIT TAKEN FROM IT,
03621 R 043706 R            DAC     %FAC3           /MANTISSA  PORTION COMPLETE,
03622 R 200527 R    %FDF    LAC OPCODE
03623 R 504603 R            AND (FL)
03624 R 544604 R            SAD (CAL)
03625 R 143704 R            DZM %FAC1               /MUST EXIT WITH = 0 FOR I DIVISION
03626 R 203711 R            LAC %HAC2
03627 R 044557 R            DAC GUARD
03630 R 200536 R            LAC OVRFL1
03631 R 741200 A            SNA
03632 R 603600 R            JMP %FDEXIT
03633 R 200537 R            LAC UNDFL1
03634 R 340541 R            TAD UNDFLN
03635 R 741200 A            SNA
03636 R 603600 R            JMP %FDEXIT
03637 R 200527 R            LAC OPCODE
03640 R 504610 R            AND (DP)
03641 R 544610 R            SAD (DP)
03642 R 603646 R            JMP ,+4
03643 R 203704 R            LAC %FAC1
03644 R 544577 R            SAD (400)
03645 R 741000 A            SKP
03646 R 140536 R            DZM OVRFL1
03647 R 140537 R            DZM UNDFL1
03650 R 140541 R            DZM UNDFLN
03651 R 603600 R            JMP %FDEXIT             /EXIT ROUTINE WITHOUT NORMALIZING,
```

```
03652 R 000000 A    OCHECK  0
03653 R 200527 R            LAC OPCODE
03654 R 504575 R            AND (OP6TO9)
03655 R 544600 R            SAD (MUL)
03656 R 745000 A            SKP!CLL
03657 R 544601 R            SAD (DIV)
03660 R 745000 A            SKP!CLL
03661 R 544637 R            SAD (RDIV)
03662 R 443652 R            ISZ OCHECK
03663 R 623652 R            JMP* OCHECK
                        /INCREMENT EXPONENT ROUTINER
03664 R 003664 R    INCEXP  .DSA .
03665 R 443704 R            ISZ %FAC1
03666 R 740000 A            NOP
03667 R 200527 R            LAC OPCODE
03670 R 504610 R            AND (OP)
03671 R 544610 R            SAD (OP)
03672 R 603700 R            JMP .+6
03673 R 203704 R            LAC %FAC1
03674 R 544710 R            SAD (777400)
03675 R 440536 R            ISZ OVRFL1
03676 R 544577 R            SAD (400)
03677 R 440536 R            ISZ OVRFL1
03700 R 203704 R            LAC %FAC1
03701 R 544626 R            SAD (400000)
03702 R 440536 R            ISZ OVRFL1
03703 R 623664 R            JMP* INCEXP
03704 R 000000 A    %FAC1   0
03705 R 000000 A    %FAC2   0
03706 R 000000 A    %FAC3   0
03707 R 000000 A    %FAC4   0
03710 R 000000 A    %HAC1   0
03711 R 000000 A    %HAC2   0
03712 R 000000 A    %HAC3   0
03713 R 000000 A    %HAC4   0
03714 R 000000 A    %J1     0
03715 R 000000 A    %J2     0
03716 R 000000 A    %J3     0
03717 R 000000 A    %J4     0
03720 R 000000 A    %J5     0
03721 R 000000 A    %FSIGN  0       /SIGN OF FLOATING ACC
                        /
                        /
                        /
```

```
                        ,TITLE SUBTITLE DOUBLE
                    /
                    /
                    /
                        ,IFUND %NDOUB
                    /DOUBLE PRECISION FLOATING POINT PACKAGE
                    /
                    /
                    /
                    /
                    /       DLAC            LOAD
                    /       DDAC            STORE
                    /       DFAD            ADD
                    /       DSUB            SUBTRACT
                    /       DDVD            DIVIDE
                    /       DMPY            MULTIPLY
                    /       DDVR            REVERSE DIVIDE
                    /       DSBR            REVERSE SUBTRACT
                    /
                    /
                    /
                    /REQUIRED PROGRAMS: FPOINT= GENERAL FLOATING POINT ARITHMETIC
                    /
                    /
                    /--------------------
                    /
                    /--------------------
                    /
                    /DOUBLE PRECISION FLOATING POINT LOAD (DLAC)
                    /CALLING SEQUENCE:
                    /
                    /       JMS     DLAC    /SUBR CALL (XCT IF INDIRECT),
                    /       XCT     ADDR    /ADDRESS OF DOUBLE PRECISION
                    /       NEXT INSTRUCTION        /RETURN ARG IN FLT ACC
                    /
                    /
03722 R 000000 A    DLAC    0
03723 R 102151 R            JMS     %FG,            /GET ADDRESS OF ARGUMENT
03724 R 000000 A    %DP1    0                       /ARGUMENT ADDRESS
03725 R 223724 R            LAC*    %DP1            /FIRST WORD
03726 R 043704 R            DAC     %FAC1           /EXPONENT
03727 R 443724 R            ISZ     %DP1            /POINT TO SECOND WORD
03730 R 223724 R            LAC*    %DP1
03731 R 043705 R            DAC     %FAC2
03732 R 443724 R            ISZ     %DP1
03733 R 223724 R            LAC*    %DP1            /LOW-ORDER MANTISSA
03734 R 043706 R            DAC     %FAC3
03735 R 203722 R    DLAC1   LAC DLAC
03736 R 504701 R            AND (77777)
03737 R 544711 R            SAD (%DP9A)
03740 R 603745 R            JMP DLAC2
03741 R 544712 R            SAD (%DP6A)
03742 R 603745 R            JMP DLAC2
03743 R 102024 R            JMS %FLOT,
03744 R 623722 R            JMP* DLAC
```

```
03745 R 200527 R     DLAC2    LAC OPCODE
03746 R 504575 R              AND (OP6TO9)
03747 R 544602 R              SAD (RSUB)
03750 R 623722 R              JMP* DLAC
03751 R 544637 R              SAD (RDIV)
03752 R 623722 R              JMP* DLAC
03753 R 102024 R              JMS %FLOT,
03754 R 623722 R              JMP*    DLAC
                              ,EJECT
```

```
                        /DOUBLE PRECISION FLOATING STORE        (DDAC)
                        /CALLING SEQUENCE:
                        /
                        /       JMS     DDAC    /SUBR CALL(VALUE IN FLOATING ACC)
                        /       XCT     ADDR    /ADDR OF ARG (XCT IF INDIRECT)
                        /       NEXT INSTRUCTION        /SUBR RETURN (VALUE IN ARG AND FLOATING ACC)
                        /
03755 R 000000 A    DDAC    0
03756 R 102151 R            JMS     %FG,            /GET ADDRESS
03757 R 000000 A    %DP2    0
03760 R 203704 R            LAC     %FAC1           /EXPONENT
03761 R 063757 R            DAC*    %DP2
03762 R 443757 R            ISZ     %DP2
03763 R 203705 R            LAC     %FAC2           /2ND WORD
03764 R 063757 R            DAC*    %DP2
03765 R 443757 R            ISZ     %DP2            /3RD WORD
03766 R 203706 R            LAC     %FAC3
03767 R 063757 R            DAC*    %DP2
03770 R 623755 R            JMP*    DDAC
                        /DOUBLE PRECISION FLOATING POINT SUBTRACT (DSUB)
                        /CALLING SEQUENCE:
                        /
                        /       JMS     DSUB    /MINUEND IN FLOATING ACC
                        /       XCT     ADDR    /ADDR OF SUBTRAHEND (XCT IF INDIRECT)
                        /       NEXT INSTRUCTION        /SUBR RETURN (DIFFERENCE IN FLOATING AC)
                        /
03771 R 000000 A    DSUB    0
03772 R 102151 R            JMS     %FG,            /GET ADDRESS OF SUBTRAHEND
03773 R 000000 A    %DP3    0                       /STORED ADDRESS OF SUBTRAHEND
03774 R 102142 R            JMS     %FNEG,          /NEGATE MINUEND (SUBTRAHEND-MINUED)
03775 R 104055 R            JMS     DFAD            /DOUBLE PRECISION ADD
03776 R 403773 R            %DP3+400000             /INDIRECT ADDRESS (XCT) OF ARG
03777 R 623771 R            JMP*    DSUB            /EXIT
                        /
                        /DOUBLE PECISION FLOATING POINT MULTIPLY (DMPY)
                        /CALLING SEQUENCE:
                        /
                        /       JMS     DMPY    /SUBR CALL (MULTIPLICAND IN FLATING ACC)
                        /       XCT     ADDR    /ADDR OF MULTIPLIER (XCT IF INDIRECT)
                        /       NEXT INSTRUCTION        /SUBR RETURN (PRODUCT IN FLOATING ACC)
                        /
04000 R 000000 A    DMPY    0
04001 R 102151 R            JMS     %FG,            /GET ADDRESS OF MULTIPLIER AND
04002 R 000000 A    %DP4    0                       /STORE HERE
04003 R 102300 R            JMS     %FH,            /HOLD FLOATING ACC,
04004 R 103722 R            JMS     DLAC
04005 R 404002 R            %DP4+400000     /MULTIPLIER
04006 R 103061 R            JMS     %FM,            /GENERAL MULTIPLY
04007 R 102310 R            JMS     %FIR,           /ROUND AND SIGN
04010 R 000000 A                    0               /ROUND OFF BIT
04011 R 777777 A                    777777          /MASK
04012 R 624000 R            JMP*    DMPY            /EXIT
                            .EJECT
```

```
                        /--------------------
                        /
                        /DOUBLE PRECISION FLOATING POINT REVERSE DIVIDE (DDVR)
                        /CALLING SEQUENCE:
                        /
                        /       JMS     DDVR            /SUBR CALL (DIVISOR IN FLT ACC)
                        /       XCT     ADDR            /ADDRESS OF DIVIDEND
                        /       NEXT    INSTRUCTION     /RETURN (QUOTIENT) IN FLOATING ACCUMULATOR)
                        /
04013 R 000000 A    DDVR    0
04014 R 102151 R            JMS     %FG,            /GET ADDRESS OF DIVIDEND
04015 R 000000 A    %DP5    0
04016 R 103755 R            JMS     DDAC            /STORE DOUBLE
04017 R 003714 R            ,DSA    %J1
04020 R 103722 R            JMS     DLAC            /LOAD DOUBLE
04021 R 404015 R            %DP5+400000     /DIVIDEND
04022 R 104025 R            JMS     DDVD            /DIVIDE
04023 R 003714 R            ,DSA    %J1
04024 R 624013 R            JMP*    DDVR            /EXIT
                        /
                        /--------------------
                        /
                        /DOUBLE PRECISION FLOATING POINT DIVIDE (DDVD)
                        /CALLING SEQUENCE:
                        /
                        /       JMS     DDVD    /SUBR CALL (DIVIDEND IN FLOATING ACC)
                        /       XCT     ADDR    /XCT IF INDIRECT (ADDR OF DOUBLE PRECISION DIVISOR)
                        /       NEXT INSTRUCTION        /RETURN (QUOTIENT IN FLOATING ACC)
                        /
04025 R 000000 A    DDVD    0
04026 R 102151 R            JMS     %FG,            /GET ADDRESS OF ARGUMENT
04027 R 000000 A    %DP6    0                       /ADDRESS OF DIVISOR
04030 R 102300 R            JMS     %FH,            /HOLD DIVIDEND
04031 R 103722 R            JMS     DLAC
04032 R 404027 R            %DP6+400000
04033 R 103231 R    %DP6A   JMS     %FD,            /GENERAL FLOATING DIVIDE
04034 R 000000 A            CAL                     /36 BIT AUOTIENT
04035 R 000001 A                    1               /LEAST SIGNIFICANT QUOTIENT BIT
04036 R 102310 R            JMS     %FIR,           /ROUND AND INSERT SIGN
04037 R 000000 A                    0               /ROUND OFF BIT
04040 R 777777 A                    777777          /MASK
04041 R 624025 R            JMP*    DDVD            /EXIT
                            ,EJECT
```

```
                        /DOUBLE PRECISION FLOATING POINT REVERSE SUBTRACT (DSBR)
                        /CALLING SEQUENCE:
                        /
                        /       JMS     DSBR    /SUBTRAHEND IN FLOATING ACC,
                        /       XCT     ADDR    /ADDRESS OF MINUEND
                        /       NEXT INSTRUCTION        /DIFFERENCE IN FLOATING ACC,
                        /
04042 R 000000 A    DSBR    0
04043 R 102151 R            JMS     %FG,
04044 R 000000 A    %DP7    0
04045 R 103755 R            JMS DDAC
04046 R 003714 R            ,DSA %J1
04047 R 103722 R            JMS DLAC
04050 R 404044 R            %DP7+400000
04051 R 102142 R            JMS     %FNEG,          /NEGATE SUBTRAHEND
04052 R 104055 R            JMS     DFAD            /ADD DOUBLE
04053 R 003714 R                    %J1     /MINUEND-SUBTRAHEND
04054 R 624042 R            JMP*    DSBR
                        /
                        /--------------------
                        /
                        /DOUBLE PRECISION FLOATING POINT ADD (DFAD)
                        /CALLING SEQUENCE:
                        /
                        /       JMS     DFAD    /AUGEND IN FLOATING ACC
                        /       ,DSA    ADDR+400000     /XCT IF INDIRECT
                        /       NEXT INSTRUCTION        /SUBR RETURN (SUM IN FLOATING ACC)
                        /
04055 R 000000 A    DFAD    0
04056 R 102151 R            JMS     %FG,            /GET ADDRESS OF ADDEND
04057 R 000000 A    %DP9    0                       /STORE ADDRESS
04060 R 102300 R            JMS     %FH,            /HOLD FLOATING ACC
04061 R 103722 R            JMS     DLAC
04062 R 404057 R            %DP9+400000
04063 R 102447 R    %DP9A   JMS     %FA,            /GENERAL FLOATING ADD
04064 R 000043 A                    43              /MAX SHIFT
04065 R 102310 R            JMS     %FIR,           /ROUND AND SIGN
04066 R 000000 A                    0               /LEAST SIGNIFICANT BIT
04067 R 777777 A                    777777          /MASK
04070 R 624055 R            JMP*    DFAD
                            ,EJECT
```

```
                        ,TITLE SUBTITLE SINGLE
                        ,IFUND  %NSING
                    /SINGLE PRECISION FLOATING POINT PACKAGE
                    /
                    /
                    /                                   SINGLE PRECISION
                    /
                    /
                    /       FLAC                        LOAD
                    /       FDAC                        DEPOSIT
                    /       FAD                         ADD
                    /       FSUB                        SUBTRACT
                    /       FMPY                        MULTIPLY
                    /       FDVD                        DIVIDE
                    /       FSBR                        REVERSE SUBTRACT
                    /       FDVR                        REVERSE DIVIDE
                    /
                    /
                    /
                    /REQUIRED PROGRAMS: FPOINT(GENERAL FLOATING ARITHMETIC)
                    /
                    /TERMS: 1, A-REG PDP-15 HARDWARE ACCUMULATOR
                    /       2, FLT ACC FLOATING ACCUMULATOR (%FAC1,%FAC2,%FAC3)
                    /       3, HELD ACC HELD ACCUMULATOR (%HAC1, %HAC2,%HAC3)
                    /
                    /
                    /-------------------
                    /
                    /FLOATING POINT LOAD (FLAC)
                    /CALLING SEQUENCE:
                    /       JMS     FLAC            /SUBR CALL
                    /       ,DSA    ADDR+400000     /ADDR OF S,P, FLOATING POINT NUMBER
                    /       NEXT INSTRUCTION                /SUBR RETURN
                    /
                    /
04071 R 000000 A    FLAC    0
04072 R 102151 R            JMS     %FG,    /GET ADDRESS
04073 R 000000 A    %SP1    0               /(ADDRESS OF SINGLE PRECISION WORD)
04074 R 224073 R            LAC*    %SP1    /HIGH-ORDER WORD AND SPLIT
04075 R 504630 R            AND     (777000 /GET LOW-ORDER MANTISSA
04076 R 043706 R            DAC     %FAC3
04077 R 224073 R            LAC*    %SP1    /SPLIT AGAIN
04100 R 504655 R            AND     (000777 /GET EXPONENT
04101 R 244577 R            XOR     (000400 /EXTEND SIGN BIT TO FILL OUT EXPONENT WORD,
04102 R 344710 R            TAD     (777400
04103 R 043704 R            DAC     %FAC1
04104 R 444073 R            ISZ     %SP1    /BUMP TO SECOND WORD
04105 R 224073 R            LAC*    %SP1
04106 R 504660 R            AND     (377777
04107 R 043705 R            DAC     %FAC2
04110 R 224073 R            LAC*    %SP1
04111 R 504626 R            AND     (400000
04112 R 043717 R            DAC     %J4
```

```
04113 R 203705 R            LAC     %FAC2
04114 R 343717 R            TAD     %J4
04115 R 043705 R            DAC     %FAC2
04116 R 624071 R            JMP*    FLAC    /EXIT
                      /
                      /--------------------
                      /
                      /FLOATING POINT STORE
                      /CALLING SEQUENCE;
                      /       JMS     FDAC            /SUBR CALL
                      /       XCT     ADDR            /ADDR+400000(IF INDIRECT)
                      /       NEXT INSTRUCTION                        /RETURN
                      /
                      /
04117 R 000000 A      FDAC    0
04120 R 102151 R            JMS     %FG,    /GET ADDRESS
04121 R 000000 A      %SP2    0               /(ADDRESS OF SINGLE PRECISION WORD)
04122 R 203704 R            LAC     %FAC1   /GET EXPONENT
04123 R 504655 R            AND     (000777 /STRIP OFF LEFT HALF
04124 R 064121 R            DAC*    %SP2    /STORE  EXPONENT
04125 R 203706 R            LAC     %FAC3   /GET LOW-ORDER MANTISSA
04126 R 504630 R            AND     (777000 /STRIP OF RIGHT HALF
04127 R 364121 R            TAD*    %SP2    /MERGE WITH EXPONENT
04130 R 064121 R            DAC*    %SP2    /STORE COMPLETED WORD
04131 R 444121 R            ISZ     %SP2    /POINT TO NEXT RECEIVING AREA,
04132 R 203705 R            LAC     %FAC2   /HIGH-ORDER MANTISSA
04133 R 064121 R            DAC*    %SP2    /STORE IT
04134 R 624117 R            JMP*    FDAC    /EXIT
                      /
                      /--------------------
                      /
                      /FLOATING POINT ADD                     (FAD)
                      /CALLING SEQUENCE;
                      /       JMS     FAD             /SUBR CALL
                      /       XCT     ADDR            /(XCT IF INDIRECT)
                      /       NEXT INSTRUCTION                /RETURN HERE
                      /
                      /THIS SUBROUTINE DOES THE SET-UP WORK FOR SUBROUTINE %FA,
                      /THE AUGEND IS STORED IN THE HELD ACCUMULATOR AND
                      /THE ADDEND IS LOADED INTO THE FLOATING ACCUMULATOR,
                      /ROUNDED AND THE SIGN IS INSERTED,
                      /
                      /
04135 R 000000 A      FAD     0
04136 R 102151 R            JMS     %FG,    /GET ADDRESS
04137 R 000000 A      %SP3    0               /(ADDRESS OF SINGLE PRECISION ADDEND)
04140 R 102300 R            JMS     %FH,    /MOVE FLOATING ACC TO HELD ACC,
04141 R 104071 R            JMS     FLAC    /LOAD FLOATING AC
04142 R 404137 R            %SP3+400000
04143 R 102447 R            JMS     %FA,    /GENERAL FLOATING ADD
04144 R 000043 A                    43
04145 R 102310 R            JMS     %FIR,   /FLOATING INSERT AND ROUNT
04146 R 000400 A                    400     /ROUND
```

```
04147 R 777000 A                    /77700Ø  /MASK
04150 R 624135 R            JMP*    FAD      /RETURN
                    /FLOATING POINT SUBTRACT
                    /CALLING SEQUENCE:
                    /       JMS     FSUB     /SUBR CALL
                    /       XCT     ADDR     /XCT IF INDIRECT
                    /       NEXT INSTRUCTION         /SUBR RETURN
                    /
                    /FLOATING POINT SUBTRACT DOES THE SETUP WORK REQUIRED TO
                    /PERFORM THE FLOATING POINT ADD INSTRUCTION, THE MINUEND
                    /IS NEGATED AND UPON RETURNING FROM %FA, THE SUM IS
                    /NEGATED, THIS IS DONE TO TAKE FULL ADVANTAGE OF %FA, AND
                    /TO MINIMIZE CORE,
                    /
04151 R 000000 A    FSUB    Ø
04152 R 102151 R            JMS     %FG,     /GET ADDRESS OF SUBTRAHEND
04153 R 000000 A    %SP4    Ø                /STORE ADDRESS
04154 R 102142 R            JMS     %FNEG,   /NEGATE FLOATING ACC
04155 R 104135 R            JMS     FAD
04156 R 404153 R            %SP4+4ØØØØØ      /,DSA % SP4+4ØØØØØ (INDIRECT)
04157 R 624151 R            JMP*    FSUB     /EXIT
                    /
                    /-------------------
                    /
                    /FLOATING POINT MULTIPLY (FMPY)
                    /CALLING SEQUENCE:
                    /       JMS     FMPY     /ENTRY-EXIT
                    /       XCT     ADDR     /ADDRESS OF MULTIPLIER(XCT IF INDIRECT)
                    /       NEXT INSTRUCTION         /RETURN HERE
                    /
                    /FMPY DOES THE SETUP WORK NECESSARY TO EXECUTE THE GENERAL
                    /FLOATING POINT MULTIPLY INSTRUCTION: NAMELY,STORING THE
                    /MULTIPLICAND IN THE HELD ACCUMULATOR AND
                    /LOADING THE FLOATING POINT ACCUMULATOR WITH
                    /MULTIPLIER
                    /
04160 R 000000 A    FMPY    Ø
04161 R 102151 R            JMS     %FG,     /GET ADDRESS
04162 R 000000 A    %SP5    Ø                /ADDRESS OF SINGLE PRECISION MULTIPLICAND,
04163 R 102300 R            JMS     %FH,     /STORE IN HELD ACC,
04164 R 104071 R            JMS     FLAC     /LOAD MULTIPLIER
04165 R 404162 R            %SP5+4ØØØØØ      /INDIRECT ADDRESS
04166 R 103061 R            JMS     %FM,     /GENERAL FLOATING POINT MULTIPLY
04167 R 102310 R            JMS     %FIR,    /ROUND AND INSERT SIGN OF PRODUCT
04170 R 000400 A                    4ØØ
04171 R 777000 A                    777000   /MASK FOR ROUND OFF
04172 R 624160 R            JMP*    FMPY     /EXIT
                            ,EJECT
```

```
                        /--------------------
                        /
                        /FLOATING POINT DIVIDE (FDVD)
                        /CALLING SEQUENCE:
                        /       JMS     FDVD    /SUBR CALL
                        /       XCT     ADDR            /ADDR, OF DIVISOR
                        /       NEXT INSTRUCTION        /SUBR RETURN
                        /
                        /FDVD DOES THE SET UP FOR %FD, THE GENERAL PURPOSE FLOATING
                        /POINT INSTRUCTION, IT DOES THIS BY
                        /STORING THE DIVIDEND IN THE HELD ACC AND BY STORING
                        /DIVISOR IN THE FLOATING ACC.
                        /
                        /
04173 R 000000 A    FDVD    0
04174 R 102151 R            JMS     %FG,    /GET ADDRESS OF DIVISOR
04175 R 000000 A    %SP6    0               /ADDRESS OF DIVISOR
04176 R 102300 R            JMS     %FH,    /HOLD IT
04177 R 104071 R            JMS     FLAC
04200 R 404175 R            %SP6+400000     /LOAD DIVISOR
04201 R 103261 R            JMS     %FD,    /GENERAL FLOATING DIVIDE
04202 R 000000 A            CAL
04203 R 000001 A                    1       /QUOTIENT BIT
04204 R 102310 R            JMS     %FIR,   /INSERT SIGN,ROUNDOFF
04205 R 000400 A                    400
04206 R 777000 A                    777000
04207 R 624173 R            JMP*    FDVD
                        .EJECT
```

```
                        /--------------------
                        /
                        /REVERSE SUBTRACT       (FSBR)
                        /CALLING SEQUENCE:
                        /       JMS     FSBR    /SUBR CALL
                        /       XCT     ADDR    /ADDR OF MINUEND(XCT IF INDIRECT)
                        /       NEXT INSTRUCTION        /SUBR RETURN DIFFERENCE IN FLOATING ACC.
                        /
                        /
04210 R 000000 A    FSBR    0
04211 R 102151 R            JMS     %FG,    /GET ADDRESS OF MINUEND
04212 R 000000 A    %SP7    0
04213 R 104117 R            JMS FDAC
04214 R 003714 R            ,DSA %J1
04215 R 104071 R            JMS FLAC
04216 R 404212 R            %SP7+400000
04217 R 102142 R            JMS     %FNEG,  /NEGATE SUBTRAHEND
04220 R 104135 R            JMS     FAD     /ADD
04221 R 003714 R            ,DSA    %J1     /MINUEND-SUBTRAHEND
04222 R 624210 R            JMP*    FSBR
                        /
                        /
                        /--------------------
                        /
                        /REVERSE DIVIDE (FDVR)
                        /CALLING SEQUENCE:
                        /       JMS     FDVR    /CALL,DIVISOR IN FLOATING ACC,
                        /       XCT     ADDR    /ADDR OF DIVIDEND(ACT IF INDIRECT)
                        /       NEXT INSTRUCTION        /SUBR RETURN(QUOTIENT IN FLOATING ACC,)
                        /
                        /
04223 R 000000 A    FDVR    0
04224 R 102151 R            JMS     %FG,    /GET ADDRESS OF DIVIDEND
04225 R 000000 A    %SP8    0
04226 R 104117 R            JMS     FDAC    /SAVE DIVISOR
04227 R 003714 R            ,DSA    %J1     /USE HELD ACCUMULATOR TEMPORARILY
04230 R 104071 R            JMS     FLAC
04231 R 404225 R            %SP8+400000     /(DIVIDEND)
04232 R 104173 R            JMS     FDVD
04233 R 003714 R            ,DSA    %J1     /(ADDR OF DIVISOR)
04234 R 624223 R            JMP*    FDVR
                            ,EJECT
```

```
                        /TEST FOR DATA 'SW10' WHICH INHIBITS THE 'FPP15' TEST,
                        /
04235 R 000000 A    HOLDSW  0
04236 R 750004 A            LAS
04237 R 500010 R            AND     USERSW+10               /DATA 'SW10'
04240 R 741200 A            SNA                     /IS SWITCH SET?
04241 R 624235 R            JMP*    HOLDSW          /NO, EXIT
04242 R 200073 R            LAC     DSTSW0
04243 R 044546 R            DAC     SAVDST
04244 R 204713 R            LAC     (HOLD,1         /YES, HALT PROGRAM
04245 R 040073 R            DAC     DSTSW0
04246 R 604256 R            JMP     EXITM5          /EXIT '-5'
04247 R 204546 R    HOLD,1  LAC     SAVDST
04250 R 040073 R            DAC     DSTSW0          /RESTORE RETURN ADDRESS
04251 R 604236 R            JMP     HOLDSW+1        /RETEST SWITCH
                        /SUBROUTINE TO UPDATE "DSTSW0" ON EXITING
                        /
04252 R 000000 A    PCNEXT  0
04253 R 204252 R            LAC     PCNEXT
04254 R 040073 R            DAC     DSTSW0
04255 R 604256 R            JMP     EXITM5
                        /SUBROUTINE TO EXIT '-5',
                        /
04256 R 777773 A    EXITM5  LAW     -5
04257 R 040020 R            DAC     SYSERR
04260 R 620062 R            JMP*    SERVICE
                        /SETUP TO TYPE 'DONE',
                        /
04261 R 777774 A    EXITM4  LAW     -4
04262 R 040020 R            DAC     SYSERR
04263 R 140021 R            DZM     SYSERR+1
04264 R 140022 R            DZM     ERCODE
04265 R 620062 R            JMP*    SERVICE
                        /ERROR 1, FP15 CONFIGURATION NOT PRESENT
04266 R 777777 A    ERROR1  LAW     -1
04267 R 040020 R            DAC     SYSERR
04270 R 204643 R            LAC     (1                  /ERROR CODE
04271 R 040022 R            DAC     ERCODE
04272 R 777777 A            LAW     -1
04273 R 040021 R            DAC     ERCODE-1
04274 R 620062 R            JMP*    SERVICE
                        /RE-SET 'DSTSW0' TO CONTINUE AFTER TYPING ERROR,
04275 R 000000 A    EXTERR  0
04276 R 200073 R            LAC     DSTSW0
04277 R 044562 R            DAC     SAVADR#
04300 R 204714 R            LAC     (,+3
04301 R 040073 R            DAC     DSTSW0
04302 R 620062 R            JMP*    SERVICE
04303 R 204562 R            LAC     SAVADR
04304 R 040073 R            DAC     DSTSW0
04305 R 624275 R            JMP*    EXTERR
                            .EJECT
```

```
04306 R 000000 A    SETDMP  0
04307 R 144544 R            DZM     APION           /IS 'API' ON?
04310 R 705512 A            RPL
04311 R 044544 R            DAC     APION           /SET 'API' SWITCH
04312 R 705514 A            ISA+10                  /TURN 'API' OFF
04313 R 700002 A            IOF                     /TURN INTERRUPT OFF,
04314 R 717300 A            DMN                     /TURN ON FPP DIAG, MODE
04315 R 000000 A            0
04316 R 710314 A            FPT                     /DUMMY INSTRUCTION
04317 R 000000 A            0
04320 R 710177 A            DSR+77                  /READ REGISTERS,
04321 R 000547 R            REGADR                  /STORE DATA IN TABLE
04322 R 710100 A            DSR
04323 R 000547 R            REGADR
04324 R 703304 A            DBK                     /TURN DIAG MODE OFF,
04325 R 700042 A            ION                     /ENABLE 'PI'
04326 R 204544 R            LAC     APION
04327 R 504626 R            AND     (400000
04330 R 705504 A            ISA                     /ENABLE 'API' IF IT WAS ON,
04331 R 624306 R            JMP*    SETDMP
                    /ERROR 2
                    /DATA ERROR
                            /
04332 R 000000 A    ERROR2  0
04333 R 777776 A            LAW     -2
04334 R 040020 R            DAC     SYSERR
04335 R 777777 A            LAW     -1
04336 R 040021 R            DAC     SYSERR+1
04337 R 204644 R            LAC     (2
04340 R 040022 R            DAC     ERCODE
04341 R 104275 R            JMS     EXTERR          /TYPE ERROR 2 HEADER
04342 R 104427 R            JMS     ERROR           /DUMP DATA REGISTERS
04343 R 624332 R            JMP*    ERROR2
                            /
                    /ERROR 3
                    /UNEXPECTED 'JEA' BREAK
                            /
04344 R 040023 R    ERROR3  DAC     ERCODE+1        /SAVE NUMBER OF ERROR
04345 R 777776 A            LAW     -2
04346 R 040020 R            DAC     SYSERR
04347 R 777776 A            LAW     -2
04350 R 040021 R            DAC     SYSERR+1
04351 R 204645 R            LAC     (3
04352 R 040022 R            DAC     ERCODE
04353 R 104275 R            JMS     EXTERR
04354 R 104427 R            JMS     ERROR
04355 R 600444 R            JMP     FPABORT
                            .EJECT
```

```
                        /ERROR 4
                        /EXPECTED 'JEA' BREAK BUT NONE OCCURRED
                                /
04356 R 040023 R    ERROR4  DAC     ERCODE+1                /SAVE NUMBER OF ERROR,
04357 R 777776 A            LAW     -2
04360 R 040020 R            DAC     SYSERR
04361 R 777776 A            LAW     -2
04362 R 040021 R            DAC     SYSERR+1
04363 R 204646 R            LAC     (4
04364 R 040022 R            DAC     ERCODE
04365 R 104275 R            JMS     EXTERR
04366 R 104427 R            JMS     ERROR
04367 R 600444 R            JMP     FPABORT
                                /
                        /ERROR 5
                        /LOAD AND STORE FAILURE
                                /
04370 R 777776 A    ERROR5  LAW     -2
04371 R 040020 R            DAC     SYSERR
04372 R 777777 A            LAW     -1
04373 R 040021 R            DAC     SYSERR+1
04374 R 204647 R            LAC     (5
04375 R 040022 R            DAC     ERCODE
04376 R 104275 R            JMS     EXTERR
04377 R 104427 R            JMS     ERROR
04400 R 600444 R            JMP     FPABORT
                                /
                        /ERROR 6
                        /AN UNEXPECTED 'BRANCH' OCCURRED
                                /
04401 R 777776 A    ERROR6  LAW     -2
04402 R 040020 R            DAC     SYSERR
04403 R 777776 A            LAW     -2
04404 R 040021 R            DAC     SYSERR+1
04405 R 204650 R            LAC     (6
04406 R 040022 R            DAC     ERCODE          /#1, ERROR CODE 6
04407 R 201764 R            LAC     NBRANCH
04410 R 040023 R            DAC     ERCODE+1        /#2, FAILING 'BRANCH' IOT
04411 R 104275 R            JMS     EXTERR
04412 R 104427 R            JMS     ERROR
04413 R 600444 R            JMP     FPABORT
                            ,EJECT
```

```
                        /ERROR 7
                        /NO 'BRANCH' OCCURRED WHEN EXPECTED,
                                /
04414 R 777776 A        ERROR7  LAW     -2
04415 R 040020 R                DAC     SYSERR
04416 R 777776 A                LAW     -2
04417 R 040021 R                DAC     SYSERR+1
04420 R 204651 R                LAC     (7
04421 R 040022 R                DAC     ERCODE          /#1, ERROR CODE 7
04422 R 201772 R                LAC     YBRANCH
04423 R 040023 R                DAC     ERCODE+1        /#2, FAILING 'BRANCH' IOT
04424 R 104275 R                JMS     EXTERR
04425 R 604427 R                JMP     ERROR
04426 R 600444 R                JMP     FPABORT
                        /*************************************************************
                        /
                        /ERROR FORMAT FOR ALL DATA ERRORS
                        /
                        /FP15T2-RATIO-FUNCTION-FMB H-LOW-EPA-FMA H-LOW-EPA
                        /FMA H- FMA L- EPA- (STORED DATA)- (CLACULATED DATA)
                        /
                        /*************************************************************
                                /
04427 R 000000 A        ERROR   0
04430 R 104306 R                JMS     SETDMP          /DUMP DIAG REGISTERS
04431 R 777776 A                LAW     -2
04432 R 040020 R                DAC     SYSERR
04433 R 777770 A                LAW     -10
04434 R 040021 R                DAC     SYSERR+1        /PRINT 10 DATA WORDS
04435 R 204561 R                LAC     RATIO
04436 R 344715 R                TAD     (401
04437 R 040022 R                DAC     ERCODE          /#1, RATIO
04440 R 200527 R                LAC     OPCODE
04441 R 040023 R                DAC     ERCODE+1        /#2, FUNCTION
04442 R 200517 R                LAC     %XHAC2
04443 R 040024 R                DAC     ERCODE+2        /#3, FMB-H ;BEFORE
04444 R 200520 R                LAC     %XHAC3
04445 R 040025 R                DAC     ERCODE+3        /#4, FMB-L ;BEFORE
04446 R 200516 R                LAC     %XHAC1
04447 R 040026 R                DAC     ERCODE+4        /#5, EPA ;BEFORE
                        ,EJECT
```

```
04450 R 200514 R        LAC     %XFAC2
04451 R 040027 R        DAC     ERCODE+5        /#6,FMA-H ;BEFORE
04452 R 200515 R        LAC     %XFAC3
04453 R 040030 R        DAC     ERCODE+6        /#7, FMA-L ;BEFORE
04454 R 200513 R        LAC     %XFAC1
04455 R 040031 R        DAC     ERCODE+7        /#8, EPA ;BEFORE
04456 R 104275 R        JMS     EXTERR          /DUMP DATA
04457 R 777776 A        LAW     -2
04460 R 040020 R        DAC     SYSERR
04461 R 777772 A        LAW     -6
04462 R 040021 R        DAC     SYSERR+1
04463 R 200553 R        LAC     REGADR+4
04464 R 040022 R        DAC     ERCODE          /#1, FMA H ;(DRR)
04465 R 200554 R        LAC     REGADR+5
04466 R 040023 R        DAC     ERCODE+1        /#2, FMA L ;(DRR)
04467 R 200552 R        LAC     REGADR+3
04470 R 040024 R        DAC     ERCODE+2        /#3, EPA ;(DRR)
04471 R 200532 R        LAC     %ANSR2
04472 R 040025 R        DAC     ERCODE+3        /#4, FMA H ;STORED
04473 R 200533 R        LAC     %ANSR3
04474 R 040026 R        DAC     ERCODE+4        /#5, FMA L ;STORED
04475 R 200531 R        LAC     %ANSR1
04476 R 040027 R        DAC     ERCODE+5        /#6, EPA ;STORED
04477 R 104275 R        JMS     EXTERR          /DUMP DATA,
04500 R 777776 A        LAW     -2
04501 R 040020 R        DAC     SYSERR
04502 R 777775 A        LAW     -3
04503 R 040021 R        DAC     SYSERR+1
04504 R 203705 R        LAC     %FAC2
04505 R 040022 R        DAC     ERCODE          /#1, FMA H ;CALCULATED
04506 R 203706 R        LAC     %FAC3
04507 R 040023 R        DAC     ERCODE+1        /#2, FMA L ;CALCULATED
04510 R 203704 R        LAC     %FAC1
04511 R 040024 R        DAC     ERCODE+2        /#3, EPA ;CALCULATED
04512 R 104275 R        JMS     EXTERR
04513 R 104235 R        JMS     HOLDSW          /CHECK FOR THE HOLD SWITCH,
04514 R 624427 R        JMP*    ERROR
                        ,EJECT
```

```
                        /RANDOM NUMBER GENERATOR
                                /
04515 R 000000 A    RANGEN  0
04516 R 340000 A            TAD 0
04517 R 344540 R            TAD RANA
04520 R 344541 R            TAD RANB
04521 R 344542 R            TAD RANC
04522 R 044540 R            DAC RANA
04523 R 750010 A            GLK
04524 R 343711 R            TAD %HAC2
04525 R 344540 R            TAD RANA
04526 R 344541 R            TAD RANB
04527 R 344542 R            TAD RANC
04530 R 044541 R            DAC RANB
04531 R 750010 A            GLK
04532 R 343712 R            TAD %HAC3
04533 R 344540 R            TAD RANA
04534 R 344541 R            TAD RANB
04535 R 344542 R            TAD RANC
04536 R 044542 R            DAC RANC
04537 R 624515 R            JMP* RANGEN
                            /
04540 R 721320 A    RANA    721320
04541 R 627745 A    RANB    627745
04542 R 013212 A    RANC    013212
04543 R 000000 A    PGMLOP  0
04544 R 000000 A    APION   0
04545 R 000000 A    PASCNT  0
04546 R 000000 A    SAVDST  0
04547 R 000000 A    AUTORG  0
04550 R 000000 A    TEMP1   0
04551 R 000000 A    TEMP2   0
                            .EJECT
```

```
                000000 A                .END
    04572 R 000074 R *L
    04573 R 720000 A *L
    04574 R 001432 R *L
    04575 R 007400 A *L
    04576 R 006000 A *L
    04577 R 000400 A *L
    04600 R 001400 A *L
    04601 R 002000 A *L
    04602 R 001000 A *L
    04603 R 000040 A *L
    04604 R 000000 A *L
    04605 R 000531 R *L
    04606 R 000513 R *L
    04607 R 000516 R *L
    04610 R 000100 A *L
    04611 R 000532 R *L
    04612 R 000514 R *L
    04613 R 000517 R *L
    04614 R 000533 R *L
    04615 R 000515 R *L
    04616 R 000520 R *L
    04617 R 770367 A *L
    04620 R 000010 A *L
    04621 R 003000 A *L
    04622 R 770177 A *L
    04623 R 003600 A *L
    04624 R 777747 A *L
    04625 R 000030 A *L
    04626 R 400000 A *L
    04627 R 000140 A *L
    04630 R 777000 A *L
    04631 R 777777 A *L
    04632 R 001032 R *L
    04633 R 000747 R *L
    04634 R 037777 A *L
    04635 R 777032 R *L
    04636 R 777115 R *L
    04637 R 002400 A *L
    04640 R 000037 A *L
    04641 R 000027 A *L
    04642 R 001777 A *L
    04643 R 000001 A *L
    04644 R 000002 A *L
    04645 R 000003 A *L
    04646 R 000004 A *L
    04647 R 000005 A *L
    04650 R 000006 A *L
    04651 R 000007 A *L
    04652 R 000011 A *L
    04653 R 000012 A *L
    04654 R 000013 A *L
    04655 R 000777 A *L
```

```
04656 R 000060 A *L
04657 R 000020 A *L
04660 R 377777 A *L
04661 R 200000 A *L
04662 R 000025 A *L
04663 R 000014 A *L
04664 R 000015 A *L
04665 R 000035 A *L
04666 R 777767 A *L
04667 R 000170 A *L
04670 R 000050 A *L
04671 R 716601 A *L
04672 R 716602 A *L
04673 R 716603 A *L
04674 R 716604 A *L
04675 R 716606 A *L
04676 R 716610 A *L
04677 R 716620 A *L
04700 R 000662 R *L
04701 R 077777 A *L
04702 R 777377 A *L
04703 R 000377 A *L
04704 R 000043 A *L
04705 R 002650 R *L
04706 R 300000 A *L
04707 R 100000 A *L
04710 R 777400 A *L
04711 R 004063 R *L
04712 R 004033 R *L
04713 R 004247 R *L
04714 R 004303 R *L
04715 R 000401 A *L
        SIZE=04721      NO ERROR LINES
```

AC1LOC	00521	R	AC2LOC	00522	R	AC3LOC	00523	R	ADD	006000	A
ADD15	300000	A	ADRESS	04552	R	ADSIGN	03042	R	AEOPCD	01032	R
AOPCDB	00747	R	APION	04544	R	AUTORG	04547	R	AXKX	000062	A
BAC	716620	A	BLE	716603	A	BMA	716602	A	BNA	716610	A
BPA	716604	A	BR	006400	A	BRAN	01771	R	BRANCH	04553	R
BRU	716606	A	BZA	716601	A	CAC0	02377	R	CAC0EX	02412	R
CFPU	00247	R	CFPUST	00270	R	COMPLD	04554	R	CTEST0	00304	R
CTEST1	00330	R	CTEST2	00340	R	CTEST3	00334	R	DDAC	03755	R
DDVD	04025	R	DDVR	04013	R	DFAD	04055	R	DIV	002000	A
DIVSW	00524	R	DLAC	03722	R	DLAC1	03735	R	DLAC2	03745	R
DMF	717200	A	DMN	717300	A	DMPY	04000	R	DOFEX	01226	R
DOIEX	01125	R	DP	000100	A	DPFLAD	01071	R	DPFLDV	01077	R
DPFLEX	01214	R	DPFLML	01075	R	DPFLRD	01103	R	DPFLRS	01101	R
DPFLSB	01073	R	DPFRES	01573	R	DPIEX	01121	R	DPIEX1	01157	R
DPIEX2	01200	R	DPIRES	01625	R	DRR	710000	A	DSBR	04042	R
DSR	710100	A	DSTSW0	00073	R	DSUB	03771	R	DV	712000	A
DVBY0	01465	R	DVBY00	00542	R	EINT11	01540	R	EINT12	01542	R
EINT13	01546	R	EINT14	01550	R	EINT15	01552	R	EINT30	01544	R
EINT35	01554	R	EOPCDB	00745	R	ERCODE	00022	R	ERROR	04427	R
ERROR1	04266	R	ERROR2	04332	R	ERROR3	04344	R	ERROR4	04356	R
ERROR5	04370	R	ERROR6	04401	R	ERROR7	04414	R	ESGLOC	00511	R
EXITM4	04261	R	EXITM5	04256	R	EXPLOC	00512	R	EXTERR	04275	R
FACLOC	00510	R	FAD	04135	R	FDAC	04117	R	FDVD	04173	R
FDVR	04223	R	FEXIT	00505	R	FFAIL	04555	R	FL	000040	A
FLAC	04071	R	FMPY	04160	R	FNM	713250	A	FNXSA	01351	R
FPABOR	00444	R	FPACTV	00526	R	FPALLO	00441	R	FPANSR	00352	R
FPERRO	00436	R	FPEXER	00074	R	FPFMA	00225	R	FPFMB	00350	R
FPLOOP	00205	R	FPPADD	01033	R	FPPDIV	01052	R	FPPMUL	01045	R
FPPRDV	01064	R	FPPRSB	01057	R	FPPSUB	01040	R	FPRAN	00656	R
FPT	710314	A	FPULD	00224	R	FPUST	00351	R	FPUXA1	00127	R
FPUXA2	00142	R	FPUXA3	00163	R	FPUX2A	00155	R	FSBR	04210	R
FSTSUB	04556	R	FSUB	04151	R	GOGOGO	03054	R	GOTOGO	03035	R
GUARD	04557	R	HOLDSW	04235	R	HOLD.1	04247	R	IDEXIT	03445	R
IDIFR	03454	R	IDIVID	03405	R	IDVBY0	00544	R	IDVC	03455	R
IDVC3	03476	R	IDVC4	03512	R	IDV2	03440	R	IFREST	01556	R
IM	000200	A	INCEXP	03664	R	INIT	00032	R	INITAD	01776	R
JEADRS	04560	R	KT2000	00546	R	K1000	00545	R	LD	003000	A
LJE	715400	A	LOOP1	00465	R	MUL	001400	A	NBRANC	01764	R
NEXMEM	01476	R	NOBRAN	01762	R	OCHECK	03652	R	OPCDBF	00662	R
OPCODE	00527	R	OP6TO9	007400	A	OVFLO	01442	R	OVRFLR	00540	R
OVRFLX	00543	R	OVRFL1	00536	R	PASCNT	04545	R	PCNEXT	04252	R
PGMLOP	04543	R	RANA	04540	R	RANB	04541	R	RANC	04542	R
RANDAT	00567	R	RANDT1	00630	R	RANDT2	00644	R	RANGEN	04515	R
RATIO	04561	R	RDIV	002400	A	REGADR	00547	R	RSUB	001000	A
RSW	00525	R	SAVADR	04562	R	SAVDST	04546	R	SBRAN	01752	R
SERVIC	00062	R	SETDMP	04306	R	SJE	715600	A	SOFCOD	01415	R
SOFEX	01262	R	SOFNR	01310	R	SOFNRX	01405	R	SOFTEX	04563	R
SOF1	01326	R	SOF2	01335	R	SOF3	01401	R	SPFLAD	01105	R
SPFLDV	01113	R	SPFLEX	01244	R	SPFLML	01111	R	SPFLRD	01117	R
SPFLRS	01115	R	SPFLSB	01107	R	SPFRES	01557	R	SPIRES	01606	R
ST	003600	A	SUB	000400	A	SWITCH	04564	R	SYSERR	00020	R
TBAC	01746	R	TBLE	01711	R	TBMA	01703	R	TBNA	01736	R
TBPA	01725	R	TBRU	01733	R	TBZA	01660	R	TBZA1	01670	R

```
TEMP    00530 R    TEMP1   04550 R    TEMP2   04551 R    TEST1   00411 R
TEST2   00432 R    TEST3   00426 R    TFBRA   01635 R    TNEXMM  01500 R
TPRT    00353 R    TPRT1   00367 R    TPRT2   00400 R    UN      000020 A
UNDFLN  00541 R    UNDFL1  00537 R    UNFLO   01454 R    UR      000010 A
USERSW  00000 R    UU      000030 A   XFLOW   01502 R    XJEA    01432 R
YBRANC  01772 R    %ANEG,  02013 R    %ANSR1  00531 R    %ANSR2  00532 R
%ANSR3  00533 R    %DP1    03724 R    %DP2    03757 R    %DP3    03773 R
%DP4    04002 R    %DP5    04015 R    %DP6    04027 R    %DP6A   04033 R
%DP7    04044 R    %DP9    04057 R    %DP9A   04063 R    %FAA    02541 R
%FAB    02550 R    %FABB1  02624 R    %FABB2  02637 R    %FABB3  02644 R
%FAB,B  02600 R    %FAB,C  02650 R    %FAC    02705 R    %FAC1   03704 R
%FAC2   03705 R    %FAC3   03706 R    %FAC4   03707 R    %FAC4A  04565 R
%FAD    02750 R    %FAD,D  02762 R    %FAEX   03017 R    %FAEXT  03056 R
%FAEXX  03012 R    %FAE,X  03000 R    %FANG1  02774 R    %FAT,B  02516 R
%FA,    02447 R    %FA,T   02505 R    %FDA    03531 R    %FDB    03552 R
%FDC    03563 R    %FDD    03602 R    %FDE    03616 R    %FDEXI  03600 R
%FDF    03622 R    %FD,    03231 R    %FD0    03275 R    %FD1    03343 R
%FD2    03323 R    %FD3    03340 R    %FD4    03331 R    %FE     03010 R
%FGEX,  02170 R    %FG,    02151 R    %FH,    02300 R    %FIRA   02356 R
%FIREX  02331 R    %FIR,   02310 R    %FIR,A  02334 R    %FIXB   02121 R
%FIX,   02102 R    %FLOT,  02024 R    %FLOT1  02065 R    %FLOT2  02070 R
%FMA    03154 R    %FMB    03222 R    %FMC    03225 R    %FM,    03061 R
%FM1    03132 R    %FM2    03112 R    %FM2A   03120 R    %FM3    03127 R
%FNEG,  02142 R    %FNORA  02224 R    %FNORB  02267 R    %FNOR,  02172 R
%FN,    02260 R    %FSIGN  03721 R    %FS,    02360 R    %F%     04566 R
%HAC1   03710 R    %HAC2   03711 R    %HAC3   03712 R    %HAC4   03713 R
%J1     03714 R    %J1A    04567 R    %J2     03715 R    %J3     03716 R
%J4     03717 R    %J4%    04570 R    %J5     03720 R    %MQ     04571 R
%NA1    00534 R    %NA2    00535 R    %PFAC   02670 R    %RL     02572 R
%SNEG,  02017 R    %SP1    04073 R    %SP2    04121 R    %SP3    04137 R
%SP4    04153 R    %SP5    04162 R    %SP6    04175 R    %SP7    04212 R
%SP8    04225 R    %SWICH  02423 R    %XFAC1  00513 R    %XFAC2  00514 R
%XFAC3  00515 R    %XHAC1  00516 R    %XHAC2  00517 R    %XHAC3  00520 R
```

```
USERSW  00000 R   UR      000010 A   SYSERR  00020 R   UN      000020 A
ERCODE  00022 R   UU      000030 A   INIT    00032 R   FL      000040 A
AXKX    000062 A  SERVIC   00062 R   DSTSW0  00073 R   FPEXER   00074 R
DP      000100 A  FPUXA1   00127 R   FPUXA2  00142 R   FPUX2A   00155 R
FPUXA3   00163 R  IM      000200 A   FPLOOP  00205 R   FPULD    00224 R
FPFMA    00225 R  CFPU     00247 R   CFPUST  00270 R   CTEST0   00304 R
CTEST1   00330 R  CTEST3   00334 R   CTEST2  00340 R   FPFMB    00350 R
FPUST    00351 R  FPANSR   00352 R   TPRT    00353 R   TPRT1    00367 R
SUB     000400 A  TPRT2    00400 R   TEST1   00411 R   TEST3    00426 R
TEST2    00432 R  FPERRO   00436 R   FPALLO  00441 R   FPABOR   00444 R
LOOP1    00465 R  FEXIT    00505 R   FACLOC  00510 R   ESGLOC   00511 R
EXPLOC   00512 R  %XFAC1   00513 R   %XFAC2  00514 R   %XFAC3   00515 R
%XHAC1   00516 R  %XHAC2   00517 R   %XHAC3  00520 R   AC1LOC   00521 R
AC2LOC   00522 R  AC3LOC   00523 R   DIVSW   00524 R   RSW      00525 R
FPACTV   00526 R  OPCODE   00527 R   TEMP    00530 R   %ANSR1   00531 R
%ANSR2   00532 R  %ANSR3   00533 R   %NA1    00534 R   %NA2     00535 R
OVRFL1   00536 R  UNDFL1   00537 R   OVRFLR  00540 R   UNDFLN   00541 R
DVBY00   00542 R  OVRFLX   00543 R   IDVBY0  00544 R   K1000    00545 R
KT2000   00546 R  REGADR   00547 R   RANDAT  00567 R   RANDT1   00630 R
RANDT2   00644 R  FPRAN    00656 R   OPCDBF  00662 R   EOPCDB   00745 R
AOPCDB   00747 R  RSUB    001000 A   AEOPCD  01032 R   FPPADD   01033 R
FPPSUB   01040 R  FPPMUL   01045 R   FPPDIV  01052 R   FPPRSB   01057 R
FPPRDV   01064 R  DPFLAD   01071 R   DPFLSB  01073 R   DPFLML   01075 R
DPFLDV   01077 R  DPFLRS   01101 R   DPFLRD  01103 R   SPFLAD   01105 R
SPFLSB   01107 R  SPFLML   01111 R   SPFLDV  01113 R   SPFLRS   01115 R
SPFLRD   01117 R  DPIEX    01121 R   DOIEX   01125 R   DPIEX1   01157 R
DPIEX2   01200 R  DPFLEX   01214 R   DOFEX   01226 R   SPFLEX   01244 R
SOFEX    01262 R  SOFNR    01310 R   SOF1    01326 R   SOF2     01335 R
FNXSA    01351 R  MUL     001400 A   SOF3    01401 R   SOFNRX   01405 R
SOFCOD   01415 R  XJEA     01432 R   OVFLO   01442 R   UNFLO    01454 R
OVBY0    01465 R  NEXMEM   01476 R   TNEXMM  01500 R   XFLOW    01502 R
EINT11   01540 R  EINT12   01542 R   EINT30  01544 R   EINT13   01546 R
EINT14   01550 R  EINT15   01552 R   EINT35  01554 R   IFREST   01556 R
SPFRES   01557 R  DPFRES   01573 R   SPIRES  01606 R   DPIRES   01625 R
TFBRA    01635 R  TBZA     01660 R   TBZA1   01670 R   TBMA     01703 R
TBLE     01711 R  TBPA     01725 R   TBRU    01733 R   TBNA     01736 R
TBAC     01746 R  SBRAN    01752 R   NOBRAN  01762 R   NBRANC   01764 R
BRAN     01771 R  YBRANC   01772 R   INITAD  01776 R   DIV     002000 A
%ANEG,   02013 R  %SNEG,   02017 R   %FLOT,  02024 R   %FLOT1   02065 R
%FLOT2   02070 R  %FIX,    02102 R   %FIXB   02121 R   %FNEG,   02142 R
%FG,     02151 R  %FGEX,   02170 R   %FNOR,  02172 R   %FNORA   02224 R
%FN,     02260 R  %FNORB   02267 R   %FH,    02300 R   %FIR,    02310 R
%FIREX   02331 R  %FIR,A   02334 R   %FIRA   02356 R   %FS,     02360 R
CAC0     02377 R  RDIV     002400 A   CAC0EX  02412 R   %SWICH   02423 R
%FA,     02447 R  %FA,T    02505 R   %FAT,B  02516 R   %FAA     02541 R
%FAB     02550 R  %RL      02572 R   %FAB,B  02600 R   %FABB1   02624 R
%FABB2   02637 R  %FABB3   02644 R   %FAB,C  02650 R   %PFAC    02670 R
%FAC     02705 R  %FAD     02750 R   %FAD,D  02762 R   %FANG1   02774 R
LD      003000 A  %FAE,X   03000 R   %FE     03010 R   %FAEXX   03012 R
%FAEX    03017 R  GOTOGO   03035 R   ADSIGN  03042 R   GOGOGO   03054 R
%FAEXT   03056 R  %FM,     03061 R   %FM2    03112 R   %FM2A    03120 R
%FM3     03127 R  %FM1     03132 R   %FMA    03154 R   %FMB     03222 R
%FMC     03225 R  %FD,     03231 R   %FD0    03275 R   %FD2     03323 R
```

%FD4	03331	R	%FD3	03340	R	%FD1	03343	R	IDIVID	03405	R
IDV2	03440	R	IDEXIT	03445	R	IDIFR	03454	R	IDVC	03455	R
IDVC3	03476	R	IDVC4	03512	R	%FDA	03531	R	%FDB	03552	R
%FDC	03563	R	ST	003600	A	%FDEXI	03600	R	%FDD	03602	R
%FDE	03616	R	%FDF	03622	R	OCHECK	03652	R	INCEXP	03664	R
%FAC1	03704	R	%FAC2	03705	R	%FAC3	03706	R	%FAC4	03707	R
%HAC1	03710	R	%HAC2	03711	R	%HAC3	03712	R	%HAC4	03713	R
%J1	03714	R	%J2	03715	R	%J3	03716	R	%J4	03717	R
%J5	03720	R	%FSIGN	03721	R	DLAC	03722	R	%DP1	03724	R
DLAC1	03735	R	DLAC2	03745	R	DDAC	03755	R	%DP2	03757	R
DSUB	03771	R	%DP3	03773	R	DMPY	04000	R	%DP4	04002	R
DDVR	04013	R	%DP5	04015	R	DDVD	04025	R	%DP6	04027	R
%DP6A	04033	R	DSBR	04042	R	%DP7	04044	R	DFAD	04055	R
%DP9	04057	R	%DP9A	04063	R	FLAC	04071	R	%SP1	04073	R
FDAC	04117	R	%SP2	04121	R	FAD	04135	R	%SP3	04137	R
FSUB	04151	R	%SP4	04153	R	FMPY	04160	R	%SP5	04162	R
FDVD	04173	R	%SP6	04175	R	FSBR	04210	R	%SP7	04212	R
FDVR	04223	R	%SP8	04225	R	HOLDSW	04235	R	HOLD,1	04247	R
PCNEXT	04252	R	EXITM5	04256	R	EXITM4	04261	R	ERROR1	04266	R
EXTERR	04275	R	SETDMP	04306	R	ERROR2	04332	R	ERROR3	04344	R
ERROR4	04356	R	ERROR5	04370	R	ERROR6	04401	R	ERROR7	04414	R
ERROR	04427	R	RANGEN	04515	R	RANA	04540	R	RANB	04541	R
RANC	04542	R	PGMLOP	04543	R	APION	04544	R	PASCNT	04545	R
SAVDST	04546	R	AUTORG	04547	R	TEMP1	04550	R	TEMP2	04551	R
ADRESS	04552	R	BRANCH	04553	R	COMPLD	04554	R	FFAIL	04555	R
FSTSUB	04556	R	GUARD	04557	R	JEADRS	04560	R	RATIO	04561	R
SAVADR	04562	R	SOFTEX	04563	R	SWITCH	04564	R	%FAC4A	04565	R
%F%	04566	R	%J1A	04567	R	%J4%	04570	R	%MQ	04571	R
ADD	006000	A	BR	006400	A	OP6TO9	007400	A	ADD15	300000	A
DRR	710000	A	DSR	710100	A	FPT	710314	A	DV	712000	A
FNM	713250	A	LJE	715400	A	SJE	715600	A	BZA	716601	A
BMA	716602	A	BLE	716603	A	BPA	716604	A	BRU	716606	A
BNA	716610	A	BAC	716620	A	DMF	717200	A	DMN	717300	A