SDS 920/930 COMPUTER
PROGRAMMED OPERATORS
TECHNICAL MANUAL

SDS 900020E July 1965

SCIENTIFIC DATA SYSTEMS/1649 Seventeenth Street/Santa Monica, California/UP1-0960

Printed in U.S.A.

©1965 Scientific Data Systems, Inc.

REVISIONS

This publication, SDS 90 00 20E, is a reprint of the SDS 920/930 Computer Programmed Operators Technical Manual, SDS 900020D. The programs and programming descriptions are unchanged. However, page numbers have been changed to section numbers to facilitate future modification and updating.

See Contents pages iii and iv, and Index pages xv and xvi for section numbering. Logical file designations are listed on the Contents pages and explained on page xiv.

CONTENTS

CONTENTS (cont.)

INTRODUCTION

The SDS Programmed Operator enables a programmer to code a subroutine call with a single instruction, just as if the subroutine were an actual machine instruction. Other computers usually perform standard subroutine calls by executing a transfer to the starting location of the subroutine and, at the same time, preserving a return address. This procedure requires an operation code (indicating a transfer) and an operand address (indicating the starting address of the subroutine). If the subroutine should require an additional operand, as in a floating-point add subroutine, for example, the calling sequence must be longer to accommodate the specification of the operand.

The SDS Programmed Operator (abbreviated POP) uses the operation code to indicate the transfer address. When the computer detects a "one" in bit position 2 of an instruction, bit positions 2 through 8 are not interpreted as a normal instruction, but instead are treated as an address to which the computer transfers control. Thus, the operand address field is free to designate an address for use by the subroutine. There are 64 (decimal) locations [$(100)_8$ through $(177)_8$] to which a transfer may occur. These 64 locations constitute a linkage table; they normally contain appropriate unconditional transfer instructions (BRU) to maintain the communication link between the POP code and the subroutine being called by it.

At the time the computer detects the POP code in the main program, four things happen. The address (contents of P register) at which the POP code occurs in the main program is stored in location 0. The indirect address bit (bit 9) is set in location 0. The contents of the overflow register are preserved in bit 0 of location 0. Transfer is made to the location given by the POP code.

For example, let location 01000 contain an instruction 010400200 and the overflow register be set (i.e., =1). As the program executes the instruction at 01000, location 0 will be set to 040041000, consisting of the address 01000; a one at bit 9 (indirect address); and a one in bit 0 (contents of the overflow register). Transfer will then be made to location 0104, which should contain an unconditional branch to the POP subroutine. If a normal BRR 0 is used to exit from the POP subroutine, return will be made to location 01001, and status of the overflow register will be re-established.

By referencing location 0 directly, the address of the location from which the transfer was made may be picked up. By referencing 0 indirectly, the contents of the location from which transfer was made may be picked up.

In the preceding illustration, for example,

```
LDA   0
```

would load the A register with the contents of 0 (i.e., 40041000). By masking, the condition of the overflow may be determined, or address may be extracted.

On the other hand,

```
LDA   *0
```

would load the A register with the contents of 01000 (i.e., 010400200). By masking, address portion may be extracted for use by the POP subroutine.

By judicious use of the programmed operator principle, a one-to-one correspondence may be maintained between SDS 920 instructions and SDS 910 instructions. For example, XMA is a 920 machine instruction; its function may be simulated on the SDS 910 by a subroutine, and this subroutine may be called by means of a programmed operator. Thus, the main program requires the same number of instructions for either the SDS 910 or 920.

The following operations take place when the computer detects a programmed operator:

1. $(P) \rightarrow (0)_{10-23}$; save P register for return address
2. $1 \rightarrow (0)_9$; insert indirect address bit
3. $(O_f) \rightarrow (0)_s$; preserve status of overflow toggle
4. $(C)_{2-8} \rightarrow (P)$; branch to location indicated on POP code

SDS 920/930 PROGRAMMED OPERATOR EXAMPLE

Location	Instruction				Effective Address	Contents of Effective Address	Location 0	O_f	A Register	B Register
MAIN PROGRAM - XMP is a POP Reference										
01342	14002163		XMP	02163	02163	00000012		Set	00000144	01234567
TRANSFER LOCATION (0140) - Assigned by Loader										
00140	00104000						400401342	Reset		
POP SUBROUTINE - Relocated by Loader at 04000										
04000	03604005	$XMP	STB	TEMPB						
04001	06440000		MUL	*0	02163	00000012			00000000	00003720
04002	06700027		LSH	27					00001750	00000000
04003	07504005		LDB	TEMPB	04005	01234567			00001750	01234567
04004	05100000		BRR	0				Set		
04005	00000000	TEMPB	PZE			01234567				
01343	Continue in main program									

Explanation: XMP is a programmed operator that produces the integer product of the integer in the A register and the integer contained in the effective address. Overflow is set if the integer product exceeds the capacity of a single register. The contents of the B and X registers are unaffected by this "instruction". In this example, XMP is assigned POP transfer address 0140 and a BRU 4000 is set in location 040 by the Loader.

SDS 900 SERIES PROGRAM LIBRARY

PROGRAM DESCRIPTION

Catalog No. 202003D

SDS 920/930 PROGRAMMED OPERATOR PACKAGE

GENERAL

This package contains a basic set of Programmed Operator routines for use with the SDS 920/930 Computers. The package comes in two forms. The POP routines can be contained as a Programmed Operator Library on the MONARCH System tape, and loaded by the MONARCH Loader. Or, they can be on paper tape or cards in Standard SDS Format and loaded by the Universal Binary Loader.

POPs may be used only in programs assembled in SYMBOL or META-SYMBOL.[†] Note that the POPs in this package may not be used with FORTRAN II, ALGOL, or REAL-TIME FORTRAN. (FORTRAN has its own set of Run-Time POPs. See FORTRAN II Operations Manual, SDS 900046D.)

HOW TO PROGRAM POPs

Each POP is represented by mnemonic which may be written exactly as a machine language instruction in coding a program. A typical sequence might be:

```
        LDA   MANTIS+1   MANTISSA
        LDB   MANTIS     EXPONENT
MULT    FLM   FLOATN     SINGLE PRECISION FLOATING MULTIPLY
```

After the execution of the FLM POP at location MULT, the A register would contain the mantissa, the B register the exponent, of the Single Precision Floating Point Multiply. Note that the POP will access the operand portion FLOATN during its execution, in this case to get the multiplier. Each POP Description contains explicit directions as to the use of the symbol or quantity in the operand field of the POP.

Warning: Use only the mnemonic forms of these standard POPs; do not attempt to generate an octal code which will induce a POP transfer at execution time. For example,

```
EX      FORM    9, 15
        :
        :
        EX      0105, 01000
```

[†]For use of POPs with SYMBOL 4 or SYMBOL 8 Assemblers, or with HELP, see Programmed Operators SDS 920 Manual, SDS 900020B.

would certainly generate a line of code of the form 010501000 at assembly time, but would be illegal for purposes of the load function. The reasons will become clear after the reader understands the sections explaining how the Assemblers and Loaders deal with POPs. If information in the A, B, and X registers is not to be used by the POP, it is saved and restored upon return to the main program.

LOADING POPs

A. MONARCH

1. Use a △LOAD instruction to load the main program, subroutines, and POP subroutines you have written yourself. If any program is relocatable, a bias must be given. The POPs will then be loaded following the last relocatable program. If, however, all the program are absolute, a bias location at which the POPs may be loaded must be given.

2. The Loader will search the MONARCH POP Library for unsatisifed POP references.

3. If a GO or STOP directive is given in the △LOAD instruction, no symbol table will be typed before execution.

   If a TGO or TSTP directive is given in the △LOAD instruction, the symbol table containing all references (program and POP) is typed. Following each reference will be an octal quantity. If bit 0 of this number is set, the reference is satisfied; if bit 1 is set, the reference is unsatisfied, and no linkages have been set up.

4. Execute the program.

B. UNIVERSAL BINARY LOADER

1. Set a starting location in the A register for relocatable program to be loaded. The POPs will be loaded following the last relocatable program. If all programs to be loaded are absolute, set in the A register the address of a location where POPs may be loaded.

2. Set breakpoints. Breakpoint settings described in Universal Binary Loader (Catalog Number 000020) are to be closely followed if loading is to be successful. Setting Breakpoints 1, 2, and 3 will ensure that the loader will halt after each program loaded.

3. Load the Universal Binary Loader by standard fill.

4. Load the main program and each subprogram.

5. Mount the POP tape or deck. Clear the halt to load.

6. If BPT 3 is set, the computer will halt with a BRU to the end transfer address in the C register. Halts may occur between (5) and (6). See Universal Binary Loader write-up for detailed explanation.

Errors: MISSING DEFS, together with a list of missing POP subroutines will be typed if all POP references in the programs are not satisfied. Note that the POP Library is the last tape/deck to be loaded.

HOW THE ASSEMBLER TREATS POPs

As the SYMBOL and META-SYMBOL Assemblers encounter mnemonics not recognized as standard machine mnemonics, Assembler directives, PROC calls, etc., they arbitrarily assign each a unique sequence number between 0100 and 0177 (for example, FLM, not a standard machine mnemonic, might be assigned sequence number 0105). Each time the same mnemonic is encountered in one program during one assembly, it is assigned the same sequence number (i.e., each time FLM is encountered it would be assigned 0105). The Assembler flags each of these with an "I" as an illegal instruction mnemonic. It merely means that it was unable to find that mnemonic in its table of standard mnemonics.

The octal representation of the generated code will contain the sequence number in bits 2-8.

e.g., FLM 0201

If FLM is given sequence number 0105, the following line of code would be generated:

I 10500201 FLM 0201

Note that the sequence number is not the number of the location to which transfer will be made during execution. A further transformation and reassignment is made during loading. (See the next section, "How the Loader Treats POPs".)

When the object tape or deck is produced by the Assembler, each data block will contain information as to which instructions are POPs. (See MONARCH Reference Manual (SDS 900566), Section III, the MONARCH LOADER, for more explicit information.)

HOW THE LOADER TREATS POPs

As each program or subprogram is loaded, the names of POP references are gathered, together with their sequence number, into a table used by the Loader. As POPs are loaded from the POP tape or MONARCH Library, the name of each POP definition in the POP Library is compared with the entries in the POP reference table created by the Loader to see if that POP subroutine should be loaded. If so, the sequence number in each POP instruction in the programs loaded is replaced by the actual POP transfer code in all locations where the POP occurs; i.e., FLM might have been given sequence number 0110 by the Assembler, but the loader might assign POP code 0124 to the FLM instruction. In this case, during execution, transfer would be made to location 0124 rather than to 0110. Location 0124 would contain finally a BRU to the FLM POP subroutine. If the Loader is successful in satisfying all POP references, no error message occurs. If any POP references are still unsatisfied, an error message "MISSING DEFS" together with their names will be

typed. After all loading is complete, the Loader initializes by setting up BRUs to POP subroutines in appropriate locations 0100 to 0177.

OTHER USE OF THE PROGRAMMED OPERATOR FACILITY

Of the four directives (OPD, POPD, FORM and DATA) capable of generating octal code to cause POP transfer at execution time, only POPD does so in a way that enables the Loaders (MONARCH and UBL) to set up appropriate linkages at load time. Use of POPD is described in detail in the next section.

OPD, FORM and DATA may be used to generate octal code which will cause POP transfer at execution time with certain limitations. For instance, the following examples all generate the line of code 012300050 at assembly time.

Example 1:

```
INST   FORM   9,15
       .
       .
       .
       INST   0123,050
```

Example 2:

```
TRAP   OPD    012300050
       .
       .
       .
       TRAP
```

Example 3:

```
       DATA   012300050
```

The danger is that with these directives the Loader does not modify bits 2-8 during load time, and a conflict with existing POP references may occur.

However, if care is exercised in selection of locations 0100-0177 that do not interfere with Loader assignments, the POP mechanism may serve as a "trapping" facility.

For example, since the Loader assigns from 0100 up, "trap" assignments could be made, numbered from 0177 down.

```
TRP1    OPD     017700000
        .
        .
        .
        TRP1    ALPHA
```

If the current value of ALPHA were 02050, the resulting instruction would be:

```
017702050       TRP1    ALPHA
```

At load time, the address could be modified for relocatability, but the bits 2-8 would not be changed.

At execution time, a POP transfer would be made to location 0177.

Note that the programmer would have to store a BRU to the "trap" subroutine in location 0177.

WRITING POP SUBROUTINES - THE POPD DIRECTIVE

If the user desires, he can write new subroutines to service any of the present POP mnemonics. To write a new subroutine for FLM, code the subroutine in the following manner:

```
$FLM    POPD
LOC1    STX     TEMP    FIRST LINE OF SUBROUTINE
        .               SUBROUTINE CODE
        .
        .

        BRR     0       RETURN TO MAIN PROGRAM
```

Note that it is not necessary to reserve the first location of the subroutine (i.e., LOC1) for the return address, since location 0 is used for this purpose. It will be unnecessary to change any coding in the main program, since FLM will cause a transfer to a POP location at execution time.

The new POP subroutine may be assembled with or separately from the main program which calls it. If the new POP subroutine is loaded after the main program but before the POP Library, it satisfies the POP reference and the subroutine is not called from the POP Library. Note that the POP subroutine is really a service subroutine, and may be coded using all the capability of SYMBOL and META-SYMBOL.

Of course, the user can define any new mnemonics or redefine standard machine mnemonics (except Assembler directives such as LDA, MIN, SKG, etc.) since POPD directive overrides system mnemonics in both SYMBOL and META-SYMBOL. For instance, a user might desire to write a fast, 12-bit binary-to-decimal conversion. The mnemonic FBD could be used as a "POP" reference, and the subroutine coded $FBD POPD, etc. Or, if the user desired to cause TRTW to incorporate several tape tests, he could assign TRTW as $TRTW POPD and code a suitable subroutine incorporating all the tests.

MODIFYING OR UPDATING THE PROGRAMMED OPERATOR LIBRARY

The POP Library is available in three forms: (1) paper tape, (2) binary cards, and (3) part of MONARCH System tape. In all three cases, this library consists of a number of "Logical Files", each a complete program with one or more external entry points. (See Contents, page iii, for Logical File Number designations.) Each Logical File has been assembled as a separate subroutine, although it may contain more than one POP definition. For example, Logical File No. 21, SQR, is a separate subroutine with one entry point, SSQR; however, Logical File No. 16, CSD, SND, is a single subroutine with two entry points, SCSD and SSND.

Note that program numbers do not necessarily correspond with Logical File numbers. For example, Cat. No. 203022-B consists of four separate subroutines: (1) DPN (Logical File No. 25); (2) DPD, DPM, DPS, DPA (Logical File No. 26); (3) LDP (Logical File No. 27); and (4) STD (Logical File No. 28). Also, a Section may contain writeups for more than one Logical File (e.g., Sections 25, 26, and 27).

All three forms (paper tape, card, MONARCH) of the POP Library follow the Logical File arrangement outlined on the Contents page. Each Logical File starts with a Type 2 (External Definition) record giving the POP definition and ends with a Type 3 (END) record. Each Logical File in the POP Library on the MONARCH tape is a △2 record under the △1 LIBRARY label.

The POP Library is generally modified for two reasons: (1) insertion and deletion of routines to create an augmented or abbreviated library; or (2) updating to replace current routines. The Logical Files have been numbered to facilitate these modifications on the paper tape version of the POP Library by means of Cat. No. 012014, SYMBOL Reproduce and Update. The card deck version may be altered by physically rearranging or editing, and the MONARCH System Tape must be modified by its own "UPDATE" Routine.

INDEX OF 920/930 PROGRAMMED OPERATORS

Mnemonic	Catalog Number	Section Number	910 Time† (milliseconds)	Space Oct/Dec	Description
ATD	203032B	14	5.0-5.5	236/158	Arctan, Fixed, Double
ATF	203026C	15	4.9-6.4	371/249	Arctan, Floating
ATN	203007B	13	1.032	71/57	Arctan of A, Fixed, Single
BDD	203037B	1	6.304-8.800	270/184	Binary-Decimal, Fixed, Double
BDF	203039B	3	12.9+065E E=dec. expon.	253/171	Binary-Decimal, Floating, Double
BFS	203014B	2	2.824	113/75	Binary-Decimal, Floating, Single
BID	203012B	7	1.904-2.240	141/97	Binary-Decimal, Fixed, Single
COS	203018B	18	0.464-0.504	40/32	Cosine of A
CSD	203034B	16	3.59-3.86	224/148	Cosine, Fixed, Double
CSF	203028C	17	4.25-5.45	331/217	Cosine, Floating
DBD	203036B	4	6.096-6.448	262/178	Decimal-Binary, Fixed, Double
DBF	203038B	6	3.8-8.5	251/169	Decimal-Binary, Floating, Double
DFS	203015B	5	2.312	64/52	Decimal-Binary, Floating, Single
DIB	203013B	8	2.400-2.784	123/83	Decimal-Binary, Fixed, Single
DPA	203016B	25	0.160	114/76	Add, Fixed, Double
DPD	203022B	25	1.016-1.400	114/76	Divide, Fixed, Double
DPM	203040B	25	0.504	114/76	Multiply, Fixed, Double
DPN	203022B	25	0.072-0.104	12/10	Negate, Fixed, Double
DPS	203017B	25	0.160	114/76	Subtract, Fixed, Double
DSQ	203035B	19	1.112-1.320	122/82	Square Root, Fixed, Double
EXF	203025C	12	2.24-7.93	237/159	Exponential (2, e, 10), Floating
EXP	203008B	11	0.824+scaling	76/62	Exponential (2, e, 10) of A
FFF	203011C	22	0.280-0.632	122/82	Fixed-Floating Conversion
FLA	203023C	24	0.656	320/208	Add, Floating, Double
FLD	203023C	24	1.072	320/208	Divide, Floating, Double
FLM	203023C	24	0.736	320/208	Multiply, Floating, Double
FLN	203023C	24	0.152	320/208	Negate, Floating, Double
FLS	203023C	24	0.784	320/208	Subtract, Floating, Divide
FSA	203010B	23	0.352-0.480	174/124	Add, Floating, Single
FSD	203010B	23	0.464-0.480	174/124	Divide, Floating, Single
FSM	203010B	23	0.248-0.264	174/124	Multiply, Floating, Single
FSN	203010B	23	0.064-0.072	174/124	Negate, Floating, Single
FSQ	203029B	20	1.056-1.080	107/71	Square Root, Floating, Double
FSS	203010B	23	0.368-0.480	174/124	Subtract, Floating, Single

†To extrapolate 930 times, multiply 920 times by a factor 1.75/8 = 0.219.

INDEX (cont.)

Mnemonic	Catalog Number	Section Number	910 Time (milliseconds)	Space Oct/Dec	Description
LDP	203022B	25	0.120	6/6	Load, Double Precision
LGF	203024C	10	3.08-6.15	230/152	Log (2, e, 10), Floating
LOG	203009B	9	904 μsec +normalize	60/48	Log (2, e, 10) of A
LQP	203021B	27	0.200	12/10	Load Quadruple Precision
LTP	203020B	26	0.160	10/8	Load, Triple precision
SIN	203006B	18	0.448-0.488	40/32	Sine of A
SND	203033B	16	3.55-3.81	224/148	Sine, Fixed, Double
SNF	203027C	17	4.2-5.3	331/217	Sine, Floating
SQR	209019B	21	0.384-0.576	124/84	Square Root of A
STD	203022B	25	0.160	6/6	Store, Double Precision
STP	203020B	26	0.216	11/9	Store, Triple Precision
STQ	203021B	27	0.256	13/11	Store, Quadruple Precision

IDENTIFICATION: Binary to Decimal Conversion, Double, Fixed - BDD

AUTHOR: Richard S. Resnick, SDS

ACCEPTED: 17 April 1963

COMPUTER CONFIGURATION: Any SDS 920/930 Computer

PURPOSE: To convert a double precision fixed point binary number in A, B with the binary point location in the address field of BDD programmed operator, into thirteen 6-bit characters in A, B and extended register locations 2 and 3, with sign, decimal point, and spacer character.

Sixteen characters (four words) are necessary to represent the final converted decimal number completely formated for output.

PROGRAMMED OPERATORS: DPN, DPA, DPS, DPM, LDP, STD

STORAGE: Instructions and constants: 270 oct, 184 dec

Uses temporary storage locations 16 through 22.

Uses extended register locations 2 through 11.

TIMING: 6.304 to 8.800 m.s.

SOURCE LANGUAGE: SYMBOL

USE:

1. FUNCTION

To convert a fixed point binary number in A, B with the binary point location in the address field of the BDD programmed operator into thirteen 6-bit BCD characters in A, B location 2 and location 3 with sign, decimal point, and spacer character. The final sixteen character decimal number with format symbols replaces the argument in A, B and locations 2 and 3 in the following manner:

USE: (cont.)

Table 1

	Range of Argument	B	A	Location 2	Location 3
i = 13	$0 \leq \|A\| < 1$	SP± . X	X X X X	X X X X	X X X X
i = 12	$1 \leq \|A\| < 1$	SP± X .	X X X X	X X X X	X X X X
i = 11	$10 \leq \|A\| < 10^2$	SP± X X	. X X X	X X X X	X X X X
i = 10	$10^2 \leq \|A\| < 10^3$	SP± X X	X . X X	X X X X	X X X X
i = 9	$10^3 \leq \|A\| < 10^4$	SP± X X	X X . X	X X X X	X X X X
i = 8	$10^4 \leq \|A\| < 10^5$	SP± X X	X X X .	X X X X	X X X X
i = 7	$10^5 \leq \|A\| < 10^6$	SP± X X	X X X X	. X X X	X X X X
i = 6	$10^6 \leq \|A\| < 10^7$	SP± X X	X X X X	X . X X	X X X X
i = 5	$10^7 \leq \|A\| < 10^8$	SP± X X	X X X X	X X . X	X X X X
i = 4	$10^8 \leq \|A\| < 10^9$	SP± X X	X X X X	X X X .	X X X X
i = 3	$10^9 \leq \|A\| < 10^{10}$	SP± X X	X X X X	X X X X	. X X X
i = 2	$10^{10} \leq \|A\| < 10^{11}$	SP± X X	X X X X	X X X X	X . X X
i = 1	$10^{11} \leq \|A\| < 10^{12}$	SP± X X	X X X X	X X X X	X X . X
i = 0	$10^{12} \leq \|A\| < 10^{13}$	SP± X X	X X X X	X X X X	X X X .

2. ARGUMENT

The argument is a double precision number in A, B with the binary point location in the 9 least significant bits of the programmed operator address. The Q (binary point location) can range from +87 to -87. If the Q exceeds this or if the absolute value of the argument is equal to or greater than 10^{13} at Q = 44, overflow will be set and the program will exit. When overflow occurs the original argument is lost.

3. ACCURACY

The output is accurate to thirteen decimal digits.

METHOD:

The argument is normalized and made positive. If Q is less than or equal to zero, the argument is shifted right Q places and handled like an argument whose Q is zero. If Q is greater than zero, the argument is shifted right 44-Q places and tested for the interval in which it lies. The intervals are denoted by subscript i and are described in Table 1.

A is then multiplied by $10^{-(i-1)}$. This results in a number greater than 1 and less than 10. This is the first decimal digit. By successively extracting off the newly formed digit, and multiplying by 10, all thirteen digits are formed. The placing of the decimal point is a function of i.

```
                                    1  $EDD   POPD    017000000
00000  0 37 00007                   2         STX     Tx
00001  0 77 40000                   3         EAX     *0                  PUT Q INTO X
00002  0 35 00017                   4         STA     WK1
00003  0 76 00000                   5         LDA     0
00004  0 35 00006                   6         STA     EXIT                SAVE EXIT ADDR
00005  0 76 00017                   7         LDA     WK1
00006  0 67 10054                   8         NOD     44
00007  0 72 00265                   9         SKA     =040000000
I 00010  1 05 00000                10         DPN
00011  0 72 00265                  11         SKA     =040000000
00012  4 01 00014                  12         BRU     FIX,4               ARG EQU -1QO
00013  4 01 00016                  13         BRU     OK,4                ARG OK
00014  4 41 00015                  14  FIX    BRX     $+1,4               INCREMENT Q
00015  06624001                    15         DATA    06624001
I 00016  1 01 00010                16  OK     STD     AL
00017  4 76 00147                  17         LDA     MINUS,4
00020  0 53 00017                  18         SKN     WK1                 TEST ARG NEG
00021  4 76 00146                  19         LDA     PLUS,4
00022  0 35 00016                  20         STA     WKO                 INIT FIRST WORD
00023  4 76 00173                  21         LDA     3Q23,4              INIT WORD COUNT
00024  0 35 00022                  22         STA     COUNT               WORD COUNT EQU 4
00025  4 76 00145                  23         LDA     XMA,4               INIT STORE WORD
00026  4 35 00110                  24         STA     LOOP+2,4            4 WORDS TO BE STORED
00027  0 46 00200                  25         CXA                         INSPECT Q
00030  0 46 00500                  26         RCH     0500                EXPON OF A INTO X
00031  0 46 00200                  27         CXA
00032  4 73 00156                  28         SKG     44Q23,4             TEST Q GTR 44
00033  0 73 00266                  29         SKG     =0
00034  4 01 00055                  30         BRU     NEGQ,4
00035  4 54 00156                  31         SUB     44Q23,4
00036  0 46 01000                  32         CNA                         A EQU 44-Q
00037  0 35 00005                  33         STA     EXP
00040  0 46 00400                  34         CAX
00041  0 76 00011                  35         LDA     AH
00042  2 66 00000                  36         RSH     0,2                 SCALE ARG AT 44
I 00043  5 06 00175                37         DPS     10E13,4
00044  0 72 00265                  38         SKA     =040000000
```

```
00045  4 01 00047      39          BRU     ARG0K,4
00046  4 51 00144      40          BRR     0VFL0,4
00047  4 71 00153      41  ARG0K   LDX     M27Q23,4
I 00050  7 03 00232      42          DPA     9EX,6           FIND RANGE OF ARG
00051  4 41 00052      43          BRX     $+1,4
00052  0 72 00265      44          SKA     =040000000
00053  4 41 00050      45          BRX     ARG0K+1,4
00054  4 01 00067      46          BRU     SCALE,4
00055  0 46 01000      47  NEGQ    CNA
00056  4 73 00156      48          SKG     44Q23,4         TEST -Q GTR 44
00057  0 73 00267      49          SKG     =-1
00060  4 51 00144      50          BRR     0VFL0,4
00061  0 46 00400      51          CAX
I 00062  1 00 00010      52          LDP     AL
00063  2 66 00000      53          RSH     0,2             SCALE AT 0
00064  2 46 00000      54          RCH     0,2             CLEAR X
I 00065  1 01 00010      55          STD     AL
00066  0 37 00005      56          STX     EXP
00067  0 46 00200      57  SCALE   CXA                     A EQU -2E
00070  0 66 00001      58          RSH     1               A EQU -E
00071  0 46 01000      59          CNA                     A EQU E
00072  0 35 00004      60          STA     P0INT
00073  0 46 00441      61          RCH     0441            CAX,CXB,CLA
00074  6 54 00157      62          SUB     RSH,6           FORM RIGHT SHIFT
00075  0 63 00005      63          ADM     EXP
00076  0 46 00020      64          CBX
I 00077  1 00 00010      65          LDP     AL
I 00100  7 04 00263      66          DPM     10MX,6
00101  0 66 40005      67          RSH     *EXP
00102  4 71 00155      68          LDX     M2Q23,4         INIT 2 CHAR
00103  4 01 00106      69          BRU     L00P,4
00104  4 61 00110      70  W0RD    MIN     L00P+2,4
00105  4 71 00154      71          LDX     M4Q23,4         INIT 4 CHAR
00106  0 36 00010      72  L00P    STB     AL
00107  0 46 00004      73          CAB
00110  0 62 00016      74          XMA     WK0
00111  0 60 00004      75          SKR     P0INT           TEST FOR DECIMAL
00112  4 01 00121      76          BRU     DIGIT,4         FORM NEXT DIGIT
00113  4 75 00152      77          LDB     DECMAL,4        FORM DECIMAL POINT
```

```
00114  0 67 20006      78          LCY     6                   PACK DECIMAL POINT
00115  4 23 00110      79          EXU     LOOP+2,4
00116  0 36 00004      80          STB     POINT
00117  0 75 00010      81          LDB     AL
00120  4 01 00134      82          BRU     FLAGR,4
00121  0 67 20006      83  DIGIT   LCY     6                   PACK NEW DIGIT
00122  4 23 00110      84          EXU     LOOP+2,4
00123  0 75 00010      85          LDB     AL
00124  4 14 00150      86          ETR     MASK,4
00125  0 35 00011      87          STA     AH
00126  0 67 00002      88          LSH     2                   START DP ADD
00127  0 46 00014      89          XAB
00130  0 55 00010      90          ADD     AL
00131  0 46 00014      91          XAB
00132  0 57 00011      92          ADC     AH
00133  0 67 00001      93          LSH     1                   NEW DIGIT
00134  4 41 00106      94  FLAGR   BRX     LOOP,4              CHAR COUNT
00135  0 60 00022      95          SKR     COUNT               WORD COUNT
00136  4 01 00104      96          BRU     WORD,4
I      00137  1 00 00020      97          LDP     WK2
I      00140  1 01 00002      98          STD     2
I      00141  1 00 00016      99          LDP     WK0
00142  0 71 00007     100          LDX     TX
00143  0 51 00006     101          BRR     EXIT
00144  4 51 00141     102  OVFLO   BRR     $-3,4
00145  0 62 00016     103  XMA     XMA     WK0
00146  00001212        104  PLUS    DATA    01212
00147  00001240        105  MINUS   DATA    01240
00150  00777777        106  MASK    DATA    0777777
00151  17777777        107  XMASK   DATA    017777777
00152  33100000        108  DECMAL  DATA    033100000
00153  77777745        109  M27Q23  DATA    -27
00154  77777774        110  M4Q23   DATA    -4
00155  77777776        111  M2Q23   DATA    -2
00156  00000054        112  44Q23   DATA    44
00157  77777777        113  RSH     DATA    -1
00160  00000050        114          DATA    40
00161  00000044        115          DATA    36
00162  00000041        116          DATA    33
```

```
00163  00000036        117             DATA    30
00164  00000032        118             DATA    26
00165  00000027        119             DATA    23
00166  00000024        120             DATA    20
00167  00000020        121             DATA    16
00170  00000015        122             DATA    13
00171  00000012        123             DATA    10
00172  00000006        124             DATA    6
00173  00000003        125  3G23       DATA    3
00174  00000000        126             DATA    0
00175  45200000        127  1CE13      DATA    045200000
00176  22141163        128             DATA    022141163
00177  33100000        129             DATA    033100000
00200  20275716        130             DATA    020275716
00201  34240000        131             DATA    034240000
00202  01506141        132             DATA    01506141
00203  26020000        133             DATA    026020000
00204  00123643        134             DATA    0123643
00205  42150000        135             DATA    042150000
00206  00010303        136             DATA    010303
00207  11644000        137             DATA    011644000
00210  00000655        138             DATA    0655
00211  72452000        139             DATA    072452000
00212  00000052        140             DATA    052
00213  22521000        141             DATA    022521000
00214  00000004        142             DATA    04
00215  33356400        143             DATA    033356400
00216  00000000        144             DATA    0
00217  02576200        145             DATA    02576200
00220  00000000        146             DATA    0
00221  00214500        147             DATA    0214500
00222  00000000        148             DATA    0
00223  00016040        149             DATA    016040
00224  00000000        150             DATA    0
00225  00001320        151             DATA    01320
00226  00000000        152             DATA    0
00227  00000110        153             DATA    0110
00230  00000000        154             DATA    0
00231  01133660        155             DATA    01133660
```

```
00232  21457146        156  9EX     DATA    021457146
00233  41362640        157          DATA    041362640
00234  25772777        158          DATA    025772777
00235  31657400        159          DATA    031657400
00236  33371577        160          DATA    033371577
00237  50115540        161          DATA    050115540
00240  21134057        162          DATA    021134057
00241  42141100        163          DATA    042141100
00242  25363073        164          DATA    025363073
00243  32571300        165          DATA    032571300
00244  32657712        166          DATA    032657712
00245  40553674        167          DATA    040553674
00246  20615736        168          DATA    020615736
00247  10706660        169          DATA    010706660
00250  24761326        170          DATA    024761326
00251  53070420        171          DATA    053070420
00252  32155613        172          DATA    032155613
00253  22743260        173          DATA    022743260
00254  20304467        174          DATA    020304467
00255  07534140        175          DATA    07534140
00256  24365605        176          DATA    024365605
00257  31463160        177          DATA    031463160
00260  31463146        178          DATA    031463146
00261  00000000        179          DATA    0
00262  20000000        180          DATA    020000000
00263  00000000        181  10MX    DATA    0
00264  24000000        182          DATA    024000000
       00000004        183  POINT   EQU     04
       00000005        184  EXP     EQU     05
       00000006        185  EXIT    EQU     06
       00000007        186  TX      EQU     07
       00000010        187  AL      EQU     010
       00000011        188  AH      EQU     011
       00000016        189  WK0     EQU     016
       00000017        190  WK1     EQU     017
       00000020        191  WK2     EQU     020
       00000021        192  WK3     EQU     021
       00000022        193  COUNT   EQU     022
00265  40000000        194          END
00266  00000000
00267  77777777
```

IDENTIFICATION: Binary to Decimal Conversion, Single, Floating - BFS

AUTHOR: Richard S. Resnick, SDS

ACCEPTED: 15 January 1963

COMPUTER CONFIGURATION: Any SDS 920/930 Computer

PURPOSE: To convert a single precision floating point binary number with the mantissa in A and exponent in B to a formatted scientific notation six digit decimal number with decimal exponent into B, A and Location 2. The format symbols and digits are in BCD 6 bits per character form.

Twelve characters (3 words) are necessary to represent the final converted number completely formatted for output.

PROGRAMMED OPERATORS: EXP

STORAGE: Instructions plus constants: 113 oct, 75 dec

Uses temporary storage locations 17 thru 21 and location 2.

TIMING: 2.824 milliseconds.

SOURCE LANGUAGE: SYMBOL

USE:

1. FUNCTION

The final twelve character converted decimal number with format symbols replaces the argument in A, B in the following manner:

± . X X	X X X X	± X X
B	A	Location 2

The address field is not used.

2. ARGUMENT

The argument is a single precision floating point number with Mantissa in A and exponent in B. On exit, the 6 bit

USE: (Cont.) characters are in the designated registers in the following manner:

A = 3rd, 4th, 5th and 6th digits of the mantissa.

B = Sign, decimal point, 1st and 2nd digits of the mantissa.

Location 2 = space, sign and 2 digits of the exponent.

3. ERROR

If the argument exceeds 10^{100} in absolute value, overflow will be set and the result will be erroneous.

METHOD: The exponent of the argument is converted to a decimal exponent by extracting the integer portion of the product $Q \log_{10} 2 = I \pm F$. The fractional portion represents the none binary portion of the number that is converted by finding 10^{F} (expoential program).

Multiplying the result times the mantissa completes the arithmetic part of the conversion.

Extracting the actual decimal digits is done by multiplying by 10^{-I} and adding to from the polynomial:

$$\sum_{0}^{5} A_i \, 10^{-i}$$

The results are formatted as previously mentioned. The max error is 10^{-6}.

```
                                  1  $BFS   POPD    017100000
00000  0 37 00020                 2  BFS    STX     TX                SAVE X
00001  0 35 00021                 3         STA     TM                SAVE SIGN OF MANT.
00002  0 76 00000                 4         LDA     0                 SAVE THE EXIT
00003  4 35 00075                 5         STA     EXIT,4
00004  2 46 00010                 6         CBA     ,2                CBA + CLX
00005  4 64 00077                 7         MUL     MC,4              FORM I + F
00006  0 35 00002                 8         STA     TI                SAVE I
00007  0 46 10012                 9         BAC
00010  0 66 20001                10         RCY     1                 SCALE F AT 0
00011  1 01 00002                11         EXP     2                 ENTER 10**F
00012  0 46 00014                12         XAB
00013  0 66 20006                13         RCY     6                 SCALE 10**F AT 5
00014  0 64 00021                14         MUL     TM                D = B*10**F
00015  0 72 00112                15         SKA     =040000000
00016  4 01 00021                16         BRU     $+3,4             YES
00017  4 75 00102                17         LDB     BC,4              NO 00 SP. INTO B
00020  4 01 00023                18         BRU     $+3,4
00021  0 46 01000                19         CNA                       NEGATE A
00022  4 75 00103                20         LDB     BC+1,4            00-. INTO B
00023  4 55 00110                21         ADD     RCM,4             ROUND
00024  4 72 00076                22         SKA     EC,4              TEST 1ST WORD ZERO
00025  4 01 00033                23         BRU     $+6,4             NOT ZERO
00026  0 36 00021                24         STB     TM                SAVE B
00027  0 67 20005                25         LCY     5                 SET UP NEXT CHAR
00030  4 64 00100                26         MUL     MC+1,4            10 AT 5
00031  0 75 00021                27         LDB     TM                RESTORE B
00032  4 01 00034                28         BRU     $+2,4             DONT INCREMENT I
00033  0 61 00002                29         MIN     TI                INCREMENT I
00034  0 67 20006                30         LCY     6                 PUT 1ST CHAR INTO B
00035  4 43 00064                31         BRM     CBD,4             ENTER BIN TO DEC
00036  0 36 00017                32         STB     TB                B REGISTER RESULT
00037  0 46 00002                33         CLB
00040  4 71 00106                34         LDX     XC,4              -4 AT 23 4 CHAR
00041  4 43 00064                35         BRM     CBD,4             ENTER BIN TO DEC
00042  0 46 10012                36         BAC
00043  0 62 00002                37         XMA     TI                A REGISTER RESULT
00044  4 65 00101                38         DIV     DC,4              100 AT 23
```

```
00045  0 72 00112      39          SKA     =040000000
00046  4 01 00051      40          BRU     $+3,4           YES
00047  4 75 00104      41          LDB     BC+2,4          NO 00 SPSP INTO B
00050  4 01 00053      42          BRU     $+3,4
00051  0 46 01000      43          CNA                     NEGATE A
00052  4 75 00105      44          LDB     BC+3,4          OOSP- INTO B
00053  4 55 00111      45          ADD     RCE,4           ROUND EXPONENT
00054  0 67 20001      46          LCY     1               SCALE EXPON AT -1
00055  4 71 00107      47          LDX     XC+1,4          -2 AT 23 2 CHAR
00056  4 43 00064      48          BRM     CBD,4           ENTER BIN TO DEC
00057  0 46 10012      49          BAC
00060  0 62 00002      50          XMA     TI              EXTENDED REG. RESULT
00061  0 75 00017      51          LDB     TB
00062  0 71 00020      52          LDX     TX              RESTORE X
00063  4 51 00075      53          BRR     EXIT,4
00064  0 00 00000      54  CBD     PZE                     BIN TO DEC CONVERT
00065  0 36 00021      55          STB     TM              SAVE B
00066  0 46 00002      56          CLB
00067  0 66 20001      57          RCY     1               SCALE RESULTS AT 0
00070  4 64 00100      58          MUL     MC+1,4          10 AT 4 DEC CHAR AT 5
00071  0 75 00021      59          LDB     TM              RESTORE B
00072  0 67 20006      60          LCY     6               NEXT CHAR INTO B
00073  4 41 00065      61          BRX     $-6,4           TEST FINISH
00074  4 51 00064      62          BRR     CBD,4
00075  0 00 00000      63  EXIT    PZE
00076  77000000        64  EC      DATA    077000000
00077  11504047        65  MC      DATA    011504047
00100  12000000        66          DATA    012000000
00101  00000144        67  DC      DATA    0144
00102  00001233        68  BC      DATA    01233
00103  00004033        69          DATA    04033
00104  00000505        70          DATA    0505
00105  00000520        71          DATA    0520
00106  77777774        72  XC      DATA    077777774
00107  77777776        73          DATA    077777776
00110  00000001        74  RCM     DATA    01
00111  00000010        75  RCE     DATA    010
       00000002        76  TI      EQU     02              EXT REG
       00000021        77  TM      EQU     021
       00000020        78  TX      EQU     020
       00000017        79  TB      EQU     017
                       80          END
00112  40000000
```

IDENTIFICATION: Binary to Decimal Conversion, Double-Precision Floating Point - BDF

AUTHOR: W. LaSor, F. Valadez, SDS

ACCEPTED: 25 April 1963

COMPUTER CONFIGURATION: Any SDS 920/930 Computer

PURPOSE: To convert a double-precision floating point binary number to an 11-digit BCD number formatted in scientific notation.

PROGRAMMED OPERATORS: DPM, DPN

STORAGE: 253 oct, 171 dec

Uses temporary storage 16, 20, 21.

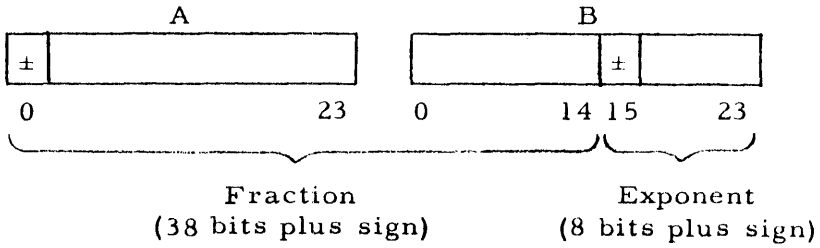
TIMING: (12.9 + .65 E) m.s., where E = decimal exponent.

SOURCE LANGUAGE: SYMBOL

USE:

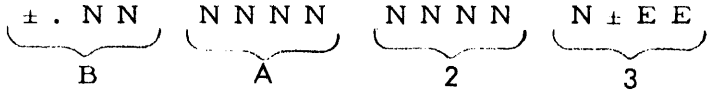
1. INPUT

The argument for the BDF program is a double-precision floating point number in the combined A, B registers as shown below:



2. OUTPUT

The subroutine exits with the BCD number in A, B and memory locations 00002 and 00003 arranged as shown below:



USE: (cont.) Where N represents the digits of the mantissa and E represents the digits of the exponent.

Example:

Upon exit from the subroutine, the number +. 12345678901 with a decimal exponent of 15 would appear as follows:

+. 12	3456	7890	1 + 15
B	A	2	3

3. ACCURACY

The maximum relative error (E_r) observed is 10^{-11}, where:

$$E_r = \left| \frac{N^* - N}{N} \right|$$

N^* = computed value

N = true value

METHOD: The decimal exponent is initially set to 0. The binary exponent is extracted and its magnitude is tested. If the binary exponent is greater than 0, the fractional portion of the argument is divided by 10, the binary exponent adjusted, and the decimal exponent incremented by one. If the binary exponent is less than -3, the fraction is multiplied by 10, the binary exponent is adjusted, and the decimal exponent decremented by one. This process continues until the binary exponent lies within the range $-3 \leqslant BE \leqslant 0$.

The resulting fraction is first rounded, then converted to decimal by successive multiplications by 10 until 11 digits have been generated. As each digit is generated, it is merged, along with the sign and decimal point, into the four-word output format.

```
                                  1  $BDF   POPD   017000000
00000  4 37 00246                 2  BDF    STX    TX,4
00001  4 71 00242                 3         LDX    M4,4
00002  4 37 00212                 4         STX    X2,4
00003  4 37 00213                 5         STX    X3,4
00004  4 71 00237                 6         LDX    PTEN,4
00005  0 37 00016                 7         STX    COUNT          SET DIGIT COUNT
00006  0 71 00000                 8         LDX    0
00007  4 37 00244                 9         STX    RET,4          SAVE RETURN
00010  0 46 00122                10         STE                   EXTRACT EXPONENT
00011  0 72 00252                11         SKA    SIGN           TEST SIGN
00012  4 01 00113                12         BRU    NEG,4
00013  0 73 00247                13         SKG    ZERO
00014  0 71 00251                14         LDX    ONE
00015  4 62 00245                15         XMA    OUT,4
00016  4 76 00223                16         LDA    SPACE,4
00017  4 62 00245                17         XMA    OUT,4
00020  4 43 00167                18         BRM    FORM,4
00021  4 37 00243                19  SET    STX    BE,4           SET BE
00022  0 35 00022                20         STA    ARG+1
00023  0 46 00001                21         CLA
00024  0 35 00020                22         STA    DE             SET DE TO ZERO
00025  4 76 00224                23         LDA    POINT,4
00026  4 35 00245                24         STA    OUT,4
00027  4 43 00167                25         BRM    FORM,4
00030  4 76 00243                26         LDA    BE,4           TEST FOR RANGE
00031  0 73 00247                27  LOOP   SKG    ZERO
00032  4 73 00242                28         SKG    M4,4
00033  4 01 00126                29         BRU    REDUCE,4       OUT OF RANGE
00034  0 46 01000                30         CNA                   IN RANGE
00035  0 46 00400                31         CAX
00036  0 76 00022                32         LDA    ARG+1
00037  2 66 00000                33         RSH    0,2            SCALE AT 0
00040  0 35 00022                34         STA    ARG+1          SAVE SCALED ARG
00041  0 36 00021                35         STB    ARG
I      00042  5 01 00235         36         DPM    TEN,4
00043  4 72 00230                37         SKA    MASK2,4        TEST FOR LEADING ZERO
00044  4 01 00164                38         BRU    LOAD,4
```

```
00045  0 67 00005      39          LSH     5
00046  0 60 00020      40          SKR     DE                  FIX EXPONENT
00047  0 20 00000      41          NOP
00050  0 46 00014      42  ROUND   XAB
00051  4 55 00225      43          ADD     RNDF,4
00052  0 46 00014      44          XAB
00053  0 57 00247      45          ADC     ZERO
00054  0 40 20001      46          OVT                         TEST ROUND OVFLO
00055  4 01 00160      47          BRU     FIX,4
00056  5 01 00235      48          DPM     TEN,4
00057  4 35 00245      49  PR      STA     OUT,4
00060  4 43 00167      50          BRM     FORM,4
00061  4 14 00227      51          ETR     MASK1,4
00062  0 67 00005      52          LSH     5
00063  0 60 00016      53          SKR     COUNT               SKIP IF DONE
00064  4 01 00056      54          BRU     PR-1,4
00065  0 76 00020      55          LDA     DE                  PROCESS EXPONENT
00066  0 72 00252      56          SKA     SIGN
00067  4 01 00153      57          BRU     MINUS,4
00070  4 75 00223      58          LDB     SPACE,4
00071  4 36 00245      59          STB     OUT,4
00072  4 43 00167      60          BRM     FORM,4
00073  0 46 00002      61  CONT    CLB
00074  0 66 00027      62          RSH     23                  SCALE AT 46
00075  4 65 00237      63          DIV     PTEN,4
00076  0 67 00022      64          LSH     18
00077  4 35 00245      65          STA     OUT,4
00100  4 43 00167      66          BRM     FORM,4
00101  4 36 00245      67          STB     OUT,4
00102  4 43 00167      68          BRM     FORM,4
00103  4 76 00222      69          LDA     TAB+3,4             SET OP WORD
00104  0 35 00003      70          STA     3
00105  4 76 00221      71          LDA     TAB+2,4
00106  0 35 00002      72          STA     2
00107  4 76 00220      73          LDA     TAB+1,4
00110  4 75 00217      74          LDB     TAB,4
00111  4 71 00246      75          LDX     TX,4                RESTORE INDEX AND EXIT
00112  4 51 00244      76          BRR     RET,4
00113  0 66 00001      77  NEG     RSH     1
```

```
00114  4 41 00115      78          BRX     $+1,4              CORRECT EXPONENT
00115  4 37 00243      79          STX     BE,4
I      00116  1 02 00000      80          DPN
00117  4 71 00243      81          LDX     BE,4
00120  0 67 10002      82          NOD     2
00121  4 62 00245      83          XMA     OUT,4
00122  0 76 00252      84          LDA     SIGN
00123  4 62 00245      85          XMA     OUT,4
00124  4 43 00167      86          BRM     FORM,4
00125  4 01 00021      87          BRU     SET,4
00126  0 72 00252      88  REDUCE  SKA     SIGN
00127  4 01 00141      89          BRU     MLT,4
00130  4 54 00232      90          SUB     P3,4
00131  0 46 00400      91          CAX
00132  0 76 00022      92          LDA     ARG+1
I      00133  5 01 00240      93          DPM     TENTH,4
00134  0 67 10001      94          NOD     1
00135  0 35 00022      95          STA     ARG+1
00136  0 46 00200      96          CXA
00137  0 61 00020      97          MIN     DE
00140  4 01 00031      98          BRU     LOOP,4
00141  4 54 00242      99  MLT     SUB     M4,4
00142  0 60 00020     100          SKP     DE
00143  0 20 00000     101          NOP
00144  0 46 00400     102          CAX
00145  0 76 00022     103          LDA     ARG+1
I      00146  5 01 00233     104          DPM     P10,4
00147  0 67 10001     105          NOD     1
00150  0 35 00022     106          STA     ARG+1
00151  0 46 00200     107          CXA
00152  4 01 00031     108          BRU     LOOP,4
00153  0 75 00252     109  MINUS   LDB     SIGN
00154  4 36 00245     110          STB     OUT,4
00155  4 43 00167     111          BRM     FORM,4
00156  0 46 01000     112          CNA
00157  4 01 00073     113          BRU     CONT,4
00160  0 61 00020     114  FIX     MIN     DE                 ADJUST EXPONENT
00161  4 76 00231     115          LDA     P1,4
00162  0 46 00002     116          CLR
```

```
00163  4 01 00057      117            BRU     PR,4
00164  0 76 00022      118  LOAD      LDA     ARG+1
00165  0 75 00021      119            LDB     ARG
00166  4 01 00050      120            BRU     ROUND,4          GO ROUND
00167  0 00 00000      121  FORM      PZE
00170  4 35 00214      122            STA     SAVEA,4          SAVE REGISTERS
00171  4 36 00215      123            STB     SAVEB,4
00172  4 37 00216      124            STX     SAVEX,4
00173  4 75 00245      125            LDB     OUT,4
00174  4 71 00213      126            LDX     X3,4
00175  6 76 00223      127            LDA     TAB+4,6
00176  0 67 00006      128            LSH     6
00177  6 35 00223      129            STA     TAB+4,6
00200  4 37 00213      130            STX     X3,4
00201  4 71 00212      131            LDX     X2,4
00202  4 41 00205      132            BRX     $+3,4
00203  4 61 00213      133            MIN     X3,4             INCREMENT WORD
00204  4 71 00242      134            LDX     M4,4
00205  4 37 00212      135            STX     X2,4
00206  4 76 00214      136            LDA     SAVEA,4          RESTORE REGISTERS
00207  4 75 00215      137            LDB     SAVEB,4
00210  4 71 00216      138            LDX     SAVEX,4
00211  4 51 00167      139            BRR     FORM,4
00212  0 00 00000      140  X2        PZE
00213  0 00 00000      141  X3        PZE
00214  0 00 00000      142  SAVEA     PZE
00215  0 00 00000      143  SAVEB     PZE
00216  0 00 00000      144  SAVEX     PZE
00217                  145  TAB       BSS     4
00223  12000000        146  SPACE     DATA    012000000
00224  33000000        147  POINT     DATA    033000000
00225  00001300        148  RNDF      DATA    01300
00226  77763500        149  CORR      DATA    077763500
00227  00777777        150  MASK1     DATA    0777777
00230  77000000        151  MASK2     DATA    077000000
00231  01000000        152  P1        DATA    1*/(23-5)
00232  00000003        153  P3        DATA    3
00233  00000000        154  P10       DATA    0.024000000
00234  24000000
```

```
00235  00000000          155  TEN     DATA    0.012000000
00236  12000000
00237  00000012          156  PTEN    DATA    10
00240  31463150          157  TENTH   DED     .1*/(47+3)
00241  31463146
00242  77777774          158  M4      DATA    -4
00243  0 00 00000        159  BE      PZE
00244  0 00 00000        160  RET     PZE
00245  0 00 00000        161  OUT     PZE
00246  0 00 00000        162  TX      PZE
       00000016          163  COUNT   EQU     016
       00000020          164  DE      EQU     020
       00000021          165  ARG     EQU     021
00247  00000000          166  ZERO    DATA    0
00250  77777777          167  ONES    DATA    -1
00251  00000001          168  ONE     DATA    01
00252  40000000          169  SIGN    DATA    040000000
                         170          END
```

IDENTIFICATION: Decimal to Binary Conversion, Double, Fixed - DBD

AUTHOR: Richard S. Resnick, SDS

ACCEPTED: 26 April 1963

COMPUTER CONFIGURATION: Any SDS 920/930 Computer

PURPOSE: To convert a thirteen digit decimal number with space, sign and decimal point from sixteen 6-bit BCD characters in A, B, Location 2 and Location 3 to a fixed point double precision binary number in A, B.

PROGRAMMED OPERATORS: DPN, DPM, DPA

STORAGE: Instructions and constants: 262 oct, 178 dec

Uses temporary storage locations 16 thru 22, and 5 thru 11.

TIMING: 6.096 to 6.448 milliseconds

SOURCE LANGUAGE: SYMBOL

USE:

1. FUNCTION

To convert a thirteen digit decimal number with space, sign and decimal point from sixteen 6-bit BCD characters in A, B, Location 2 and Location 3 to a fixed point double precision binary number in A, B at a Q (binary point location) designated by the address field of the DBD programmed operator.

2. ARGUMENT

The argument is sixteen 6-bit BCD characters in A, B, Location 2 and Location 3. Thirteen of these characters are numbers, one character a space, one character a sign and one character a decimal point. The space is the first character and is followed by a sign and fourteen more characters of which one is a decimal point. The

USE: (cont.) characters must be arranged in A, B, Location 2 and Location 3 in the following manner:

Sp ± X X	X X X X	X X X X	X X X X
B	A	2	3

The fourteen X's represent thirteen decimal digits and one decimal point.

The address field of the DBD programmed operator determines the binary scaling of the resultant binary number.

3. ACCURACY

The resultant binary number is accurate to 44 bits.

4. ERROR

Overflow is set when the converted argument is too large to fix at the desired Q.

METHOD: The argument is converted to a double precision number by individually converting the contents of B, A, Location 2 and Location 3 to fixed point single precision integers scaled at 23. A flag is set when the decimal occurs in one of the four words, then the converted word in B is multiplied by 10^4 or 10^3 depending on whether the decimal point was in A, and then added to A. Similarly, the converted word in Location 2 is multiplied and added to the word in Location 3. The result of the B, A calculation is then multiplied by 10^7 or 10^8 depending on the decimal in Location 2 or Location 3, and added to the Location 2, Location 3 calculation. The result is a double precision integer which is multiplied by 10^{-i} where i is the number of digits to the left of the decimal point. The resultant number is then scaled to the desired binary location by shifting.

```
                                1  $DBD    P0PD    017100000
00000  0 37 00005               2          STX     TX
00001  0 71 00000               3          LDX     0
00002  0 37 00006               4          STX     EXIT
00003  0 35 00007               5          STA     WK1
00004  0 46 00021               6          RCH     021                 CBX,CLA
00005  0 35 00010               7          STA     DECMAL
00006  4 52 00174               8          SKB     1Q6,4               TEST NEG ARG
00007  0 76 00261               9          LDA     =040000000
00010  0 35 00011              10          STA     SIGN
00011  4 43 00141              11          BRM     CONVRT,4            CONVERT 1ST WORD
00012  0 02 20001              12          EOM     020001              TURN OFF OVFLO
00013  0 35 00017              13          STA     SUM+1
00014  0 71 00007              14          LDX     WK1
00015  4 43 00141              15          BRM     CONVRT,4            CONVERT 2ND WORD
00016  0 62 00017              16          XMA     SUM+1
00017  0 67 00016              17          LSH     14
00020  0 40 20001              18          SKS     020001              TEST FOR DECIMAL
00021  4 01 00024              19          BRU     $+3,4
00022  4 64 00201              20          MUL     10KQ14,4            10K*WORD1
00023  4 01 00025              21          BRU     $+2,4
00024  4 64 00200              22          MUL     1KQ14,4             1K*WORD1
00025  0 63 00017              23          ADM     SUM+1               10K*WORD1 + WORD2
00026  0 71 00002              24          LDX     2                   WORD 3
00027  4 43 00141              25          BRM     CONVRT,4            CONVERT 3RD WORD
00030  4 43 00170              26          BRM     Q,4                 OVFLO FLAG
00031  0 35 00016              27          STA     SUM
00032  0 02 20001              28          EOM     020001              TURN OFF OVFLO
00033  0 71 00003              29          LDX     3                   WORD4
00034  4 43 00141              30          BRM     CONVRT,4            CONVERT 4TH WORD
00035  4 75 00170              31          LDB     Q,4
00036  4 43 00170              32          BRM     Q,4                 OVFLO FLAG
00037  0 46 00014              33          XAB
00040  4 16 00170              34          MRG     Q,4
00041  4 35 00170              35          STA     Q,4
00042  0 46 00014              36          XAB
00043  0 62 00016              37          XMA     SUM
00044  0 40 20001              38          SKS     020001              TEST FOR DECIMAL
```

```
00045  4 01 00050      39          BRU     $+3,4
00046  4 64 00203      40          MUL     10KQ24,4                10K*W3RD3
00047  4 01 00051      41          BRU     $+2,4
00050  4 64 00202      42          MUL     1KQ24,4                 1K*W0RD3
00051  0 46 00014      43          XAB
00052  0 55 00016      44          ADD     SUM
00053  0 35 00016      45          STA     SUM
00054  0 46 00014      46          XAB
00055  0 57 00262      47          ADC     =0
00056  0 62 00017      48          XMA     SUM+1
00057  0 46 00002      49          CLB
00060  0 67 00003      50          LSH     3
00061  4 53 00170      51          SKN     Q,4                     TEST FOR DECIMAL
00062  4 01 00065      52          BRU     $+3,4
00063  4 64 00204      53          MUL     10M,4                   10M*SUM+1
00064  4 01 00066      54          BRU     $+2,4
00065  4 64 00205      55          MUL     100M,4                  100M*SUM+1
I      00066  1 01 00016      56          DPA     SUM                     100M*SUM+1 + SUM
00067  0 71 00020      57          LDX     EXP
00070  6 71 00206      58          LDX     SCALE,6                 Q OF 10**-I + 47
00071  0 67 10057      59          NOD     47
00072  4 37 00170      60          STX     Q,4
00073  0 62 00020      61          XMA     EXP
00074  0 46 00022      62          RCH     022                     CBX,CLB
00075  0 67 00001      63          LSH     1
00076  0 46 00440      64          RCH     0440                    CAX,CXB
00077  0 76 00020      65          LDA     EXP
I      00100  7 02 00225      66          DPM     10MI,6
00101  0 46 00014      67          XAB
00102  4 14 00172      68          ETR     MASK,4                  77777770
00103  0 46 00014      69          XAB
00104  0 53 00011      70          SKN     SIGN
00105  4 01 00107      71          BRU     $+2,4
I      00106  1 03 00000      72          DPN                             NEGATE
00107  0 71 00005      73          LDX     TX
00110  0 77 40006      74          EAX     *EXIT
00111  0 46 00160      75          XEE
00112  0 46 00160      76          XEE
00113  4 35 00141      77          STA     CONVRT,4
```

```
00114  0 46 00200      78          CXA
00115  4 54 00170      79          SUB     Q,4
00116  4 71 00141      80          LDX     CONVRT,4
00117  0 72 00025      81          SKA     21          TEST NEG SCALING
00120  4 01 00127      82          BRU     LSH,4
00121  4 73 00175      83          SKG     47Q23,4
00122  4 01 00124      84          BRU     $+2,4
00123  4 01 00133      85          BRU     CLR,4
00124  0 46 00600      86          XXA
00125  2 66 00000      87          RSH     0,2
00126  4 01 00136      88          BRU     LEAVE,4
00127  0 46 01000      89  LSH     CNA
00130  4 73 00175      90          SKG     47Q23,4
00131  4 01 00134      91          BRU     $+3,4
00132  4 51 00140      92          BRR     OVFLO,4
00133  2 46 00003      93  CLR     RCH     03,2
00134  0 46 00600      94          XXA
00135  2 67 00000      95          LSH     0,2
00136  0 71 00005      96  LEAVE   LDX     TX
00137  0 51 00006      97          BRR     EXIT
00140  4 51 00135      98  OVFLO   BRR     $-3,4
00141  0 00 00000      99  CONVRT  PZE                 DEC TO BIN CONVERSION
00142  0 46 30003     100          CLR
00143  4 76 00177     101          LDA     3Q23,4
00144  0 35 00021     102          STA     FLAG
00145  0 36 00022     103          STB     WORD
00146  0 46 00041     104  LOOP    RCH     041         CXB,CLA
00147  0 67 00006     105          LSH     6
00150  0 46 00020     106          CBX
00151  4 73 00176     107          SKG     9Q23,4      TEST A GTR 9
00152  4 01 00157     108          BRU     DIGIT,4
00153  0 76 00010     109          LDA     DECMAL
00154  0 35 00020     110          STA     EXP
00155  4 51 00156     111          BRR     SET,4
00156  4 51 00163     112  SET     BRR     COUNT,4
00157  0 61 00010     113  DIGIT   MIN     DECMAL
00160  0 62 00022     114          XMA     WORD
00161  4 64 00173     115          MUL     10Q4,4
00162  0 67 00004     116          LSH     4
```

```
00163  0 63 00022      117  COUNT  ADM     WORD
00164  0 60 00021      118         SKR     FLAG
00165  4 01 00146      119         BRU     LOOP,4
00166  0 76 00022      120         LDA     WORD
00167  4 51 00141      121         BRR     CONVRT,4
00170  0 00 00000      122  Q      PZE                          SAVE OVFLO
00171  4 51 00170      123         BRR     Q,4                  RETURN
00172  77777770        124  MASK   DATA    077777770
00173  24000000        125  10Q4   DATA    024000000
00174  00400000        126  1Q6    DATA    0400000
00175  00000057        127  47Q23  DATA    47
00176  00000011        128  9Q23   DATA    9
00177  00000003        129  3Q23   DATA    3
00200  01750000        130  1KQ14  DATA    1000*/(23-14)
00201  23420000        131  10KQ14 DATA    10000*/(23-14)
00202  00000764        132  1KQ24  DATA    1000*/(23-24)
00203  00011610        133  10KQ24 DATA    10000*/(23-24)
00204  02304550        134  10M    DATA    02304550
00205  27657020        135  10OM   DATA    027657020
00206  00000004        136  SCALE  DATA    4
00207  00000010        137         DATA    8
00210  00000013        138         DATA    11
00211  00000016        139         DATA    14
00212  00000022        140         DATA    18
00213  00000025        141         DATA    21
00214  00000030        142         DATA    24
00215  00000034        143         DATA    28
00216  00000037        144         DATA    31
00217  00000042        145         DATA    34
00220  00000046        146         DATA    38
00221  00000051        147         DATA    41
00222  00000054        148         DATA    44
00223  00000054        149         DATA    44
00224  00000060        150         DATA    48
00225  50222740        151  10MI   DATA    050222740
00226  34113411        152         DATA    034113411
00227  01133660        153         DATA    01133660
00230  21457146        154         DATA    021457146
00231  41362640        155         DATA    041362640
```

```
00232  25772777          156            DATA    025772777
00233  31657400          157            DATA    031657400
00234  33371577          158            DATA    033371577
00235  50115540          159            DATA    050115540
00236  21134057          160            DATA    021134057
00237  42141100          161            DATA    042141100
00240  25363073          162            DATA    025363073
00241  32571300          163            DATA    032571300
00242  32657712          164            DATA    032657712
00243  40553674          165            DATA    040553674
00244  20615736          166            DATA    020615736
00245  10706660          167            DATA    010706660
00246  24761326          168            DATA    024761326
00247  53070420          169            DATA    053070420
00250  32155613          170            DATA    032155613
00251  22743260          171            DATA    022743260
00252  20304467          172            DATA    020304467
00253  07534140          173            DATA    07534140
00254  24365605          174            DATA    024365605
00255  31463160          175            DATA    031463160
00256  31463146          176            DATA    031463146
00257  00000000          177            DATA    0
00260  20000000          178            DATA    020000000
       00000005          179  TX        EQU     05
       00000006          180  EXIT      EQU     06
       00000007          181  WK1       EQU     07
       00000010          182  DECMAL    EQU     010
       00000011          183  SIGN      EQU     011
       00000016          184  SUM       EQU     016
       00000020          185  EXP       EQU     020
       00000021          186  FLAG      EQU     021
       00000022          187  WORD      EQU     022
                         188            END
00261  40000000
00262  00000000
```

IDENTIFICATION: Decimal to Binary Conversion, Single, Floating - DFS

AUTHOR: Richard S. Resnick, SDS

ACCEPTED: 16 January 1963

COMPUTER CONFIGURATION: Any SDS 920/930 Computer

PURPOSE: To convert a formatted scientific notation six digit decimal number with decimal exponent* into a single precision floating point number with a binary mantissa in A and a binary exponent in B.

PROGRAMMED OPERATORS: EXP

STORAGE: Instructions plus constants: 64 oct, 52 dec

Uses temporary storage locations 16 thru 21 and location 2.

TIMING: 2.312 milliseconds

SOURCE LANGUAGE: SYMBOL

USE:

1. FUNCTION

The argument in A, B and Location 2 is replaced by a converted single precision floating point binary number with mantissa in A and Q or exponent in B.

2. ARGUMENT

The argument is a decimal number using scientific notation of the form ±. XXXXXX with exponent interpreted as 10^X in the form ±XX where X ranges from +77 to -77. Each character (symbol or digit) is in BCD form 6 bits per character arranged on entry in the following manner:

± . X X	X X X X	± X X
B	A	Location 2

*The symbols and digits are in BCD 6 bits per character (12 characters in all) in A and B and location 2.

USE: (Cont.)

B = Sign, decimal point, 1st and 2nd decimal digits.

A = 3rd, 4th, 5th and 6th decimal digits.

A and B refer to the decimal mantissa.

Location 2 = Space, sign and 2 decimal digits.

Location 2 refers to the decimal exponent.

METHOD:

The mantissa of the argument (Md) is converted to a binary fraction by the equation:

$$Md = \frac{\sum_{0}^{5} Di \cdot 10^{i}}{10^{6}}$$

where the decimal number is $\pm D_5 D_4 D_3 D_2 D_1 D_0$.

The exponent is converted in a similar manner and a binary exponent (Q) formed by the equation:

$$Q + F = X \log_2 10$$

where X = converted decimal exponent and F is the fractional portion of the converted binary exponent. The final binary mantissa (Mb) is formed by the equation:

$$Mb = Md\ 2^{F}$$

The results are normalized and formed as previously mentioned. The max error is 2^{-20}.

```
                                    1  $DFS    POPD    017100000
00000  0 37 00021                   2  DFS     STX     TX                  SAVE X
00001  0 71 00000                   3          LDX     0                   SAVE EXIT
00002  4 37 00053                   4          STX     EXIT,4
00003  0 46 00014                   5          XAB
00004  0 35 00016                   6          STA     TS                  SAVE SIGN
00005  4 14 00054                   7          ETR     EC,4                EXTRACT OFF 2 CHAR
00006  4 71 00057                   8          LDX     XC,4                -5 AT 23
00007  4 01 00012                   9          BRU     $+3,4
00010  0 75 00017                  10          LDB     TB                  BEGIN DEC TO BIN
00011  0 67 20006                  11          LCY     6
00012  0 35 00020                  12          STA     TA
00013  0 36 00017                  13          STB     TB
00014  4 14 00056                  14          ETR     EC+2,4              77777700
00015  4 64 00060                  15          MUL     MC,4                -54/64 AT 0
00016  0 55 00020                  16          ADD     TA
00017  4 41 00010                  17          BRX     $-7,4
00020  4 65 00062                  18          DIV     DC,4                10**6 AT 23
00021  0 53 00016                  19          SKN     TS                  TEST NEGATIVE
00022  4 01 00024                  20          BRU     $+2,4
00023  0 46 01000                  21          CNA                         NEGATE A
00024  0 62 00002                  22          XMA     TI                  STORE A, INT INTO A
00025  4 55 00063                  23          ADD     CON,4               EXTEND SIGN OF INT
00026  0 35 00020                  24          STA     TA                  SAVE RESULTS
00027  4 14 00055                  25          ETR     EC+1,4              7700
00030  4 64 00060                  26          MUL     MC,4                -54/64 AT 0
00031  0 55 00020                  27          ADD     TA
00032  4 14 00054                  28          ETR     EC,4                7777
00033  0 46 00002                  29          CLB
00034  0 40 20001                  30          SKS     020001              TEST NEG EXPON
00035  0 46 01000                  31          CNA
00036  0 67 20002                  32          LCY     2
00037  4 64 00061                  33          MUL     MC+1,4              LOG 10 AT 2
00040  0 46 00412                  34          RCH     0412                CAX + CBA + CLB
00041  0 66 20001                  35          RCY     1
I      00042  1 01 00000          36          EXP     0                   ENTER 2**X
00043  0 66 20002                  37          RCY     2
00044  0 46 10012                  38          BAC
```

```
00045  0 64 00002      39          MUL     TI
00046  4 41 00047      40          BRX     $+1,4          INCREMENT X
00047  0 67 10036      41          NOD     30             NORMALIZE RESULT
00050  0 46 00040      42          CXB                    RESTORE REGISTERS
00051  0 71 00021      43          LDX     TX             RESTORE X
00052  4 51 00053      44          BRR     EXIT,4
00053  0 00 00000      45  EXIT    PZE
00054  00007777        46  EC      DATA    07777
00055  00007700        47          DATA    07700
00056  77777700        48          DATA    077777700
00057  77777773        49  XC      DATA    077777773
00060  45000000        50  MC      DATA    045000000
00061  32446474        51          DATA    032446474
00062  03641100        52  DC      DATA    03641100
00063  25400000        53  CON     DATA    025400000
       00000021        54  TX      EQU     021
       00000020        55  TA      EQU     020
       00000017        56  TB      EQU     017
       00000016        57  TS      EQU     016
       00000002        58  TI      EQU     02
                       59          END
```

IDENTIFICATION: Decimal to Binary Conversion, Double Precision Floating Point - DBF

AUTHOR: W. LaSor, F. Valadez, SDS

ACCEPTED: 26 April 1963

COMPUTER CONFIGURATION: Any SDS 920/930 Computer

PURPOSE: To convert a decimal number expressed in scientific notation to a double precision floating point number.

PROGRAMMED OPERATORS: DPM

STORAGE: 251 oct, 169 dec

TIMING: 3.8 - 8.5 milliseconds

SOURCE LANGUAGE: SYMBOL

USE:

1. INPUT

The argument is a BCD number consisting of sign, decimal point, 11 digits, and signed decimal exponent. The argument occupies the A and B registers and memory locations 00002 and 00003 as shown below:

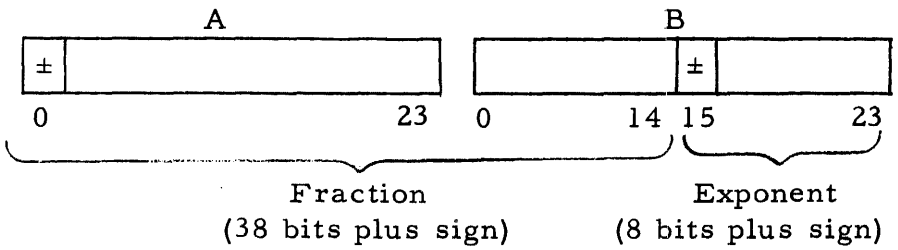
B	A	2	3
+.NN	NNNN	NNNN	N±EE
Fraction			Exponent

Where N represents the digits of the mantissa and E represents the digits of the exponent.

2. OUTPUT

The subroutine exits with a double precision floating point number in the combined A, B registers as shown below:

USE: (cont.)



3. ACCURACY

The maximum relative error (E_r) observed is 10^{-11}, where:

$$E_r = \left| \frac{N^* - N}{N} \right|$$

N^* = computed value

N = true value

4. ERROR

If the absolute argument exceeds 10^{77}, the overflow indicator will be turned on and the result will be set to 0.

METHOD: The decimal fraction is formed as a double precision integer, then multiplied by 10^{-11} to form a fraction.

The decimal exponent is converted to binary and examined bit-by-bit. For each bit that is a one, the mantissa is multiplied by a power of 10 from a table of $_{10}2^n$.

All operations are performed on double precision normalized numbers with exponents maintained in a third word. Upon completion of processing, the final exponent is packed into the last 9 bits of the B register.

```
                                    1  $DBF    POPD    017100000
00000   4 37 00241                  2          STX     TX,4            SAVE RETURN
00001   0 71 00000                  3          LDX     0
00002   4 37 00245                  4          STX     EXIT,4
00003   0 46 00014                  5          XAB
00004   4 35 00246                  6          STA     SIGN,4
00005   4 43 00123                  7          BRM     SEP,4           FORM HIGH-ORDER
00006   4 76 00246                  8          LDA     SIGN,4
00007   0 66 00006                  9          RSH     6               SIX DIGITS AT 46
00010   4 14 00157                 10          ETR     077,4
00011   4 43 00132                 11          BRM     MUL,4
00012   4 64 00153                 12          MUL     100K,4
00013   4 35 00243                 13          STA     SUM+1,4
00014   4 36 00242                 14          STB     SUM,4
00015   0 76 00003                 15          LDA     3               FORM LOW-ORDER
00016   0 75 00002                 16          LDB     2               FIVE DIGITS AT 46
00017   0 67 20006                 17          LCY     6
00020   4 35 00244                 18          STA     SE,4
00021   4 43 00123                 19          BRM     SEP,4
00022   0 46 30003                 20          CLR
00023   4 43 00132                 21          BRM     MUL,4
00024   0 67 00001                 22          LSH     1
00025   4 55 00242                 23          ADD     SUM,4           FORM FINAL N
00026   0 46 00014                 24          XAB
00027   4 57 00243                 25          ADC     SUM+1,4
00030   4 53 00246                 26          SKN     SIGN,4          IF N NEGATIVE,
00031   4 01 00040                 27          BRU     $+7,4           FORM COMPLEMENT
00032   4 35 00243                 28          STA     SUM+1,4
00033   4 36 00242                 29          STB     SUM,4
00034   0 46 30003                 30          CLR
00035   4 54 00242                 31          SUB     SUM,4
00036   0 46 00014                 32          XAB
00037   4 56 00243                 33          SUC     SUM+1,4
00040   4 71 00151                 34          LDX     D46,4           NORMALIZE N
00041   0 67 10056                 35          NOD     46
I  00042   5 01 00154              36          DPM     10R,4
00043   0 67 10001                 37          NOD     1
00044   4 35 00243                 38          STA     SUM+1,4
```

```
00045  4 36 00242      39          STB     SUM,4
00046  0 46 00200      40          CXA
00047  4 55 00156      41          ADD     REXP,4
00050  4 35 00232      42          STA     EXP,4
00051  4 75 00244      43          LDB     SE,4                CONVERT EXPONENT
00052  0 67 00014      44          LSH     12
00053  4 14 00157      45          ETR     077,4
00054  4 64 00150      46          MUL     10B24,4
00055  4 36 00234      47          STB     BLK,4
00056  0 76 00003      48          LDA     3
00057  0 46 00002      49          CLB
00060  4 14 00157      50          ETR     077,4
00061  4 55 00234      51          ADD     BLK,4
00062  4 71 00146      52          LDX     DM21,4
00063  0 66 00001      53  EX      RSH     1                   TEST EXP BITS
00064  0 52 00250      54          SKB     M0
00065  4 01 00103      55          BRU     TEST,4
00066  2 77 00002      56          EAX     2,2
00067  4 41 00063      57          BRX     $-4,4
00070  4 76 00232      58          LDA     EXP,4               TEST FOR RANGE
00071  0 02 20001      59          R0V
00072  4 73 00152      60          SKG     D255,4
00073  4 73 00144      61          SKG     DM257,4
00074  4 01 00141      62          BRU     ERR,4
00075  4 76 00243      63          LDA     SUM+1,4             GET RESULTS
00076  4 75 00242      64          LDB     SUM,4
00077  4 71 00232      65          LDX     EXP,4
00100  0 46 00140      66          LDE
00101  4 71 00241      67          LDX     TX,4
00102  4 51 00245      68          BRR     EXIT,4
00103  4 37 00233      69  TEST    STX     SAVEX,4
00104  4 62 00243      70          XMA     SUM+1,4
00105  4 75 00242      71          LDB     SUM,4
00106  4 53 00244      72          SKN     SE,4
00107  2 77 00025      73          EAX     21,2
00110  6 77 00160      74          EAX     TABN,6
I 00111  3 01 00025      75          DPM     21,2
00112  2 71 00027      76          LDX     23,2                GET EXPONENT
00113  0 67 10001      77          NOD     1
```

```
00114  4 62 00243      78          XMA     SUM+1,4
00115  0 46 00600      79          XXA
00116  4 63 00232      80          ADM     EXP,4
00117  4 76 00233      81          LDA     SAVEX,4
00120  0 46 00600      82          XXA
00121  4 36 00242      83          STB     SUM,4
00122  4 01 00066      84          BRU     EX+3,4
00123  0 00 00000      85  SEP     PZE                         SEPARATE CHARACTERS
00124  4 71 00145      86          LDX     DM5,4
00125  4 14 00157      87          ETR     077,4
00126  6 35 00241      88          STA     BLK+5,6
00127  0 67 20006      89          LCY     6
00130  4 41 00125      90          BRX     $-3,4
00131  4 51 00123      91          BRR     SEP,4
00132  0 00 00000      92  MLL     PZE                         FORM PRODUCT
00133  4 71 00145      93          LDX     DM5,4
00134  4 64 00150      94          MUL     10B24,4
00135  0 46 00014      95          XAB
00136  6 55 00241      96          ADD     BLK+5,6
00137  4 41 00134      97          BRX     $-3,4
00140  4 51 00132      98          BRR     MUL,4
00141  4 51 00141      99  ERR     BRR     $,4                 SET O.F.
00142  0 46 30003     100          CLR
00143  4 01 00101     101          BRU     TEST-2,4
00144  77777377       102  DM257   DATA    -257
00145  77777773       103  DM5     DATA    -5
00146  77777753       104  DM21    DATA    -21
00147  00000027       105  D23     DATA    23
00150  00000005       106  10B24   DATA    10*/(23-24)
00151  00000056       107  D46     DATA    46
00152  00000377       108  D255    DATA    255
00153  00303240       109  100K    DATA    100000
00154  41362634       110  10R     DATA    041362634,025772777
00155  25772777
00156  77777734       111  REXP    DATA    077777734
00157  00000077       112  077     DATA    077
00160  31463146       113  TABN    DATA    031463146,031463146,077777775
00161  31463146
00162  77777775
```

```
00163  07534122          114          DATA    07534122,024365o05,077777772
00164  24365605
00165  77777772
00166  53070415          115          DATA    053070415,032155613,077777763
00167  32155613
00170  77777763
00171  42141061          116          DATA    042141061,025363073,077777746
00172  25363073
00173  77777746
00174  27661050          117          DATA    027661050,034645312,077777713
00175  34645312
00176  77777713
00177  25521240          118          DATA    025521240,031754217,077777626
00200  31754217
00201  77777626
00202  04751236          119          DATA    04751236,025037765,077777454
00203  25037765
00204  77777454
00205  00000000          120  TABP    DATA    0,024000000,04
00206  24000000
00207  00000004
00210  00000000          121          DATA    0,031000000,07
00211  31000000
00212  00000007
00213  00000000          122          DATA    0,023420000,016
00214  23420000
00215  00000016
00216  00000000          123          DATA    0,027657020,033
00217  27657020
00220  00000033
00221  67701000          124          DATA    067701000,021606744,066
00222  21606744
00223  00000066
00224  65014000          125          DATA    065014000,023561326,0153
00225  23561326
00226  00000153
00227  51201000          126          DATA    051201000,030236017,0325
00230  30236017
00231  00000325
```

```
00232  0 00 00000      127  EXP     PZE
00233  0 00 00000      128  SAVEX   PZE
00234                   129  BLK     BSS     5
00241  0 00 00000      130  TX      PZE
00242                   131  SUM     BSS     2
00244  0 00 00000      132  SE      PZE
00245  0 00 00000      133  EXIT    PZE
00246  0 00 00000      134  SIGN    PZE
00247  00000001        135  C1      DATA    1
00250  40000000        136  MC      DATA    040000000
                       137          END
```

IDENTIFICATION: Binary to Decimal Conversion, Single, Fixed - BID

AUTHOR: Richard S. Resnick, SDS

ACCEPTED: 9 April 1963

COMPUTER CONFIGURATION: Any SDS 920/930 Computer

PURPOSE: To convert a single precision fixed point binary number in A, with the binary point location in the address field of the BID programmed operator, into six 6-bit characters in A, B with sign and decimal point.

Eight characters (two words) are necessary to represent the final converted decimal number completely formated for output.

PROGRAMMED OPERATORS: None

STORAGE: Instructions and constants: 141 oct, 97 dec

Uses temporary storage locations 12 through 15.

TIMING: 1.904 to 2.240 milliseconds

SOURCE LANGUAGE: SYMBOL

USE: 1. FUNCTION

To convert a fixed point binary number in A with the binary point location in the address field of the BID programmed operator into six 6-bit BCD characters in A, B with sign and decimal point. The final eight character decimal number with format symbols replaces the argument in A, B in the following manner:

Range of Argument	B	A
$0 \leq \|A\| < 1$	± . X X	X X X X
$1 \leq \|A\| < 10$	± X . X	X X X X
$10 \leq \|A\| < 10^2$	± X X .	X X X X
$10^2 \leq \|A\| < 10^3$	± X X X	. X X X
$10^3 \leq \|A\| < 10^4$	± X X X	X . X X
$10^4 \leq \|A\| < 10^5$	± X X X	X X . X
$10^5 \leq \|A\| < 10^6$	± X X X	X X X .

USE: (cont.)

2. ARGUMENT

The argument is a single precision number in A with the binary point location in the 9 least significant bits of the programmed operator address. The Q (binary point location) can range from +41 to -41. If the Q exceeds this or if the absolute value of the argument is equal to or greater than 10^6 at Q = 23, overflow will be set and the program will exit. When overflow occurs the original argument is lost.

3. ACCURACY

The output is accurate to six decimal digits, i.e., the least significant digit is within ± 1. Truncation rather than rounding is performed.

METHOD:

The argument is made positive and 23 is subtracted from Q. If the difference is positive, A is shifted left Q-23 places. If the difference is negative, A is shifted right 23 -Q places. This scales A at 23. A is then tested for the interval in which it lies. The intervals are denoted by subscript i and are:

$i = 6 \qquad 10^5 \leq A < 10^6$

$i = 5 \qquad 10^4 \leq A < 10^5$

$i = 4 \qquad 10^3 \leq A < 10^4$

$i = 3 \qquad 10^2 \leq A < 10^3$

$i = 2 \qquad 10 \leq A < 10^2$

$i = 1 \qquad 1 \leq A < 10$

$i = 0 \qquad 0 \leq A < 1$

A is then multiplied by $10^{-(i-1)}$. This results in a number greater than 1 and less than 10. This is the first decimal digit. By successively extracting the newly formed digit, and multiplying by 10, all six digits are formed. The placing of the decimal point is a function of i.

```
                                1  $EID    POPD    017200000
00000   0 37 00015              2          STX     TX
00001   0 77 40000              3          EAX     *0
00002   0 72 00140              4          SKA     =040000000
00003   4 01 00006              5          BRU     $+3,4
00004   4 75 00136              6          LDB     PLUS,4              PLUS SIGN
00005   4 01 00010              7          BRU     $+3,4
00006   4 75 00137              8          LDB     MINUS,4             MINUS SIGN
00007   0 46 01000              9          CNA
00010   0 36 00012             10          STB     WKB                 INITIATE FORMAT WORD
00011   0 46 00040             11          CXB
00012   0 46 00122             12          STE                         INSPECT Q
00013   0 46 00204             13          RCH     0204                CAB,CXA
00014   4 73 00121             14          SKG     86Q23,4             TEST Q GTE 87
00015   4 73 00120             15          SKG     N41Q23,4            TEST Q GTR -41
00016   4 51 00057             16          BRR     OVFLO,4
00017   4 54 00122             17          SUB     23Q23,4
00020   0 72 00140             18          SKA     =040000000
00021   4 01 00025             19          BRU     $+4,4
00022   0 46 00412             20          RCH     0412                CAX,BAC
00023   2 67 00000             21          LSH     0,2
00024   4 01 00030             22          BRU     $+4,4
00025   0 46 01000             23          CNA
00026   0 46 00412             24          RCH     0412                CAX,BAC
00027   26624000               25          DATA    026624000
00030   0 72 00140             26          SKA     =040000000
00031   4 51 00057             27          BRR     OVFLO,4
00032   4 73 00110             28          SKG     TENPX,4             TEST A GTE 10**6
00033   0 40 20001             29          SKS     020001
00034   4 51 00057             30          BRR     OVFLO,4
00035   4 71 00117             31          LDX     N7Q23,4             FIND RANGE OF Q
00036   6 73 00120             32          SKG     TENPX+8,6           TEST ARG GTE 10**X
00037   4 41 00036             33          BRX     $-1,4
00040   6 23 00110             34          EXU     LSH+7,6             LEFT SHIFT
00041   6 64 00134             35          MUL     TENMX+7,6           A*10**-X
00042   0 46 00014             36          XAB
00043   4 14 00135             37          ETR     MASK,4              SAVE 18 BITS OF B
00044   0 63 00012             38          ADM     WKB                 STORE BITS INTO WKB
```

```
00045  0 46 00014      39          XAB
00046  4 75 00124      40          LDB     2Q23,4
00047  4 43 00060      41          BRM     CONVRT,4        FORM FIRST 4 DIGITS
00050  0 36 00013      42          STB     WKA
00051  4 75 00123      43          LDB     3Q23,4
00052  4 43 00060      44          BRM     CONVRT,4        FORM NEXT 4 DIGITS
00053  0 76 00013      45          LDA     WKA
00054  0 46 00014      46          XAB
00055  0 71 00015      47          LDX     TX
00056  0 51 00000      48          BRR     0               EXIT
00057  4 51 00054      49  OVFLO   BRR     $-3,4           ERROR EXIT
00060  0 00 00000      50  CONVRT  PZE                     CONVERT BINARY TO
00061  0 36 00014      51          STB     FLAG            DECIMAL AND FORMAT
00062  0 75 00012      52          LDB     WKB
00063  4 41 00071      53  LOOP    BRX     DIGIT,4         TEST FOR DECIMAL
00064  0 46 00400      54          CAX                     SAVE NEXT DIGIT
00065  4 76 00134      55          LDA     POINT,4         DECIMAL POINT
00066  0 67 20006      56          LCY     6
00067  0 46 00600      57          XXA                     RESTORE NEXT DIGIT
00070  4 01 00076      58          BRU     COUNT,4
00071  0 67 20006      59  DIGIT   LCY     6
00072  0 36 00012      60          STB     WKB
00073  06624001        61          DATA    06624001
00074  4 64 00133      62          MUL     10Q5,4          FORM NEXT DIGIT
00075  0 75 00012      63          LDB     WKB
00076  0 60 00014      64  COUNT   SKR     FLAG            COUNT 4 DIGITS
00077  4 01 00063      65          BRU     LOOP,4
00100  4 51 00060      66          BRR     CONVRT,4
00101  0 67 00002      67  LSH     LSH     2               A GTE 10**5
00102  0 67 00005      68          LSH     5               A GTE 10**4
00103  0 67 00011      69          LSH     9               A GTE 10**3
00104  0 67 00014      70          LSH     12              A GTE 10**2
00105  0 67 00017      71          LSH     15              A GTE 10**1
00106  0 67 00023      72          LSH     19              A GTE 10**0
00107  0 67 00027      73          LSH     23              A LESS THAN 1
00110  03641077        74  TENPX   DATA    999999
00111  00303237        75          DATA    99999
00112  00023417        76          DATA    9999
00113  00001747        77          DATA    999
```

```
00114  00000143          78           DATA    99
00115  00000011          79           DATA    9
00116  00000000          80           DATA    0
00117  77777771          81  N7Q23    DATA    -7
00120  77777727          82  N41Q23   DATA    -41
00121  00000126          83  86Q23    DATA    86
00122  00000027          84  23Q23    DATA    23
00123  00000003          85  3Q23     DATA    3
00124  00000002          86  2Q23     DATA    2
00125  24761327          87  TENMX    DATA    024761327
00126  32155615          88           DATA    032155615
00127  20304470          89           DATA    020304470
00130  24365610          90           DATA    024365610
00131  31463150          91           DATA    031463150
00132  20000000          92           DATA    020000000
00133  12000000          93  10Q5     DATA    012000000
00134  33000400          94  P0INT    DATA    033000400
00135  77777700          95  MASK     DATA    077777700
00136  00000012          96  PLUS     DATA    012
00137  00000040          97  MINUS    DATA    040
       00000012          98  WKB      EQU     012
       00000013          99  WKA      EQU     013
       00000014         100  FLAG     EQU     014
       00000015         101  TX       EQU     015
                        102           END
00140  40000000
```

IDENTIFICATION: Decimal to Binary Conversion, Single, Fixed - DIB

AUTHOR: Richard S. Resnick, SDS

ACCEPTED: 24 April 1963

COMPUTER CONFIGURATION: Any SDS 920/930 Computer

PURPOSE: To convert a six digit decimal number with sign and decimal point from eight 6-bit BCD characters in A, B to a fixed point single precision binary number in A, B.

PROGRAMMED OPERATORS: None

STORAGE: Instructions and constants: 123 oct, 83 dec

Uses temporary storage locations 12 thru 21.

TIMING: 2.400 to 2.784 milliseconds

SOURCE LANGUAGE: SYMBOL

USE:

1. FUNCTION

To convert a six digit decimal number with sign and decimal point from eight 6-bit BCD characters in A, B to a fixed point single precision binary number in A, B at a Q (binary point location) designated by the address field of the DIB programmed operator.

2. ARGUMENT

The argument is eight 6-bit BCD characters in A, B. Six of these characters are numbers, one character a sign and one character a decimal point. The sign is the first character followed by seven more characters of which one is a decimal point. The characters must be arranged in A, B in the following manner:

± X X X	X X X X
B	A

The seven X's represent 6 decimal digits and one decimal point.

USE: (cont.) The address field of the DIB programmed operator determines the binary scaling of the resultant binary number.

3. ACCURACY

The resultant binary number is accurate to 24 bits.

4. ERROR

Overflow is set when the converted argument is too large to fit at the desired Q.

METHOD: The argument is converted to a single precision integer scaled at 23 by successively multiplying by ten and adding. The integer is normalized and multiplied by 10^{-i} where i is equal to the number of digits to the left of the decimal point. The resultant number is then scaled to the desired binary location by shifting.

```
                                    1  $DIB    POPD    017300000
00000   0 37 00012                  2          STX     TX
00001   0 35 00013                  3          STA     WKA
00002   0 36 00014                  4          STB     WKB             SAVE SIGN
00003   0 46 00023                  5          RCH     023             CBX,CLR
00004   0 35 00015                  6          STA     COUNT           INIT. COUNT EQU 0
00005   0 35 00016                  7          STA     SUM             INIT. SUM EQU 0
00006   4 43 00055                  8          BRM     CONVRT,4        CONVERT WKB
00007   0 71 00013                  9          LDX     WKA
00010   4 43 00055                 10          BRM     CONVRT,4        CONVERT WKA
00011   0 71 00017                 11          LDX     EXP
00012   0 76 00016                 12          LDA     SUM
00013   6 64 00112                 13          MUL     10MX,6          SUM*10**-X
00014   6 71 00103                 14          LDX     SCALE,6         X EQU SCALE
00015   0 67 10057                 15          NOD     47              X EQU SCALE + NQ
00016   0 37 00017                 16          STX     EXP             EXP EQU SCALE + NQ
00017   0 71 00012                 17          LDX     TX
00020   0 77 40000                 18          EAX     *0              X EQU 0
00021   0 46 00160                 19          RCH     0160
00022   0 46 00160                 20          RCH     0160            EXTEND SIGN OF Q
00023   0 46 00204                 21          RCH     0204            CAB, CXA
00024   0 54 00017                 22          SUB     EXP             Q-SCALE-NQ
00025   0 72 00122                 23          SKA     =040000000
00026   4 01 00041                 24          BRU     NEGQ,4
00027   4 73 00102                 25          SKG     47Q23,4
00030   4 01 00033                 26          BRU     $+3,4
00031   0 46 30003                 27          CLR
00032   4 01 00052                 28          BRU     EXIT,4
00033   0 46 00412                 29          RCH     0412            CAX, BAC
00034   0 53 00014                 30          SKN     WKB
00035   4 01 00037                 31          BRU     $+2,4
00036   0 46 01000                 32          CNA
00037   2 66 00000                 33          RSH     0,2
00040   4 01 00052                 34          BRU     EXIT,4
00041   0 46 01000                 35  NEGQ    CNA
00042   4 73 00102                 36          SKG     47Q23,4
00043   4 01 00045                 37          BRU     $+2,4
00044   4 51 00054                 38          BRR     OVFLO,4
```

```
00045  0 46 00412      39          RCH    0412                CAX, BAC
00046  0 53 00014      40          SKN     WKB
00047  4 01 00051      41          BRU     $+2,4
00050  0 46 01000      42          CNA
00051  2 67 00000      43          LSH     0,2
00052  0 71 00012      44  EXIT    LDX     TX
00053  0 51 00000      45          BRR     0
00054  4 51 00051      46  OVFLO   BRR     $-3,4
00055  0 00 00000      47  CONVRT  PZE
00056  4 76 00100      48          LDA     3Q23,4
00057  0 35 00020      49          STA     FLAG
00060  0 46 00041      50  LOOP    RCH     041                 CXB,CLA
00061  0 67 00006      51          LSH     6
00062  0 46 00022      52          RCH     022                 CBX,CLB
00063  4 73 00101      53          SKG     9Q23,4              TEST A GTR 9
00064  4 01 00070      54          BRU     DIGIT,4
00065  0 76 00015      55          LDA     COUNT
00066  0 35 00017      56          STA     EXP
00067  4 01 00075      57          BRU     COUNTR,4
00070  0 61 00015      58  DIGIT   MIN     COUNT
00071  0 62 00016      59          XMA     SUM
00072  4 64 00121      60          MUL     10Q4,4
00073  0 67 00004      61          LSH     4
00074  0 63 00016      62          ADM     SUM
00075  0 60 00020      63  COUNTR  SKR     FLAG
00076  4 01 00060      64          BRU     LOOP,4
00077  4 51 00055      65          BRR     CONVRT,4
00100  00000003        66  3Q23    DATA    3
00101  00000011        67  9Q23    DATA    9
00102  00000057        68  47Q23   DATA    47
00103  00000004        69  SCALE   DATA    4
00104  00000007        70          DATA    7
00105  00000012        71          DATA    10
00106  00000016        72          DATA    14
00107  00000021        73          DATA    17
00110  00000024        74          DATA    20
00111  00000030        75          DATA    24
00112  20615737        76  10MX    DATA    020615737
00113  24761327        77          DATA    024761327
```

```
00114  32155615          78            DATA    032155615
00115  20304470          79            DATA    020304470
00116  24365610          80            DATA    024365610
00117  31463147          81            DATA    031463147
00120  20000000          82            DATA    020000000
00121  24000000          83  10Q4      DATA    024000000
       00000012          84  TX        EQU     012
       00000013          85  WKA       EQU     013
       00000014          86  WKB       EQU     014
       00000015          87  COUNT     EQU     015
       00000016          88  SUM       EQU     016
       00000017          89  EXP       EQU     017
       00000020          90  FLAG      EQU     020
       00000021          91  AF        EQU     021
                         92            END
00122  40000000
```

IDENTIFICATION: LOG (2, e, 10) of A - LOG

AUTHOR: W. S. LaSor, SDS

ACCEPTED: 14 February 1963

COMPUTER CONFIGURATION: Any 920/930 Computer

PURPOSE: To compute the logarithm (base 2, e, 10) of an argument in the A register.

PROGRAMMED OPERATORS: None

STORAGE: Instructions and constants: 60 oct, 48 dec

Uses temporary storage locations 10 thru 13.

TIMING: 904 microseconds plus normalize time (4 microseconds per bit)

SOURCE LANGUAGE: SYMBOL

USE:

1. FUNCTION

The logarithm (base 2, e, 10) of the contents of A replaces the contents of A, B. The address field is used to define the base as follows:

ADDRESS	BASE
00000	2
00001	e
00002	10

Indexing and indirect addressing are permitted.

2. ARGUMENT

The argument is in the A register and the scaling of the argument is in the B register scaled at Q = 23. On exit, the result is in the A register scaled at Q = 5.

3. ERROR ALARMS

If the argument is less than or equal to zero, overflow is set and the subroutine exits with the registers unchanged. If

USE: (Cont.) the binary scaling, after normalization takes place, is greater than 31 in absolute value, no alarm is given but erroneous results occur.

METHOD: The argument is first tested and if greater than zero, is then normalized. This reduces the argument to the form:

$$X \cdot 2^b$$

where $1/2 \leq X < 1$

and b = binary scaling

The following approximation is used to compute $\log_2 X$:

$$\log_2 X = \left[\sum_{1}^{3} C_{2k-1} \cdot U^{2k-1}\right] - 1/2$$

for $1/2 \leq X < 1$

where $$u = \frac{X - \sqrt{1/2}}{X + \sqrt{1/2}}$$

and $C_1 = 2.8853913$

$C_3 = .96147063$

$C_5 = .59897865$

The result is scaled at Q = 5 and the binary scaling added in since:

$$\log_2 \left[X \cdot 2^b\right] \equiv b + \log_2 X$$

Finally, the answer is converted to the proper base by using the identity:

$$\log_k u \equiv \left[\log_2 U\right] \cdot \left[\log_k 2\right]$$

The absolute error ($\log^* X - \log X$) does not exceed 10^{-6} in magnitude.

```
                                    1  $LOG    POPD    012200000
00000   0 73 00057                  2  LOG     SKG     ZERO            TEST FOR LEGAL ARG
00001   4 51 00045                  3          BRR     EXIT,4          SET OVFLO
00002   0 37 00014                  4          STX     TX              SAVE X REGISTER
00003   0 46 00022                  5          RCH     022             CBX + CLB
00004   0 67 10026                  6          NOD     22              BRING INTO RANGE
00005   0 46 00600                  7          XXA
00006   4 64 00047                  8          MUL     1Q6,4           SCALE AT 5 IN B
00007   0 36 00013                  9          STB     CHAR            SAVE CHARACTERISTIC
00010   0 46 00200                 10          CXA
00011   0 66 00001                 11          RSH     1
00012   0 46 00004                 12          CAB
00013   4 55 00046                 13          ADD     K1,4            X+SQRT HALF
00014   0 35 00012                 14          STA     TA
00015   0 46 10012                 15          BAC
00016   4 54 00046                 16          SUB     K1,4            X-SQRT HALF
00017   0 65 00012                 17          DIV     TA              GET TRANSF. ARG AT 0
00020   0 35 00012                 18          STA     ARG
00021   0 64 00012                 19          MUL     ARG
00022   0 35 00015                 20          STA     ARGSQ
00023   4 64 00056                 21          MUL     C5,4            EVAL POLYNOMIAL
00024   4 55 00055                 22          ADD     C3,4
00025   0 64 00015                 23          MUL     ARGSQ
00026   0 66 00002                 24          RSH     2               SCALE AT 2
00027   4 55 00054                 25          ADD     C1,4
00030   0 64 00012                 26          MUL     ARG
00031   4 55 00053                 27          ADD     C0,4
00032   0 71 00014                 28          LDX     TX
00033   0 77 40000                 29          EAX     *0              GET BASE ADDRESS
00034   6 64 00050                 30          MUL     BASE,6
00035   0 62 00013                 31          XMA     CHAR            LOAD CHARACTERISTIC
00036   6 64 00050                 32          MUL     BASE,6
00037   0 62 00013                 33          XMA     CHAR
00040   0 46 00002                 34          CLB
00041   0 66 00003                 35          RSH     3               SCALE AT 5
00042   0 54 00013                 36          SUB     CHAR
00043   0 71 00014                 37          LDX     TX
00044   0 51 00000                 38          BRR     0               EXIT
```

```
00045  4 00 00043        39  EXIT   HLT     $-2,4
00046  13240475          40  K1     DATA    013240475
00047  00400000          41  1G6    DATA    1*/(23-6)
00050  40000000          42  BASE   DATA    040000000
00051  51643364          43         DATA    051643364
00052  66273731          44         DATA    066273731
00053  04000000          45  CC     DATA    04000000
00054  50725340          46  C1     DATA    050725340
00055  41167210          47  C3     DATA    041167210
00056  54652253          48  C5     DATA    054652253
       00000012          49  TA     EQU     10
       00000013          50  CHAR   EQU     11
       00000014          51  TX     EQU     12
       00000012          52  ARG    EQU     TA
       00000015          53  ARGSQ  EQU     13
00057  00000000          54  ZERO   DATA    0
                         55         END
```

IDENTIFICATION: Logarithm, Floating - LGF

AUTHOR: W. S. LaSor, SDS

ACCEPTED: 17 January 1964

COMPUTER CONFIGURATION: Any SDS 920/930 Computer

PURPOSE: To compute the double precision floating point logarithm of a double precision floating point argument in the A, B registers.

PROGRAMMED OPERATORS: DPM

STORAGE: Instructions and constants: 230 oct, 152 dec

TIMING: Base 2: 3.08 - 5.6 milliseconds

Base e, 10: 3.63 - 6.15 milliseconds

SOURCE LANGUAGE: SYMBOL

USE:

1. FUNCTION

The floating point logarithm (base 2, e, 10) of the contents of the A, B registers replaces the contents of A, B. The address field is used to define the base as follows:

ADDRESS	BASE
00000	2
00001	e
00002	10

Indexing and indirect addressing are permitted.

USE: (Cont.)

2. ARGUMENT

Both the argument and the result are in standard double precision floating point format (see Fixed Point Arithmetic-Double Precision description for explanation of format).

3. ERROR ALARMS

If the argument is less than or equal to zero, overflow is set and the subroutine exits with the registers unchanged.

METHOD:

The argument is first tested for magnitude and, if less than or equal to zero, overflow is set and the subroutine exits. Otherwise the exponent is extracted, floated, and saved as the characteristic for later use. The fraction, X, whose range is between 1/2 and 1, is then inspected to determine in which of 8 subintervals it falls and an appropriate multiplier is selected and the argument multiplied by this constant. At the same time, the logarithm, base 2, of this multiplier is accumulated in a sum which will be subtracted later. The intervals, together with the corresponding multipliers and logarithms are listed below:

Interval	Multiplier (A_k)	$\text{Log}_2 A_k$
.5 - .5625	1.8	.84799690655494
.5625 - .625	1.62	.69599381310989
.525 - .6875	1.47	.55581615506163
.6875 - .75	1.35	.43295940727610
.75 - .8125	1.24	.31034012061215
.8125 - .875	1.15	.20163386116965
.875 - .9375	1.08	.11103131238874
.9375 - .987	1.026	.03703073094497

The constants have been chosen such that after three multiplications at most, the resulting argument will lie in the range:

$$.987 \leq X \leq 1.013$$

METHOD: (Cont.) This logarithm is then computed by the approximation:

$$\log_2(1-X) = C_1X + C_2X^2 + C_3X^3 + C_4X^4$$

where:

$$C_1 = 1.4426950408889$$

$$C_2 = -.72134752044447$$

$$C_3 = .48089834696298$$

$$C_4 = -.36067376022224$$

and whose maximum error is 3×10^{-11}. The logarithm of the original argument is computed by:

$$\log_2 X = b + \log_2 X - \Sigma \log_2 A_k$$

where b is the floated exponent computed earlier.

Finally, the answer is converted to the proper base by using the identity:

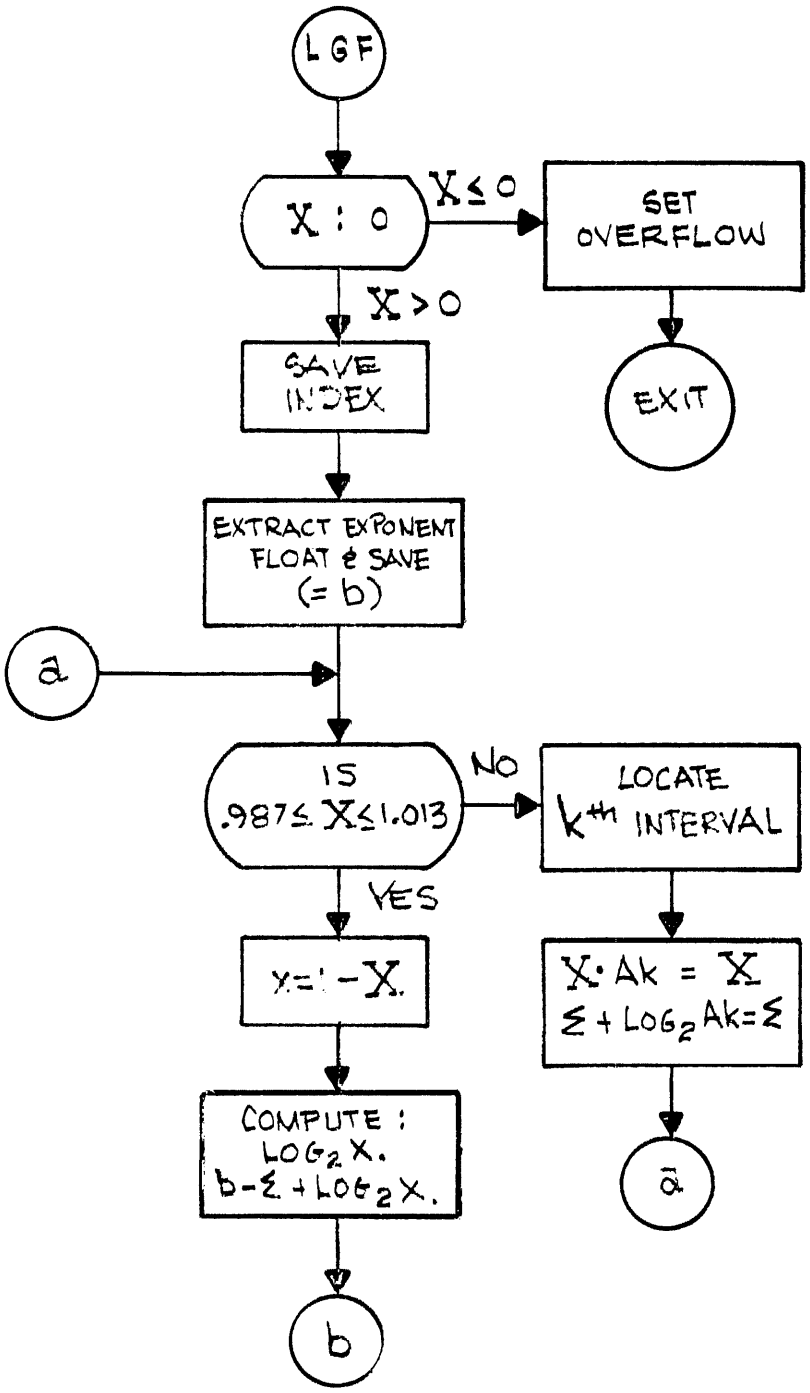
$$\log_k X \equiv (\log_2 X) \cdot (\log_k 2), \quad k = e, 10$$

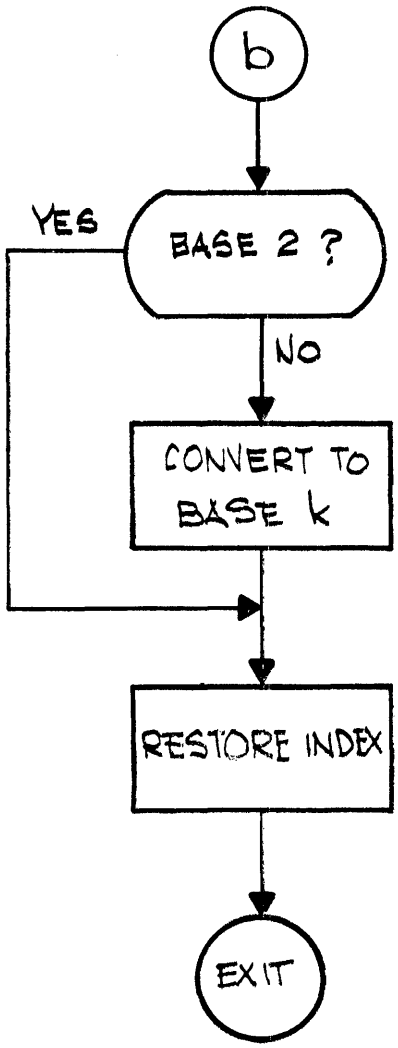
For X not in the interval (0.99, 1.01), the relative error is defined as:

$$\frac{\log^* X - \log X}{\log X}$$

and has not been observed to exceed 5×10^{-11} in magnitude. For values of X that fall in the interval stated above, the absolute error ($\log^* X - \log X$) does not exceed 5×10^{-11} in magnitude.

LOGARITHM, FLOATING - LGF




b
YES
BASE 2 ?
NO
CONVERT TO
BASE k
RESTORE INDEX
EXIT

```
                                1  SLGF    PSPD    012200000
00000  0 37 00224               2  LGF     STX     TX              SAVE INDEX
00001  0 71 00000               3          LDX     0               SAVE RETURN
00002  0 37 00220               4          STX     EXIT
00003  0 73 00225               5          SKG     ZERO            TEST FOR LEGAL ARG
00004  4 51 00131               6          BRR     ERROR,4
00005  0 46 00122               7          STE                     EXTRACT EXPONENT
00006  0 35 00216               8          STA     ARG+1           SAVE ARG
00007  0 36 00215               9          STB     ARG
00010  0 46 00202              10          RCH     0202            CXA + CLB
00011  0 77 00026              11          FAX     22              FLOAT EXPONENT
00012  0 67 10026              12          NOD     22
00013  4 35 00217              13          STA     CHAR,4          SAVE CHARACTERISTIC
00014  4 37 00221              14          STX     SHIFT,4         SAVE SHIFT COUNT
00015  0 46 30003              15          CLR                     PRESET CORRECTIVE SUM
00016  0 35 00223              16          STA     SUM+1
00017  0 36 00222              17          STB     SUM
00020  0 76 00216              18          LDA     ARG+1           BEGIN REDUCTION
00021  4 73 00144              19  LOOP    SKG     LOWER,4         TEST FOR RANGE
00022  0 72 00226              20          SKA     SIGN
00023  4 01 00045              21          BRU     EVAL,4          IN RANGE
00024  0 46 00002              22          CLB                     OUT OF RANGE
00025  0 67 20005              23          LCY     5               LOCATE SUBINTERVAL
00026  0 46 00020              24          CBX
00027  6 71 00122              25          LDX     LIST1-8,6       GET A(K) ADDRESS
00030  0 76 00222              26          LDA     SUM             ACCUM LOG A(K)
00031  2 54 00002              27          SUB     2,2
00032  0 35 00222              28          STA     SUM
00033  0 76 00223              29          LDA     SUM+1
00034  2 56 00003              30          SUC     3,2
00035  0 35 00223              31          STA     SUM+1
00036  0 76 00216              32          LDA     ARG+1
00037  0 75 00215              33          LDB     ARG
I      00040  3 01 00000       34          DPM     0,2             FORM ARG*A(K)
00041  0 67 20001              35          LCY     1               SCALE AT 0
00042  0 35 00216              36          STA     ARG+1
00043  0 36 00215              37          STB     ARG
00044  4 01 00021              38          BRU     LOOP,4          RETURN TO RANGE-TEST
```

```
00045  0 17 00226      39  EVAL   EOR     SIGN          FORM  1-ARG  AT 0
00046  0 35 00216      40         STA     ARG+1         EVAL POLYNOMIAL
00047  0 75 00215      41         LDB     ARG
I 00050  5 01 00147     42         DPM     C4,4          C4*ARG AT -1
00051  0 46 00014      43         XAB                   ADD C3 AT -1
00052  4 55 00145      44         ADD     C3,4
00053  0 46 00014      45         XAB
00054  4 57 00146      46         ADC     C3+1,4
I 00055  1 01 00215     47         DPM     ARG
00056  0 66 00001      48         RSH     1             SCALE AT 0
00057  0 46 00014      49         XAB
00060  4 55 00147      50         ADD     C2,4          ADD C2 AT 0
00061  0 46 00014      51         XAB
00062  4 57 00150      52         ADC     C2+1,4
I 00063  1 01 00215     53         DPM     ARG
00064  0 66 00001      54         RSH     1             SCALE AT 1
00065  0 46 00014      55         XAB
00066  4 54 00147      56         SUB     C1,4          ADD C1 AT 1
00067  0 46 00014      57         XAB
00070  4 56 00150      58         SUC     C1+1,4
I 00071  1 01 00215     59         DPM     ARG           LOG BASE 2 AT 1
00072  0 46 00014      60         XAB                   ADD CORRECTIVE SUM
00073  0 55 00222      61         ADD     SUM
00074  0 46 00014      62         XAB
00075  0 57 00223      63         ADC     SUM+1
00076  4 71 00221      64         LDX     SHIFT,4
00077  2 66 00000      65         RSH     0,2           SCALE RESULT
00100  4 55 00217      66         ADD     CHAR,4        ADD CHARACTERISTIC
00101  0 40 20001      67         SVT                   TEST OVERFLOW
00102  4 01 00121      68         BRU     OVFLO,4       GO CORRECT
00103  0 67 10060      69  NORM   NOD     48
00104  4 41 00105      70         BRX     $+1,4         FORM NEW EXPONENT
00105  0 37 00215      71         STX     ARG           SAVE NEW EXPONENT
00106  0 71 00224      72         LDX     TX            RECALL INDEX
00107  0 77 40220      73         EAX     *EXIT         GET BASE ADDRESS
00110  0 46 00600      74         XXA
00111  0 14 00227      75         ETR     ADDR
00112  0 50 00225      76         SKE     ZERO          TEST FOR BASE 2
00113  4 01 00125      77         BRU     CNVRT,4       CHANGE BASE
```

```
00114  0 46 00600      78          XXA
00115  0 71 00215      79          LDX     ARG                     LOAD AND PACK EXP
00116  0 46 00140      80          LDE
00117  0 71 00224      81          LDX     TX                      RESTORE X
00120  0 51 00220      82  OUT     BRR     EXIT
00121  0 66 00001      83  OVFLO   RSH     1                       RESCALE
00122  0 17 00226      84          EOR     SIGN                    CHANGE SIGN
00123  4 41 00103      85          BRX     NORM,4                  INCREMENT EXPONENT
00124  4 01 00103      86          BRU     NORM,4
00125  0 46 00600      87  CNVRT   XXA
I 00126  7 01 40141      88          DPM     *LIST2-1,6              CHANGE TO BASE X
00127  0 71 00215      89          LDX     ARG                     LOAD EXPONENT
00130  0 67 10002      90          NOD     2
00131  4 01 00116      91  ERROR   BRU     OUT-2,4                 PACK EXPON, EXIT
00132  4 00 00151      92  LIST1   HLT     AI1I,4                  MULTIPLIER ADDRESSES
00133  4 00 00155      93          HLT     AI2I,4
00134  4 00 00161      94          HLT     AI3I,4
00135  4 00 00165      95          HLT     AI4I,4
00136  4 00 00171      96          HLT     AI5I,4
00137  4 00 00175      97          HLT     AI6I,4
00140  4 00 00201      98          HLT     AI7I,4
00141  4 00 00205      99          HLT     AI8I,4
00142  4 00 00211     100  LIST2   HLT     BASEE,4                 BASE CHANGE ADDRESSES
00143  4 00 00213     101          HLT     BASE10,4
00144  37453004       102  LOWER   DATA    037453004               .987*/(23-0)
00145  34164010       103  C3      DED     .480898346963*/(47+1)
00146  36616047
00147  32651004       104  C4      DED     -.3606737602222*/(47+1)
00150  50725342
00151  14631464       105  AI1I    DED     1.6*/(47-1)
00152  34631463
00153  64227554       106          DED     .84799690655*/(47-1)
00154  15442624
00155  36560504       107  AI2I    DED     1.62*/(47-1)
00156  31727024
00157  50460640       108          DED     .69599381311*/(47-1)
00160  13105451
00161  70243654       109  AI3I    DED     1.47*/(47-1)
00162  27412172
```

```
00163  73022372        110             DED     .5558161550616*/(47-1)
00164  10711175
00165  31463144        111     AI4I    DED     1.35*/(47-1)
00166  25463146
00167  27727703        112             DED     .4329594072761*/(47-1)
00170  06732633
00171  75341216        113     AI5I    DED     1.24*/(47-1)
00172  23656050
00173  63625231        114             DED     .3103401206121*/(47-1)
00174  04756234
00175  46314630        115     AI6I    DED     1.15*/(47-1)
00176  22314631
00177  55357525        116             DED     .2016338611696*/(47-1)
00200  03163621
00201  24365604        117     AI7I    DED     1.08*/(47-1)
00202  21217270
00203  04761246        118             DED     .1110313123887*/(47-1)
00204  01615443
00205  71666210        119     AI8I    DED     1.026*/(47-1)
00206  20324773
00207  11113304        120             DED     .037030730945*/(47-1)
00210  00457266
00211  76764340        121     BASEE   DED     .69314718056*/(47-0)
00212  26134413
00213  50237360        122     BASE10  DED     .301029995664*/(47-0)
00214  11504046
00215                  123     ARG     BSS     2
00217                  124     CHAR    BSS     1
00220                  125     EXIT    BSS     1
00221                  126     SHIFT   BSS     1
00222                  127     SLM     BSS     2
00224                  128     TX      BSS     1
00225  00000000        129     ZERO    DATA    0
00226  40000000        130     SIGN    DATA    040000000
00227  00037777        131     ADDR    DATA    037777
       00000147        132     C1      EQU     C4
       00000147        133     C2      EQU     C4
                       134             END
```

IDENTIFICATION: Exponential (2, e, 10) of A - EXP

AUTHOR: W. S. LaSor, SDS

ACCEPTED: 5 March 1963

COMPUTER CONFIGURATION: Any 920/930 Computer

PURPOSE: To compute the exponential (base 2, e, or 10) of an argument in the A register.

PROGRAMMED OPERATOR: None

STORAGE: Instructions and constants: 76 oct, 62 dec

Uses temporary storage locations 10 through 12.

TIMING: 824 microseconds plus scaling time

SOURCE LANGUAGE: SYMBOL

USE:

1. FUNCTION

The exponential (base 2, e, or 10) of the contents of A replaces the contents of A, B. The address field is used to define the base as follows:

ADDRESS	BASE
00000	2
00001	e
00002	10

Indexing and indirect addressing are permitted.

2. ARGUMENT

The argument is in the A register and the scaling of the argument is B scaled at Q = 23. On exit, the result is in the A, B registers scaled at Q = 23.

3. ERROR ALARMS

If the value of independent variable is such that its exponential exceeds the capacity of the A register, overflow is set and

USE: (Cont.) the A, B registers contain erroneous data. The approximate maximum values are listed below:

BASE	Xmax
2	23.99999
e	15.94238
10	6.923689

METHOD: The argument is first converted to its base 2 equivalent using the identity:

$$k^x \equiv 2^{x \log_2 k} \; ; \; k = e, \; 10.$$

The result is then shifted to a scaling of 23 to obtain the form:

$$2^{i.f} \equiv 2^i \cdot 2^{.f}$$

where i and f are the integral and fractional parts of the argument. The following polynominal is used to compute $2^{.f}$:

$$\left[\sum_0^6 C_k X^k\right]^2$$

$C_0 = 1.0000000$

$C_1 = .34657210$

$C_2 = .06006622$

$C_3 = .00691806$

$C_4 = .00061973$

$C_5 = .000033177$

$C_6 = .000004208$

and i is used to scale the result since 2^i may be accomplished by a shift in a binary machine.

For $X \geq 0$, the relative error is no greater than 10^{-6} in magnitude. For $X < 0$, the absolute error does not exceed 10^{-6}.

```
                                    1  $EXP   POPD    012300000
00000  0 37 00014                   2  EXP    STX     TX                  SAVE X
00001  4 74 00072                   3         SKD     P21,4
00002  4 01 00053                   4         BRU     LEFT,4              GO SCALE LEFT
00003  0 37 00012                   5         STX     TA                  SAVE DIFF
00004  0 71 00014                   6         LDX     TX
00005  0 77 40000                   7         EAX     *0                  LOAD BASE ADDRESS
00006  6 64 00060                   8         MUL     BASE,6              REDUCE TO BASE 2
00007  0 71 00012                   9         LDX     TA
00010  2 66 00000                  10         RSH     0,2                 SCALE AT 23
00011  0 35 00013                  11  SAVE   STA     EXPON               SAVE INTEGER PART
00012  0 46 10012                  12         BAC
00013  0 66 20001                  13         RCY     1                   AFFIX SIGN TO F
00014  0 35 00012                  14         STA     TA
00015  4 64 00071                  15         MUL     C6,4                EVAL POLYNOMIAL
00016  4 55 00070                  16         ADD     C5,4
00017  0 64 00012                  17         MUL     TA
00020  4 55 00067                  18         ADD     C4,4
00021  0 64 00012                  19         MUL     TA
00022  4 55 00066                  20         ADD     C3,4
00023  0 64 00012                  21         MUL     TA
00024  4 55 00065                  22         ADD     C2,4
00025  0 64 00012                  23         MUL     TA
00026  4 55 00064                  24         ADD     C1,4
00027  0 64 00012                  25         MUL     TA
00030  0 66 00001                  26         RSH     1
00031  4 55 00063                  27         ADD     C0,4
00032  0 35 00012                  28         STA     TA
00033  0 64 00012                  29         MUL     TA                  2**F AT 2
00034  0 55 00074                  30         ADD     ONE
00035  4 64 00073                  31         MUL     P1,4                SCALE AT 23
00036  0 46 00400                  32         CAX                         SAVE IN X
00037  0 76 00013                  33         LDA     EXPON
00040  0 72 00075                  34         SKA     SIGN
00041  4 01 00046                  35         BRU     NEG,4
00042  0 46 00600                  36         XXA
00043  2 67 00000                  37         LSH     0,2                 SCALE BY 2**I
00044  0 71 00014                  38         LDX     TX
```

```
00045  0 51 00000      39          BRR     0
00046  0 46 01000      40  NEG     CNA
00047  0 46 00600      41          XXA
00050  2 66 00000      42          RSH     0,2             SCALE BY 2**I
00051  0 71 00014      43          LDX     TX
00052  0 51 00000      44          BRR     C
00053  2 67 00000      45  LEFT    LSH     0,2             SCALE AT 21
00054  0 71 00014      46          LDX     TX
00055  0 77 40000      47          EAX     *0              GET BASE ADDRESS
00056  6 64 00060      48          MUL     BASE,6          REDUCE TO BASE 2
00057  4 01 00011      49          BRU     SAVE,4          GO EVAL POLYNOMIAL
00060  10000000        50  BASE    DATA    010000000
00061  13425217        51          DATA    013425217
00062  32446474        52          DATA    032446474
00063  20000000        53  CC      DATA    020000000
00064  13056171        54  C1      DATA    013056171
00065  01730100        55  C2      DATA    01730100
00066  00161261        56  C3      DATA    0161261
00067  00012117        57  C4      DATA    012117
00070  00000426        58  C5      DATA    0426
00071  00000044        59  C6      DATA    044
00072  00000025        60  P21     DATA    21
00073  00000004        61  P1      DATA    1*/(23-21)
       00000012        62  TA      EQU     10
       00000013        63  EXPON   EQU     11
       00000014        64  TX      EQU     12
00074  00000001        65  ONE     DATA    01
00075  40000000        66  SIGN    DATA    040000000
                       67          END
```

IDENTIFICATION: Exponential (2, e, 10), Floating - EXF

AUTHOR: W. S. LaSor, SDS

ACCEPTED: 17 January 1964

COMPUTER CONFIGURATION: Any SDS 920/930 Computer

PURPOSE: To compute the floating point exponential (base 2, e, 10) of a floating point argument in the A, B registers.

PROGRAMMED OPERATORS: DPM

STORAGE: Instructions and constants: 237 oct, 159 dec

TIMING: Base 2, 2.24 - 7.36 milliseconds

Base e, 10, 2.81 - 7.93 milliseconds

SOURCE LANGUAGE: SYMBOL

USE:

1. FUNCTION

The floating point exponential (base 2, e, 10) of the contents of A, B replaces the contents of A, B. The contents of the X register are unchanged. The address field of the programmed operator is used as follows to define the base:

ADDRESS	BASE
00000	2
00001	e
00002	10

Indexing and indirect addressing are permitted.

2. ARGUMENT

Both the argument and the result are in standard double precision floating point format (see Fixed Point Arithmet Double Precision Description for explanation of format).

USE: (Cont.)

3. ERROR ALARMS

If the result of exponentiation is greater than 2^{255} in magnitude, the result is set to $+2^{255}$ and overflow is set. If the result is less than 2^{-256} in magnitude, the answer is set to zero but overflow is not set.

METHOD:

The argument is first converted to its base 2 equivalent by multiplication by $\log_k 2$, k = e, 10. The result is then unfloated and the integer saved as the new exponent since:

$$2^{i.f} = 2^i \cdot 2^{.f}$$

which is standard binary floating point format. The fractional remainder is then made positive and its exponential initially set to 1. At this point, the subroutine begins subtracting off those factors of $\log_2(1+2^{-k})$ which can be subtracted and still leave a positive result. For each factor that is found, the exponential is multiplied by $(1+2^{-k})$ and this result replaces the previous value. A total of 17 factors are tried and the result after all possible factors have been subtracted will always be less than 1.1×10^{-5}. This value is used to evaluate:

$$2^x = 1 + C_1 X$$

where:

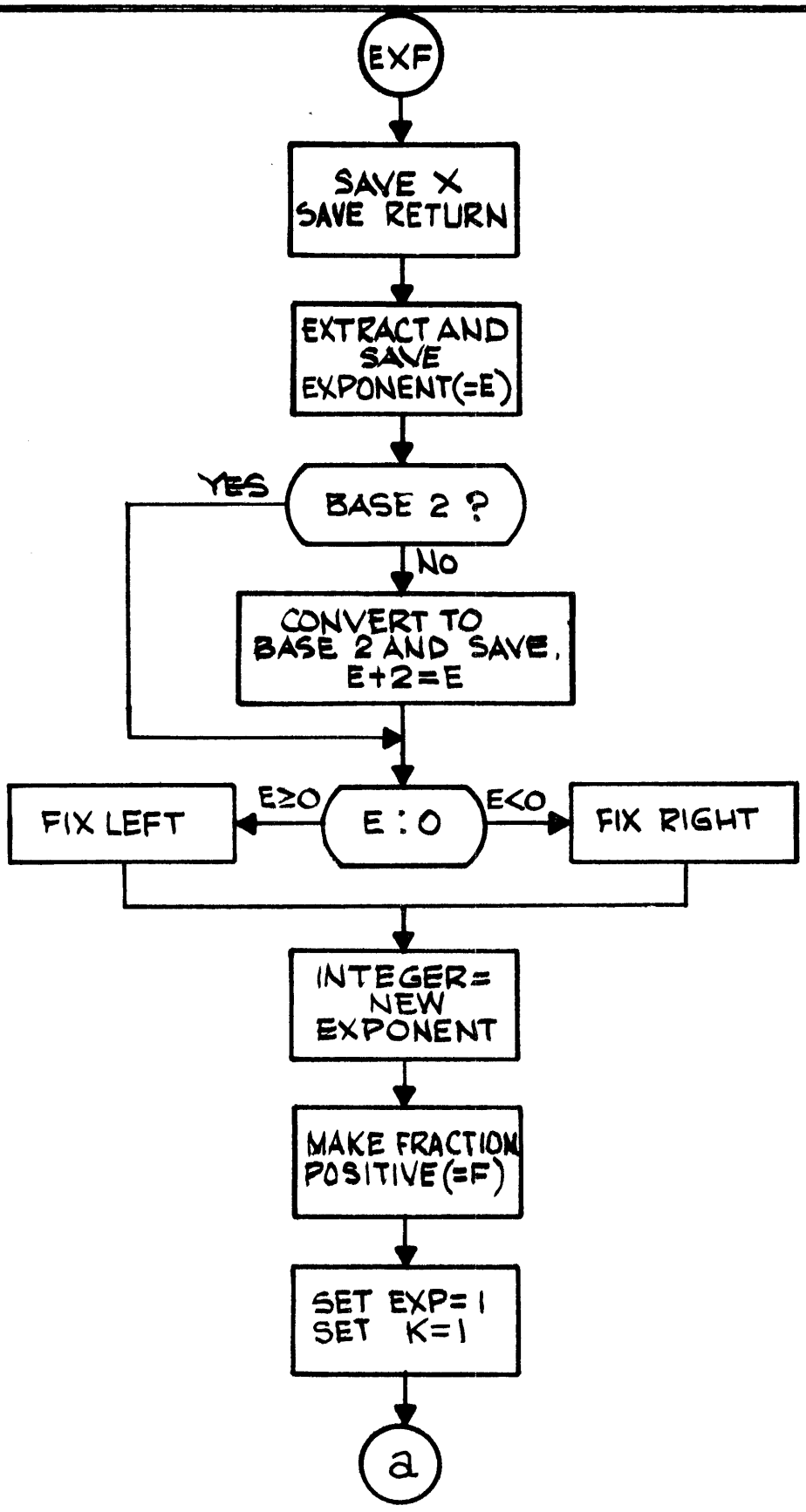
$$C_1 = .693147180578$$

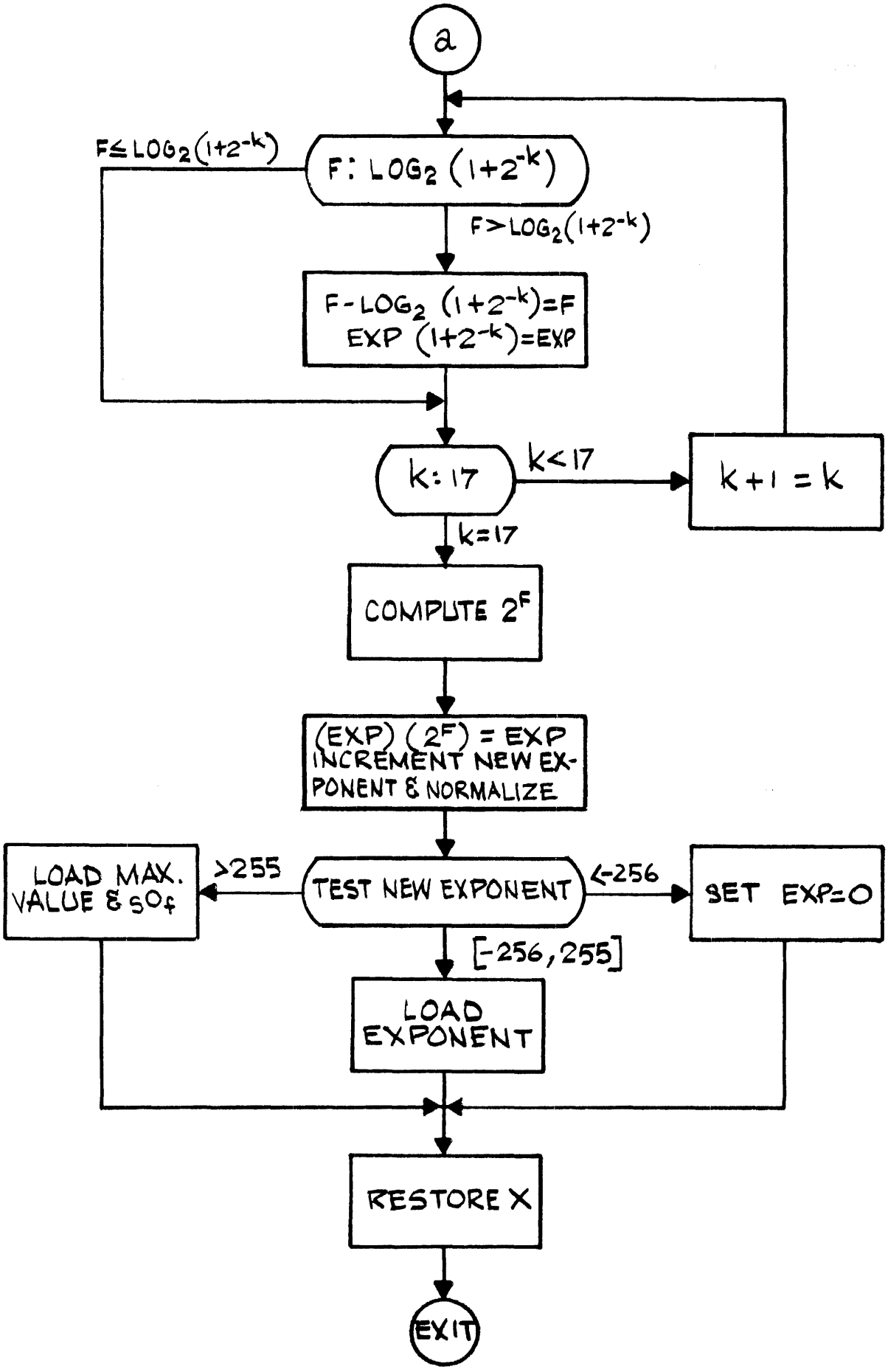
which is then multiplied by the accumulated exponential to obtain the final result. The answer is then normalized and, barring exponent overflow or underflow, the exponent is packed and the subroutine exits.

The maximum relative error:

$$\frac{(k^*)^x - k^x}{k^x} \qquad k = 2, e, 10$$

does not exceed 5×10^{-11} in magnitude.


EXF
SAVE X
SAVE RETURN
EXTRACT AND SAVE EXPONENT(=E)
YES
BASE 2 ?
No
CONVERT TO BASE 2 AND SAVE, E+2=E
FIX LEFT
E≥0
E : 0
E<0
FIX RIGHT
INTEGER= NEW EXPONENT
MAKE FRACTION POSITIVE(=F)
SET EXP=1
SET K=1
a


a
F≤LOG2(1+2-k)
F: LOG2 (1+2-k)
F>LOG2(1+2-k)
F-LOG2 (1+2-k)=F
EXP (1+2-k)=EXP
k: 17
k<17
k+1 = k
k=17
COMPUTE 2F
(EXP) (2F) = EXP
INCREMENT NEW EX-
PONENT & NORMALIZE
LOAD MAX.
VALUE & SOf
>255
TEST NEW EXPONENT
<-256
SET EXP=0
[-256,255]
LOAD
EXPONENT
RESTORE X
EXIT

```
                                1  $EXF    POPD    012300000
00000  0 37 00231               2  EXF     STX     TX              SAVE X
00001  0 71 00000               3          LDX     0               SAVE RETURN
00002  0 37 00225               4          STX     EXIT
00003  0 46 00122               5          STE                     EXTRACT EXPONENT
00004  4 37 00224               6          STX     E,4             SAVE
00005  0 71 00231               7          LDX     TX
00006  0 77 40000               8          EAX     *0              GET BASE ADDRESS
00007  0 46 00600               9          XXA
00010  0 72 00236              10          SKA     ADDR            TEST FOR BASE 2
00011  4 01 00117              11          BRU     CNVRT,4
00012  4 76 00224              12          LDA     E,4             RECALL EXPONENT
00013  0 73 00235              13  ETEST   SKG     ONES
00014  4 01 00125              14          BRU     RIGHT,4         SHIFT RIGHT TO FIX
00015  0 46 00600              15          XXA                     SHIFT LEFT TO FIX
00016  0 35 00222              16          STA     ARG+1           SAVE ARG
00017  0 36 00221              17          STB     ARG
00020  0 64 00233              18          MUL     ONE
00021  2 67 00000              19          LSH     0,2             FORM NEW EXPONENT
00022  4 35 00230              20          STA     S,4
00023  0 76 00222              21          LDA     ARG+1
00024  0 75 00221              22          LDB     ARG
00025  2 67 00000              23          LSH     0,2             GET FRACTION
00026  0 02 20001              24          ROV
00027  4 14 00146              25          ETR     MASK,4          MAKE ARG POSITIVE
00030  0 36 00221              26          STB     ARG
00031  0 35 00222              27  SET     STA     ARG+1
00032  4 76 00142              28          LDA     K1,4            SET ANSWER TO 1
00033  0 46 00002              29          CLB
00034  0 35 00227              30          STA     R+1
00035  0 36 00226              31          STB     R
00036  4 76 00145              32          LDA     P16,4           SET COUNT TO 17
00037  4 35 00223              33          STA     CNTR,4
00040  4 71 00143              34          LDX     K2,4
00041  0 76 00222              35          LDA     ARG+1           RECALL ARG
00042  0 75 00221              36          LDB     ARG
00043  6 73 00156              37  LOOP    SKG     TABLE-1,6       COMPARE KTH FACTOR
00044  4 01 00066              38          BRU     TEST,4
```

```
00045  0 46 00014      39          XAB                         SUBT KTH FACTOR
00046  6 54 00177      40          SUB     TABLE+16,6
00047  0 46 00014      41          XAB
00050  6 56 00156      42          SUC     TABLE-1,6
00051  0 35 00222      43          STA     ARG+1
00052  0 36 00221      44          STB     ARG
00053  0 76 00227      45          LDA     R+1                 MODIFY ANSWER
00054  0 75 00226      46          LDB     R
00055  2 66 00000      47          RSH     0,2
00056  0 46 00014      48          XAB
00057  0 55 00226      49          ADD     R
00060  0 46 00014      50          XAB
00061  0 57 00227      51          ADC     R+1
00062  0 35 00227      52          STA     R+1
00063  0 36 00226      53          STB     R
00064  0 76 00222      54          LDA     ARG+1
00065  0 75 00221      55          LDB     ARG
00066  4 60 00223      56  TEST    SKR     CNTR,4              TEST FOR 17TH TIME
00067  4 41 00043      57          BRX     LOOP,4
00070  5 00 00147      58          DPM     C1,4                COMPUTE 2**F
00071  4 55 00142      59          ADD     C0,4
00072  1 00 00226      60          DPM     R
00073  4 62 00230      61          XMA     S,4                 RECALL NEW EXPON
00074  4 55 00144      62          ADD     P2,4
00075  0 46 00400      63          CAX
00076  4 76 00230      64          LDA     S,4
00077  0 67 10002      65          NOD     2                   NORMALIZE RESULT
00100  0 46 00600      66          XXA
00101  4 73 00151      67          SKG     P255,4              TEST FOR EXPON OVFLO
00102  4 73 00152      68          SKG     M257,4
00103  4 01 00110      69          BRU     OUT+1,4             OVFLO
00104  0 46 00600      70          XXA
00105  0 46 00140      71          LDE                         PACK EXPONENT
00106  0 71 00231      72          LDX     TX                  RESTORE X
00107  0 51 00225      73  OLT     BRR     EXIT
00110  0 72 00234      74          SKA     SIGN                SET OVFLO IF GTR 255
00111  4 01 00115      75          BRU     $+4,4               UNDERFLOW
00112  4 76 00146      76          LDA     MASK,4              SET MAXIMUM VALUE
00113  4 75 00152      77          LDB     M257,4
```

```
00114  4 51 00141      78          BRR     OVFLO,4
00115  0 46 30003      79          CLR                         SET RESULT TO ZERO
00116  4 01 00106      80          BRU     OUT-1,4
00117  0 46 00600      81  CNVRT   XXA
00120  7 00 40136      82          DPM     *LIST-1,6           GET BASE EQUIVALENT
00121  4 62 00224      83          XMA     E,4
00122  4 55 00144      84          ADD     P2,4                ADD 2 TO EXPONENT
00123  4 71 00224      85          LDX     E,4
00124  4 01 00013      86          BRU     ETEST,4
00125  0 46 01000      87  RIGHT   CNA                         MAKE POSITIVE
00126  0 46 00600      88          XXA
00127  2 66 00000      89          RSH     0,2                 FIX ARGUMENT
00130  0 36 00221      90          STB     ARG
00131  0 46 00002      91          CLB
00132  0 72 00234      92          SKA     SIGN
00133  0 75 00235      93          LDB     ONES
00134  4 36 00230      94          STB     S,4                 FORM NEW EXPONENT
00135  4 14 00146      95          ETR     MASK,4              MAKE ARG POSITIVE
00136  4 01 00031      96          BRU     SET,4               CONTINUE
00137  4 00 00153      97  LIST    HLT     LOGE,4
00140  4 00 00155      98          HLT     LOG10,4
00141  4 00 00105      99  OVFLO   HLT     OUT-2,4             TO SET OVFLO
00142  20000000       100  K1     DATA    1*/(23-1)
00143  00040001       101  K2     DATA    040001
00144  00000002       102  P2     DATA    2
00145  00000020       103  P16    DATA    16
00146  37777777       104  MASK   DATA    037777777
00147  77374544       105  C1     DED     .693147180578*/(47-1)
00150  13056205
00151  00000377       106  P255   DATA    255
00152  77777377       107  M257   DATA    -257
00153  62453400       108  LOGE   DED     1.442695040889*/(47-2)
00154  13425216
00155  11363170       109  LOG10  DED     3.321928094887*/(47-2)
00156  32446474
00157  22560015       110  TABLE  DATA    022560015
00160  12232360       111         DATA    012232360
00161  05340032       112         DATA    05340032
00162  02630773       113         DATA    02630773
```

```
00163  01327264          114            DATA    01327264
00164  00556362          115            DATA    0556362
00165  00267745          116            DATA    0267745
00166  00134116          117            DATA    0134116
00167  00056076          118            DATA    056076
00170  00027044          119            DATA    027044
00171  00013423          120            DATA    013423
00172  00005612          121            DATA    05612
00173  00002705          122            DATA    02705
00174  00001342          123            DATA    01342
00175  00000561          124            DATA    0561
00176  00000270          125            DATA    0270
00177  00000134          126            DATA    0134
00200  07176750          127            DATA    07176750
00201  45715062          128            DATA    045715062
00202  16375721          129            DATA    016375721
00203  37262205          130            DATA    037262205
00204  67261667          131            DATA    067261667
00205  64134120          132            DATA    064134120
00206  02667347          133            DATA    02667347
00207  10666317          134            DATA    010666317
00210  03776033          135            DATA    03776033
00211  62467170          136            DATA    062467170
00212  65427550          137            DATA    065427550
00213  21660521          138            DATA    021660521
00214  16542177          139            DATA    016542177
00215  50623543          140            DATA    050623543
00216  24602400          141            DATA    024602400
00217  52357324          142            DATA    052357324
00220  25203177          143            DATA    025203177
00221                    144  ARG       BSS     2
00223                    145  CNTR      BSS     1
00224                    146  E         BSS     1
00225                    147  EXIT      BSS     1
00226                    148  R         BSS     2
00230                    149  S         BSS     1
00231                    150  TX        BSS     1
00232  00000000          151  ZERO      DATA    0
00233  00000001          152  ONE       DATA    01
00234  40000000          153  SIGN      DATA    040000000
00235  77777777          154  ONES      DATA    -1
00236  00037777          155  ADDR      DATA    037777
       00000142          156  CC        EQU     K1
                         157            END
```

IDENTIFICATION: ARC TAN of A - ATN

AUTHOR: W. S. LaSor, SDS

ACCEPTED: 14 February 1963

COMPUTER CONFIGURATION: Any 920/930 Computer.

PURPOSE: To compute arctan $\frac{y}{x}$ where y and x are numbers in the A and B registers respectively.

PROGRAMMED OPERATORS: None

STORAGE: Instructions and constants: 71 oct, 57 dec
Used standard subroutine temporary locations 12 thru 17.

TIMING: 1032 μsec

SOURCE LANGUAGE: SYMBOL

USE:

1. FUNCTION

The arctangent of the variable determined by the contents of A divided by the contents of B replaces the contents of A. The address field of ATN is not used.

2. ARGUMENTS

The numberator in A and the denominator in B may be at any scaling as long as the scaling is identical. On exit, the result is in A in fractions of a circle scaled at Q = -1 (see program description for SIN for examples). The result is not restricted to principal values, but may take on any value between -180° and 179.99998°. Thus, output from ATN is entirely compatible for input to SIN or COS.

METHOD:

The relative magnitudes of the two arguments are first determined. If the absolute value of the contents of A is less than the absolute value of the contents of B, then $u = \frac{(A)}{(B)}$ is found and the arctangent of u computed. If the relative magnitudes are not as stated above, then $u = \frac{(B)}{(A)}$ is found and the arccotangent of u is computed. The answer at this point is in either the first or fourth quadrant. The sign of the denominator of the input variable is then inspected and, if negative, 180 degrees are added to the result to obtain the final answer.

The following approximation is used:

$$\arctan u = \sum_{1}^{7} C_{2k-1} U^{2k-1}$$

where:

$C_1 = .15914533$
$C_3 = -.05302625$
$C_5 = .03152520$
$C_7 = -.02106178$
$C_9 = .01267292$
$C_{11} = -.00534860$
$C_{13} = .00108423$

Maximum absolute error is less than 10^{-6}.

```
                                1  $ATN    POPD    012500000
00000  0 37 00014               2  ATN     STX     TX              SAVE INDEX
00001  0 35 00016               3          STA     YS              SAVE SIGN OF Y
00002  0 72 00070               4          SKA     SIGN            GET ABSV(Y)
00003  0 46 01000               5          CNA
00004  0 35 00017               6          STA     Y
00005  0 46 00010               7          CBA
00006  0 35 00015               8          STA     XS              SAVE SIGN OF X
00007  0 17 00016               9          EOR     YS              GET SIGN OF RESULT
00010  0 35 00016              10          STA     RS
00011  0 46 10012              11          BAC
00012  0 72 00070              12          SKA     SIGN            GET ABSV(X)
00013  0 46 01000              13          CNA
00014  0 73 00017              14          SKG     Y               FIND LARGER ELEMENT
00015  4 01 00020              15          BRU     $+3,4
00016  4 51 00016              16          BRR     $,4             SET FLAG
00017  0 62 00017              17          XMA     Y               EXCHANGE X AND Y
00020  0 66 00001              18          RSH     1
00021  0 65 00017              19          DIV     Y               FORM X/Y OR Y/X AT 1
00022  0 35 00012              20          STA     ARG
00023  0 64 00012              21          MUL     ARG
00024  0 67 00001              22          LSH     1               ARG SQUARED AT 1
00025  0 35 00013              23          STA     ARGSQ
00026  4 64 00067              24          MUL     C13,4           EVAL POLYNOMIAL
00027  4 55 00066              25          ADD     C11,4
00030  0 64 00013              26          MUL     ARGSQ
00031  4 55 00065              27          ADD     C9,4
00032  0 64 00013              28          MUL     ARGSQ
00033  4 55 00064              29          ADD     C7,4
00034  0 64 00013              30          MUL     ARGSQ
00035  4 55 00063              31          ADD     C5,4
00036  0 64 00013              32          MUL     ARGSQ
00037  4 55 00062              33          ADD     C3,4
00040  0 64 00013              34          MUL     ARGSQ
00041  4 55 00061              35          ADD     C1,4
00042  0 64 00012              36          MUL     ARG             ARCTAN(U) AT 0
00043  0 67 00001              37          LSH     1               SCALE AT -1
00044  0 40 20001              38          OVT                     TEST FLAG
```

```
00045  4 01 00050        39          BRU     $+3,4
00046  4 54 00060        40          SUB     P90,4          GET 90-ARCTAN
00047  0 46 01000        41          CNA
00050  0 53 00015        42          SKN     XS             TEST SIGN OF X
00051  4 01 00053        43          BRU     $+2,4
00052  0 17 00070        44          EOR     SIGN           PUT IN 3RD QUAD.
00053  0 71 00014        45          LDX     TX             RESTORE INDEX
00054  0 53 00016        46          SKN     RS             AFFIX SIGN TO RESULT
00055  0 51 00000        47          BRR     0
00056  0 46 01000        48          CNA
00057  0 51 00000        49          BRR     0
00060  20000000          50  P90     DATA    1*/(23-1)
00061  12137126          51  C1      DATA    012137126
00062  71154676          52  C3      DATA    071154676
00063  10044045          53  C5      DATA    010044045
00064  65156617          54  C7      DATA    065156617
00065  14764206          55  C9      DATA    014764206
00066  65027452          56  C11     DATA    065027452
00067  04341626          57  C13     DATA    04341626
       00000012          58  ARG     EQU     012
       00000013          59  ARGSQ   EQU     013
       00000014          60  TX      EQU     014
       00000015          61  XS      EQU     015
       00000016          62  YS      EQU     016
       00000017          63  Y       EQU     017
00070  40000000          64  SIGN    DATA    040000000
       00000016          65  RS      EQU     YS
                         66          END
```

IDENTIFICATION: ARC TAN, Double Precision - ATD

AUTHOR: W. S. LaSor, SDS

ACCEPTED: 26 April 1963

COMPUTER CONFIGURATION: Any SDS 920/930 Computer

PURPOSE: To compute the double precision arctangent of the ratio of two double precision arguments; one in the A, B registers and the other in memory.

PROGRAMMED OPERATORS: DPN, DPD, DPM

STORAGE: Instructions and constants: 236 oct, 158 dec

Uses temporary storage locations 20 thru 22.

TIMING: 5.0 - 5.5 milliseconds

SOURCE LANGUAGE: SYMBOL

USE:

1. FUNCTION

The double precision arctangent of the contents of A, B divided by the contents of M+1, M replaces the contents of A, B. The contents of X are unchanged.

2. ARGUMENT

Both arguments must be in standard double precision format and identically scaled. The argument in memory is addressed by the ATD programmed operator. Thus:

ATD M

computes arctan $\left[\frac{(A,B)}{(M+1,M)}\right]$ and puts the result in A, B. The answer is in radians and lies in the range:

$$-\pi \leq \arctan \leq \pi$$

METHOD:

The sign of the result is first found by an exclusive OR between the signs of both arguments. The arguments are then changed to absolute value and the magnitudes inspected If $|(A, B)| \leq |(M+1, M)|$, $U = \frac{|(A, B)|}{|(M+1, M)|}$ is formed and the Arctan U computed; if $|(A, B)| > |(M+1, M)|$, $U = \frac{|(M+1, M)|}{|(A, B)|}$ is found and the Arccot U = Arctan 1/U is calculated.

The Arctan U is computed from the series:

$$\tan^{-1} U = U - 1/3\ U^3 + 1/5\ U^5 - 1/7\ U^7$$

$$-1/16 \leq U \leq 1/16$$

If $U > 1/16$, the reduction formula:

$$\tan^{-1} U = \tan^{-1}\left[\frac{U-X_i}{1+UX_i}\right] + \tan^{-1} X_i$$

is used where X_i is determined by the following table:

i	Range	X_i	$\tan^{-1} X_i$
1	$1/16 \leq U < 3/16$	1/8	.124, 354, 994, 547
2	$3/16 \leq U < 5/16$	1/4	.244, 978, 663, 127
3	$5/16 \leq U < 7/16$	3/8	.358, 770, 670, 271
4	$7/16 \leq U < 9/16$	1/2	.463, 647, 609, 001
5	$9/16 \leq U < 11/16$	5/8	.558, 599, 315, 344
6	$11/16 \leq U < 13/16$	3/4	.643, 501, 108, 793
7	$13/16 \leq U < 15/16$	7/8	.718, 829, 999, 622
8	$U > 15/16$	1	.785, 398, 163, 397

The sign of the result is then affixed and the answer placed in the proper quadrant according to the sign of the operand in memory.

The maximum absolute error does not exceed 10^{-12} in magnitude.

```
                                    1  $ATD    POPD    012500000
00000   0 37 00020                  2  ATD     STX     TX                  SAVE INDEX
00001   0 71 00000                  3          LDX     0                   SAVE RETURN
00002   0 37 00021                  4          STX     EXIT
00003   4 35 00225                  5          STA     YS,4                SAVE SIGN OF Y
00004   0 72 00234                  6          SKA     SIGN                GET ABSV(Y)
I 00005 1 03 00000                  7          DPN
00006   4 35 00233                  8          STA     Y+1,4
00007   4 36 00232                  9          STB     Y,4
00010   0 71 00020                 10          LDX     TX
00011   0 77 40021                 11          EAX     *EXIT
00012   2 76 00001                 12          LDA     1,2                 LOAD X
00013   2 75 00000                 13          LDB     0,2
00014   4 35 00224                 14          STA     XS,4                SAVE SIGN OF X
00015   4 17 00225                 15          ESR     YS,4                FORM SIGN OF ANSWER
00016   4 35 00225                 16          STA     RS,4
00017   2 76 00001                 17          LDA     1,2                 RECALL MSW OF X
00020   0 72 00234                 18          SKA     SIGN                GET ABSV(X)
I 00021 1 03 00000                 19          DPN
00022   0 71 00235                 20          LDX     ONES
00023   4 73 00233                 21          SKG     Y+1,4               FIND LARGER ELEMENT
00024   4 01 00031                 22          BRU     $+5,4
00025   4 62 00233                 23          XMA     Y+1,4               X GREATER THAN Y
00026   0 46 00014                 24          XAB
00027   4 62 00232                 25          XMA     Y,4                 EXCHANGE X AND Y
00030   2 46 00014                 26          RCH     014,2
00031   0 37 00022                 27          STX     FLAG                SET FLAG ACCORDINGLY
00032   0 66 00001                 28          RSH     1
I 00033 5 01 00232                 29          DPD     Y,4                 FORM X/Y OR Y/X AT 1
00034   4 35 00227                 30          STA     ARG+1,4             RESULT=U
00035   4 71 00155                 31          LDX     189,4               LOCATE INTERVAL
00036   4 54 00154                 32          SUB     185,4
00037   0 72 00234                 33          SKA     SIGN
00040   4 01 00151                 34          BRU     ATD1,4              INTERVAL 0
00041   4 41 00042                 35          BRX     $+1,4
00042   4 54 00172                 36          SUB     184,4
00043   0 72 00234                 37          SKA     SIGN
00044   4 01 00046                 38          BRU     $+2,4               JUMP ON ITH INTERVAL
```

```
00045  4 41 00042      39          BRX     $-3,4
00046  4 76 00227      40          LDA     ARG+1,4          RECALL U
00047  4 36 00226      41          STB     ARG,4
00050  6 64 00171      42          MUL     TABLE-1,6        COMPUTE U*X[I]
00051  4 35 00231      43          STA     ARGSQ+1,4
00052  4 36 00230      44          STB     ARGSQ,4
00053  4 76 00226      45          LDA     ARG,4
00054  0 66 20001      46          RCY     1                MAKE POSITIVE
00055  6 64 00171      47          MUL     TABLE-1,6
00056  0 67 20001      48          LCY     1
00057  4 55 00230      49          ADD     ARGSQ,4
00060  0 46 20005      50          ABC
00061  4 57 00231      51          ADC     ARGSQ+1,4
00062  4 55 00175      52          ADD     182,4            1+U*X[I] AT 2
00063  4 62 00227      53          XMA     ARG+1,4          RECALL U
00064  0 46 00014      54          XAB
00065  4 62 00226      55          XMA     ARG,4
00066  0 46 00014      56          XAB
00067  6 54 00171      57          SUB     TABLE-1,6        U-X[I] AT 1
I 00070  5 01 00226      58          DPD     ARG,4            (U-X[I])/(1+U*X[I])
00071  0 66 00001      59          RSH     1                SCALE AT 0
00072  4 35 00227      60  EVAL    STA     ARG+1,4
00073  4 36 00226      61          STB     ARG,4
I 00074  5 02 00226      62          DPM     ARG,4            ARG SQUARED AT 0
00075  4 35 00231      63          STA     ARGSQ+1,4
00076  4 36 00230      64          STB     ARGSQ,4
I 00077  5 0[illegible] 00170      65          DPM     C9,4             EVAL POLYNOMIAL
00100  0 46 00014      66          XAB
00101  4 55 00166      67          ADD     C7,4
00102  0 46 00014      68          XAB
00103  4 57 00167      69          ADC     C7+1,4
I 00104  5 02 00230      70          DPM     ARGSQ,4
00105  0 46 00014      71          XAB
00106  4 55 00164      72          ADD     C5,4
00107  0 46 00014      73          XAB
00110  4 57 00165      74          ADC     C5+1,4
I 00111  5 02 00230      75          DPM     ARGSQ,4
00112  0 46 00014      76          XAB
00113  4 55 00162      77          ADD     C3,4
```

```
00114  0 46 00014      78          XAB
00115  4 57 00163      79          ADC     C3+1,4
I 00116  5 02 00230    80          DPM     ARGSQ,4
00117  0 66 00002      81          RSH     2                   SCALE AT 2
00120  4 55 00175      82          ADD     182,4
I 00121  5 02 00226    83          DPM     ARG,4               ARCTAN(U) AT 2
00122  0 46 00014      84          XAB
00123  6 55 00202      85          ADD     ATN,6               ADD ATN(X(I))
00124  0 46 00014      86          XAB
00125  6 57 00213      87          ADC     ATN+9,6
00126  0 53 00022      88          SKN     FLAG                TEST FLAG
00127  4 01 00136      89          BRU     $+7,4
00130  0 46 00014      90          XAB                         GET ARCCOTANGENT
00131  4 54 00156      91          SUB     PI2,4
00132  0 46 01000      92          CNA
00133  0 46 00014      93          XAB
00134  4 56 00156      94          SUC     PI2,4
00135  0 17 00235      95          EOR     ONES
00136  4 53 00224      96          SKN     XS,4                TEST SIGN OF X
00137  4 01 00144      97          BRU     $+5,4
00140  0 46 00014      98          XAB                         PUT IN 3RD QUAD.
00141  4 54 00160      99          SUB     PI,4
00142  0 46 00014     100          XAB
00143  4 56 00161     101          SUC     PI+1,4
00144  0 71 00020     102          LDX     TX                  RESTORE INDEX
00145  4 53 00225     103          SKN     RS,4                AFFIX SIGN TO RESULT
00146  0 51 00021     104          BRR     EXIT
I 00147  1 03 00000    105          DPN
00150  0 51 00021     106          BRR     EXIT
00151  4 76 00227     107  ATD1    LDA     ARG+1,4
00152  0 67 00001     108          LSH     1
00153  4 01 00072     109          BRU     EVAL,4
00154  01000000       110  185     DATA    1*/(23-5)
00155  00040000       111  189     DATA    1*/(23-9)
00156  52104130       112  PI2     DED     1.570796326795*/(47-2)
00157  14441766
00160  24210220       113  PI      DED     3.141592653589*/(47-2)
00161  31103755
00162  25252610       114  C3      DED     -.333333333333*/(47-0)
```

```
00163  65252525
00164  46314632        115  C5      DED     .2*/(47-0)
00165  06314631
00166  33333362        116  C7      DED     -.142857142857*/(47-0)
00167  73333333
00170  43434322        117  C9      DED     .111111111111*/(47-0)
00171  03434343
00172  02000000        118  TABLE   DATA    002000000           .125*/(23-1)
00173  04000000        119          DATA    004000000           .25*/(23-1)
00174  06000000        120          DATA    006000000           .375*/(23-1)
00175  10000000        121  1S2     DATA    010000000           .5*/(23-1)
00176  12000000        122           DATA   012000000           .625*/(23-1)
00177  14000000        123          DATA    014000000           .75*/(23-1)
00200  16000000        124          DATA    016000000           .875*/(23-1)
00201  20000000        125          DATA    1*/(23-1)
00202  00000000        126  ATN     DATA    0
00203  24652664        127          DATA    024652664
00204  37445570        128          DATA    037445570
00205  50171302        129          DATA    050171302
00206  40530376        130          DATA    040530376
00207  52757434        131          DATA    052757434
00210  50623236        132          DATA    050623236
00211  61274212        133          DATA    061274212
00212  25042074        134          DATA    025042074
00213  00000000        135          DATA    0
00214  00775267        136          DATA    0775267
00215  01753335        137          DATA    01753335
00216  02675414        138          DATA    02675414
00217  03553063        139          DATA    03553063
00220  04360013        140          DATA    04360013
00221  05113617        141          DATA    05113617
00222  05600247        142          DATA    05600247
00223  06220773        143          DATA    06220773
00224                  144  XS      BSS     1
00225                  145  YS      BSS     1
00226                  146  ARG     BSS     2
00230                  147  ARGSQ   BSS     2
00232                  148  Y       BSS     2
       00000020        149  TX      EQU     020
       00000021        150  EXIT    EQU     021
       00000022        151  FLAG    EQU     022
00234  40000000        152  SIGN    DATA    040000000
00235  77777777        153  ONES    DATA    -1
       00000172        154  1B4     EQU     TABLE
       00000225        155  RS      EQU     YS
                       156          END
```

IDENTIFICATION: ARC TAN, Floating Point - ATF

AUTHOR: W. S. LaSor, SDS

ACCEPTED: 17 January 1964

COMPUTER CONFIGURATION: Any SDS 920/930 Computer

PURPOSE: To compute the floating point artangent of the ratio of two floating point arguments; one in the A, B registers and the other in memory.

PROGRAMMED OPERATORS: DPM, FLD, FLN

STORAGE: Instructions and constants: 371 oct, 249 dec

TIMING: 4.9 - 6.4 m.s.

SOURCE LANGUAGE: SYMBOL

USE:

1. FUNCTION

The floating point arctangent of the contents of A, B divided by the contents of M+1, M replaces the contents of A, B. The contents of X are unchanged.

2. ARGUMENT

Both arguments are in standard double precision floating point format. The argument in memory is addressed by the ATF programmed operator. Thus:

ATF M

computes arctan $\left[\frac{(A, B)}{(M+1, M)}\right]$ and puts the result in A, B.

The answer is in radians and lies in the range:

$$-\pi \leq \arctan < \pi$$

METHOD:

The sign of the result is first found by an exclusive OR between the signs of both arguments. The arguments are then changed to absolute value and the magnitudes inspected. If $|(A, B)| \leq |(M+1, M)|$, $U = \frac{|(A, B)|}{|(M+1, M)|}$ is formed and the Arctan U computed; if $|(A, B)| > |(M+1, M)|$, $U = \frac{|(M+1, M)|}{|(A, B)|}$ is found and the Arccot U = Arctan 1/U is calculated.

The Arctan U is computed from the series:

$$\tan^{-1} U = U - 1/3\, U^3 + 1/5\, U^5 - 1/7\, U^7$$

$$-1/16 \leq U \leq 1/16$$

If $U > 1/16$, the reduction formula:

$$\tan^{-1} U = \tan^{-1}\left[\frac{U-X_i}{1+UX_i}\right] + \tan^{-1} X_i$$

is used where X_i is determined by the following table:

i	Range	X_i	$\tan^{-1} X_i$
1	$1/16 \leq U < 3/16$	1/8	.124, 354, 994, 547
2	$3/16 \leq U < 5/16$	1/4	.244, 978, 663, 127
3	$5/16 \leq U < 7/16$	3/8	.358, 770, 670, 271
4	$7/16 \leq U < 9/16$	1/2	.463, 647, 609, 001
5	$9/16 \leq U < 11/16$	5/8	.558, 599, 315, 344
6	$11/16 \leq U < 13/16$	3/4	.643, 501, 108, 793
7	$13/16 \leq U < 15/16$	7/8	.718, 829, 999, 622
8	$U \geq 15/16$	1	.785, 398, 163, 397

The sign of the result is then affixed and the answer placed in the proper quadrant according to the sign of the operand in memory.

The maximum relative error does not exceed 5×10^{-11} in magnitude.

```
                                    1  $ATF    P0PD    012500000
00000   0 37 00360                  2  ATF     STX     TX              SAVE INDEX
00001   0 71 00000                  3          LDX     0               SAVE RETURN
00002   0 37 00357                  4          STX     EXIT
00003   0 35 00364                  5          STA     YS              SAVE SIGN OF Y
00004   0 72 00367                  6          SKA     SIGN            GET ABSV(Y)
I 00005 1 03 00000                  7          FLN
00006   4 35 00317                  8          STA     RS,4
00007   4 35 00363                  9          STA     Y+1,4
00010   4 36 00362                 10          STB     Y,4
00011   0 71 00360                 11          LDX     TX
00012   0 77 40357                 12          EAX     *EXIT
00013   0 50 00365                 13          SKE     ZERO            TEST Y FOR 0
00014   4 01 00022                 14          BRU     $+6,4
00015   2 53 00001                 15          SKN     1,2             TEST SIGN OF X
00016   4 01 00165                 16          BRU     LOADE+1,4
00017   4 76 00352                 17          LDA     PI+1,4          SET ANSWER TO -PI
00020   4 75 00351                 18          LDB     PI,4
00021   4 01 00162                 19          BRU     LOADE-2,4       LOAD EXPON AND EXIT
00022   2 76 00001                 20          LDA     1,2             LOAD X
00023   2 75 00000                 21          LDB     0,2
00024   0 35 00361                 22          STA     XS              SAVE SIGN OF X
00025   0 17 00364                 23          EOR     YS              GET SIGN OF ANSWER
00026   4 35 00317                 24          STA     RS,4
00027   2 76 00001                 25          LDA     1,2             RECALL MSW OF X
00030   0 72 00367                 26          SKA     SIGN            GET ABSV(X)
I 00031 1 03 00000                 27          FLN
00032   0 50 00365                 28          SKE     ZERO            TEST X FOR 0
00033   4 01 00041                 29          BRU     $+6,4
00034   4 76 00352                 30          LDA     PI+1,4          SET ANSWER TO -PI/2
00035   4 75 00351                 31          LDB     PI,4
I 00036 1 03 00000                 32          FLN
00037   0 77 00001                 33          EAX     1               SET EXPONENT TO 1
00040   4 01 00164                 34          BRU     LOADE,4         LOAD EXPON AND EXIT
00041   4 74 00362                 35          SKD     Y,4             FIND LARGER ELEMENT
00042   4 01 00265                 36          BRU     XCHNG,4
00043   0 71 00370                 37          LDX     ONES            Y GREATER
00044   0 37 00364                 38  SET     STX     FLAG            SET FLAG ACCORDINGLY
```

```
00045  5 02 00362        39        FLD     Y,4                 FORM X/Y OR Y/X
00046  0 46 00122        40        STE                         EXTRACT EXPONENT
00047  4 35 00354        41        STA     ARG+1,4             RESULT = U
00050  4 36 00353        42        STB     ARG,4
00051  4 37 00313        43        STX     EXP,4               SAVE EXPONENT OF U
00052  0 46 00600        44        XXA
00053  0 73 00365        45        SKG     ZERO                SCALE U AT 1
00054  4 01 00057        46        BRU     $+3,4
00055  0 46 00600        47        XXA
00056  4 01 00062        48        BRU     $+4,4
00057  0 46 01000        49        CNA                         SET SHIFT COUNT
00060  0 46 00600        50        XXA
00061  2 66 00001        51        RSH     1,2
00062  4 35 00356        52        STA     ARGSQ+1,4           SAVE SCALED ARG
00063  4 36 00355        53        STB     ARGSQ,4
00064  4 71 00307        54        FLY     189,4               LOCATE INTERVAL
00065  4 54 00305        55        SU      185,4
00066  0 72 00367        56        SKA     SIGN
00067  4 01 00172        57        BRU     ATF1,4              SMALL ARG CASE
00070  4 54 00302        58        SUB     184,4
00071  0 72 00367        59        SKA     SIGN
00072  4 01 00075        60        BRU     $+3,4
00073  4 41 00074        61        BRX     $+1,4
00074  4 41 00070        62        BRX     $-4,4
00075  4 76 00356        63        LDA     ARGSQ+1,4           RECALL SCALED ARG
00076  4 37 00315        64        STX     INTRVL,4            SAVE INTERVAL
00077  4 36 00353        65        STB     ARG,4               SAVE LSH OF U
00100  6 64 00302        66        MUL     TABLE+1,6
00101  4 62 00353        67        XMA     ARG,4               RECALL LSH OF ARG
00102  4 36 00354        68        STB     ARG+1,4
00103  0 66 20001        69        RCY     1                   MAKE POSITIVE
00104  6 64 00302        70        MUL     TABLE+1,6
00105  0 67 20001        71        LCY     1
00106  4 55 00354        72        ADD     ARG+1,4
00107  0 46 20005        73        ABC
00110  4 57 00353        74        ADC     ARG,4
00111  4 55 00310        75        ADD     182,4               1+U*X[I]
00112  0 77 00002        76        EAX     2
00113  0 67 10001        77        NOD     1                   FLOAT RESULT
```

```
00114  0 46 00140      78          LDE
00115  4 62 00356      79          XMA     ARGSQ+1,4
00116  0 46 00014      80          XAB
00117  4 62 00355      81          XMA     ARGSQ,4
00120  4 71 00315      82          LDX     INTRVL,4
00121  0 46 00014      83          XAB
00122  6 54 00302      84          SUB     TABLE+1,6
00123  0 71 00366      85          LDX     ONE
00124  0 67 10050      86          NOD     40                  FLOAT U-X[I]
00125  0 46 00140      87          LDE
I 00126  5 02 00355      88          FLD     ARGSQ,4             [U-X[I]]/[1+U*X[I]]
00127  0 46 00122      89          STE
00130  0 46 00600      90          XXA
00131  4 43 00214      91          BRM     ATAN,4              ATAN[REDUCED ARG]
00132  0 46 00600      92          XXA
00133  0 46 01000      93          CNA
00134  0 46 00600      94          XXA
00135  2 66 00001      95          RSH     1,2                 SCALE AT 1
00136  4 71 00315      96          LDX     INTRVL,4
00137  0 46 00014      97          XAB
00140  6 55 00321      98          ADD     ATN,6               ADD ATN [X[I]]
00141  0 46 00014      99          XAB
00142  6 57 00322     100          ADC     ATN+1,6
00143  0 53 00364     101  TESTF   SKN     FLAG
00144  4 01 00153     102          BRU     $+7,4
00145  0 46 00014     103          XAB
00146  4 55 00351     104          ADD     PI,4                GET ARCCOTANGENT
00147  0 46 01000     105          CNA
00150  0 46 00014     106          XAB
00151  4 57 00352     107          ADC     PI+1,4
00152  0 17 00370     108          EOR     ONES
00153  0 66 00001     109          RSH     1                   SCALE AT 2
00154  0 53 00361     110  TESTXS  SKN     XS
00155  4 01 00162     111          BRU     $+5,4
00156  0 46 00014     112          XAB                         PUT IN 3RD QUAD.
00157  4 55 00351     113          ADD     PI,4
00160  0 46 00014     114          XAB
00161  4 57 00352     115          ADC     PI+1,4
00162  0 77 00002     116          EAX     2
```

```
00163  0 67 10014     117          NOD     12
00164  0 46 00140     118  LOADE   LDE                      PACK EXPONENT
00165  0 71 00360     119          LDX     TX               RESTORE INDEX
00166  4 53 00317     120          SKN     RS,4             AFFIX SIGN TO RESULT
00167  0 51 00357     121          BRR     EXIT
00170  1 03 00000     122          FLN                      PUT IN RIGHT QUAD.
00171  0 51 00357     123          BRR     EXIT
00172  4 76 00313     124  ATF1    LDA     EXP,4
00173  4 75 00353     125          LDB     ARG,4
00174  4 71 00354     126          LDX     ARG+1,4
00175  4 43 00214     127          BRM     ATAN,4           GET ARCTAN
00176  0 53 00364     128          SKN     FLAG
00177  4 01 00205     129          BRU     $+6,4
00200  0 46 00600     130          XXA
00201  0 46 01000     131          CNA
00202  0 46 00600     132          XXA
00203  2 66 00001     133          RSH     1,2              SCALE AT 1
00204  4 01 00145     134          BRU     TESTF+2,4        EXIT THRU MAIN LINK
00205  0 53 00361     135          SKN     XS               TEST SIGN OF X
00206  4 01 00164     136          BRU     LOADE,4
00207  0 46 00600     137          XXA
00210  0 46 01000     138          CNA
00211  0 46 00600     139          XXA
00212  2 66 00002     140          RSH     2,2              SCALE AT 2
00213  4 01 00156     141          BRU     TESTXS+2,4       EXIT THRU MAIN LINK
00214  0 00 00000     142  ATAN    PZE                      ARCTAN SUBROUTINE
00215  4 73 00311     143          SKG     M20,4
00216  4 01 00263     144          BRU     RETURN,4         EXIT IF ARG SMALL
00217  0 46 00600     145          XXA
00220  0 66 00001     146          RSH     1
00221  4 41 00222     147          BRX     $+1,4
00222  4 37 00313     148          STX     EXP,4
00223  4 35 00354     149          STA     ARG+1,4
00224  4 36 00353     150          STB     ARG,4
00225  0 46 00200     151          CXA
00226  4 55 00313     152          ADD     EXP,4
00227  0 46 01000     153          CNA
00230  0 46 00400     154          CAX
00231  4 76 00354     155          LDA     ARG+1,4
```

```
I·  00232  5 01 00353      156          DPM     ARG,4              SQUARE ARGUMENT
    00233  4 35 00356      157          STA     ARGSQ+1,4
    00234  4 36 00355      158          STB     ARGSQ,4
I   00235  5 01 00347      159          DPM     C7,4               EVAL POLYNOMIAL
    00236  2 66 00000      160          RSH     0,2
    00237  0 46 00014      161          XAB
    00240  4 55 00345      162          ADD     C5,4
    00241  0 46 00014      163          XAB
    00242  4 57 00346      164          ADC     C5+1,4
I   00243  5 01 00355      165          DPM     ARGSQ,4
    00244  2 66 00000      166          RSH     0,2
    00245  0 46 00014      167          XAB
    00246  4 55 00343      168          ADD     C3,4
    00247  0 46 00014      169          XAB
    00250  4 57 00344      170          ADC     C3+1,4
I   00251  5 01 00355      171          DPM     ARGSQ,4
    00252  2 66 00000      172          RSH     0,2
    00253  0 46 00014      173          XAB
    00254  4 55 00341      174          ADD     C1,4
    00255  0 46 00014      175          XAB
    00256  4 57 00342      176          ADC     C1+1,4
    00257  5 01 00353      177          DPM     ARG,4
    00260  4 71 00313      178          LDX     EXP,4
    00261  0 67 10003      179          NOD     3
    00262  4 51 00214      180          BRR     ATAN,4
    00263  0 46 00600      181  RETURN  XXA
    00264  4 51 00214      182          BRR     ATAN,4
    00265  0 46 00600      183  XCHNG   XXA
    00266  4 72 00303      184          SKA     0P377,4
    00267  4 01 00300      185          BRU     ATF2,4             X LARGER
    00270  0 46 00600      186          XXA
    00271  4 73 00363      187          SKG     Y+1,4
    00272  4 01 00043      188          BRU     SET-1,4            Y STILL LARGER
    00273  4 62 00363      189          XMA     Y+1,4              EXCHANGE X AND Y
    00274  0 46 00014      190          XAB
    00275  4 62 00362      191          XMA     Y,4
    00276  2 46 00014      192          RCH     014,2              SET FLAG POSITIVE
    00277  4 01 00044      193          BRU     SET,4
    00300  0 46 00600      194  ATF2    XXA
```

```
00301  4 01 00273      195  TABLE  BRU     $-6,4
00302  02000000        196  1E4    DATA    002000000            .125*/(23-1)
00303  00000377        197  8P377  DATA    0377
00304  04000000        198         DATA    004000000            .25*/(23-1)
00305  01000000        199  1E5    DATA    1*/(23-5)
00306  06000000        200         DATA    006000000            .375*/(23-1)
00307  00040000        201  1E9    DATA    1*/(23-9)
00310  10000000        202  1E2    DATA    010000000            .5*/(23-1)
00311  77777754        203  M20    DATA    -20
00312  12000000        204         DATA    012000000            .625*/(23-1)
00313                  205  EXP    BSS     1
00314  14000000        206         DATA    014000000            .75*/(23-1)
00315                  207  INTRVL BSS     1
00316  16000000        208         DATA    016000000            .875*/(23-1)
00317                  209  RS     BSS     1
00320  20000000        210         DATA    1*/(23-1)
00321  51525434        211  ATN    DED     .124354994547*/(47-1)
00322  01772556
00323  77113111        212         DED     .244978663127*/(47-1)
00324  03726672
00325  20362372        213         DED     .358770670271*/(47-1)
00326  05573031
00327  01260706        214         DED     .463647609001*/(47-1)
00330  07326147
00331  25737024        215         DED     .558599315344*/(47-1)
00332  10740027
00333  21446440        216         DED     .643501108793*/(47-1)
00334  12227437
00335  42570240        217         DED     .718829999622*/(47-1)
00336  13400517
00337  52104070        218         DED     .785398163397*/(47-1)
00340  14441766
00341  77777550        219  C1     DED     .999999999999*/(47-0)
00342  37777777
00343  25252610        220  C3     DED     -.333333333333*/(47-0)
00344  65252525
00345  46314632        221  C5     DED     .2*/(47-0)
00346  06314631
00347  33333362        222  C7     DED     -.142857142857*/(47-0)
```

```
00350  73333333
00351  53567520        223  PI      DED     -3.14159265359*/(47-2)
00352  46674022
00353                  224  ARG     BSS     2
00355                  225  ARGSQ   BSS     2
00357                  226  EXIT    BSS     1
00360                  227  TX      BSS     1
00361                  228  XS      BSS     1
00362                  229  Y       BSS     2
00364                  230  YS      BSS     1
00365  00000000        231  ZERO    DATA    0
00366  00000001        232  ONE     DATA    01
00367  40000000        233  SIGN    DATA    040000000
00370  77777777        234  ONES    DATA    -1
       00000364        235  FLAG    EQU     YS
                       236          END
```

IDENTIFICATION: COSINE, Double Precision - CSD

AUTHOR: W. S. LaSor, SDS

ACCEPTED: 26 April 1963

COMPUTER CONFIGURATION: Any SDS 920/930 Computer

PURPOSE: To compute the double precision cosine of a double precision argument in the A, B registers.

PROGRAMMED OPERATORS: SND (CSD, SND are contained in the same subroutine)

STORAGE: Instructions and constants: 224 oct, 148 dec

Uses temporary storage locations 13 thru 17.

TIMING: Argument in circles: 3.59 - 3.74 m.s.

Argument in radians or degrees: 3.65 - 3.86 m.s.

SOURCE LANGUAGE: SYMBOL

USE:

1. FUNCTION

The double precision fixed-point cosine of the contents of A, B replaces the contents of A, B. The contents of X are unchanged.

2. ARGUMENT

See Program Description for SND.

METHOD: Use is made of the identity:

$$\cos X \equiv \sin (X + \frac{\pi}{2})$$

by adding 1/4 to the argument. The subroutine then exits through SND.

IDENTIFICATION: SINE, Double Precision - SND

AUTHOR: W. S. LaSor, SDS

ACCEPTED: 26 April 1963

COMPUTER CONFIGURATION: Any SDS 920/930 Computer

PURPOSE: To compute the double precision sine of a double precision argument in the A, B registers.

PROGRAMMED OPERATORS: DPM

STORAGE: Instructions and constants: 224 oct, 148 dec

Uses temporary storage locations 13 thru 17.

TIMING: Argument in circles: 3.55 - 3.7 m.s.
Argument in radians or degrees: 3.6 - 3.81 m.s.

SOURCE LANGUAGE: SYMBOL

USE:

1. FUNCTION

The double precision fixed point sine of the contents of A, B replaces the contents of A, B. The contents of X are unchanged.

2. ARGUMENT

The argument in the A, B registers is in standard double precision format and the units may be radians, degrees, or fractions of a circle. The address field of the programmed operator is used to define the units of the input argument. The address used for each set of units together with the required scaling are listed below:

USE: (Cont.)

Address	Units	Input Scaling
00000	radians	2
00001	circles	-1
00002	degrees	8

METHOD: The argument is first reduced to the interval $\frac{-\pi}{4} \leq X \leq \frac{\pi}{4}$ and the result used to evaluate one of two approximations using the identities:

Range	Sine Identity
$-\pi \leq X < \frac{-3\pi}{4}$	$-\text{Sin}\ (X + \pi)$
$\frac{-3\pi}{4} \leq X < \frac{\pi}{4}$	$-\text{Cos}\ (X + \frac{\pi}{2})$
$\frac{-\pi}{4} \leq X < \frac{\pi}{4}$	Sin X
$\frac{\pi}{4} \leq X < \frac{3\pi}{4}$	$\text{Cos}\ (X - \frac{\pi}{2})$
$\frac{3\pi}{4} \leq X < \pi$	$-\text{Sin}\ (X - \pi)$

The approximations listed below are modified Taylor's series:

$$\sin \frac{\pi X}{4} = C_1X + C_3X^3 + C_5X^5 + C_7X^7 + C_9X^9$$

$$\cos \frac{\pi X}{4} = C_0 + C_2X^2 + C_4X^4 + C_6X^6 + C_8X^8 + C_{10}X^{10}$$

for $-1 \leq X \leq 1$, where:

$C_0 = .999,999,999,999,944$
$C_1 = .785,398,163,378,792$
$C_2 = -.308,425,137,530,000$
$C_3 = -.080,745,511,814,982$
$C_4 = .015,854,344,196,643$
$C_5 = .002,490,392,478,234$

METHOD: (Cont.)

$C_6 = -.000,325,991,685,657$

$C_7 = -.000,036,571,416,916$

$C_8 = .000,003,590,472,284$

$C_9 = .000,000,308,563,132$

$C_{10} = -.000,000,024,266,335$

The maximum absolute error does not exceed 10^{-12} in magnitude.

```
                                    1  $CSD   P0PD    012700000
00000  0 37 00015                   2  CSD    STX     TX                SAVE X
00001  4 43 00147                   3         BRM     CNVRT,4           CONVERT TO CIRCLES
00002  4 55 00165                   4         ADD     P90,4             ADD 90 DEGREES
00003  0 02 20001                   5         R0V
00004  4 01 00007                   6         BRU     SND+2,4           EXIT THRU SND
                                    7  $SND   P0PD    012600000
00005  0 37 00015                   8  SND    STX     TX                SAVE X
00006  4 43 00147                   9         BRM     CNVRT,4           CONVERT TO CIRCLES
00007  0 35 00017                  10         STA     ARG
00010  0 46 00022                  11         RCH     022               CBX + CLB
00011  0 67 20003                  12         LCY     3                 LOCATE SUBINTERVAL
00012  0 46 00060                  13         XXB
00013  0 76 00017                  14         LDA     ARG
00014  6 23 00015                  15         EXU     TABLE,6           GO TO PROPER SUBR.
00015  4 01 00033                  16  TABLE  BRU     INT3,4            BRANCH TABLE
00016  4 01 00102                  17         BRU     INT4,4
00017  4 01 00102                  18         BRU     INT4,4
00020  4 01 00025                  19         BRU     INT1,4
00021  4 01 00025                  20         BRU     INT1,4
00022  4 01 00070                  21         BRU     INT2,4
00023  4 01 00070                  22         BRU     INT2,4
00024  4 01 00033                  23         BRU     INT3,4
00025  0 46 00014                  24  INT1   XAB                       SUBTRACT 180 DEG.
00026  0 17 00223                  25         E0R     0NES              NEGATE
00027  0 55 00222                  26         ADD     0NE
00030  0 46 00014                  27         XAB
00031  4 17 00166                  28         E0R     K1,4
00032  0 57 00221                  29         ADC     ZER0
00033  0 67 00001                  30  INT3   LSH     1                 SCALE ARG AT 1
00034  0 35 00020                  31         STA     ARG+1
00035  0 36 00017                  32         STB     ARG
I      00036  1 02 00017           33         DPM     ARG               SQUARE ARGUMENT
00037  0 35 00022                  34         STA     ARGSQ+1
00040  0 36 00021                  35         STB     ARGSQ
I      00041  5 02 00211           36         DPM     C9,4              EVAL POLYNOMIAL
00042  0 46 00014                  37         XAB
00043  4 55 00205                  38         ADD     C7,4
```

```
      00044  0 46 00014       39          XAB
      00045  4 57 00206       40          ADC     C7+1,4
I     00046  1 02 00021       41          DPM     ARGSQ
      00047  0 46 00014       42          XAB
      00050  4 55 00201       43          ADD     C5,4
      00051  0 46 00014       44          XAB
      00052  4 57 00202       45          ADC     C5+1,4
I     00053  1 02 00021       46          DPM     ARGSQ
      00054  0 46 00014       47          XAB
      00055  4 55 00175       48          ADD     C3,4
      00056  0 46 00014       49          XAB
      00057  4 57 00176       50          ADC     C3+1,4
I     00060  1 02 00021       51          DPM     ARGSQ
      00061  0 46 00014       52          XAB
      00062  4 55 00171       53          ADD     C1,4
      00063  0 46 00014       54          XAB
      00064  4 57 00172       55          ADC     C1+1,4
I     00065  1 02 00017       56          DPM     ARG             SINE AT 1
      00066  0 71 00015       57          LDX     TX              RESTORE X
      00067  0 51 00016       58          BRR     EXIT
      00070  4 55 00165       59  INT2    ADD     P90,4           ADD 90 DEGREES
      00071  4 43 00106       60          BRM     COS,4           EVALUATE COSINE
      00072  0 46 00014       61          XAB                     NEGATE
      00073  0 46 01000       62          CNA
      00074  0 46 00014       63          XAB
      00075  0 52 00223       64          SKB     ONES
      00076  0 55 00222       65          ADD     ONE
      00077  0 46 01000       66          CNA
      00100  0 71 00015       67          LDX     TX              RESTORE X
      00101  0 51 00016       68          BRR     EXIT
      00102  4 54 00165       69  INT4    SUB     P90,4           SUBTRACT 90 DEGREES
      00103  4 43 00106       70          BRM     COS,4           EVALUATE COSINE
      00104  0 71 00015       71          LDX     TX              RESTORE X
      00105  0 51 00016       72          BRR     EXIT
      00106  0 00 00000       73  COS     PZE                     COSINE SUBROUTINE
      00107  0 67 00001       74          LSH     1               SCALE ARG AT 1
      00110  0 35 00020       75          STA     ARG+1
      00111  0 36 00017       76          STB     ARG
I     00112  1 02 00017       77          DPM     ARG             SQUARE ARG
```

```
00113  0 35 00022      78          STA     ARGSQ+1
00114  0 36 00021      79          STB     ARGSQ
I 00115  5 02 00213    80          DPM     C10,4           EVAL POLYNOMIAL
00116  0 46 00014      81          XAB
00117  4 55 00207      82          ADD     C8,4
00120  0 46 00014      83          XAB
00121  4 57 00210      84          ADC     C8+1,4
I 00122  1 02 00021    85          DPM     ARGSQ
00123  0 46 00014      86          XAB
00124  4 55 00203      87          ADD     C6,4
00125  0 46 00014      88          XAB
00126  4 57 00204      89          ADC     C6+1,4
I 00127  1 02 00021    90          DPM     ARGSQ
00130  0 46 00014      91          XAB
00131  4 55 00177      92          ADD     C4,4
00132  0 46 00014      93          XAB
00133  4 57 00200      94          ADC     C4+1,4
I 00134  1 02 00021    95          DPM     ARGSQ
00135  0 46 00014      96          XAB
00136  4 55 00173      97          ADD     C2,4
00137  0 46 00014      98          XAB
00140  4 57 00174      99          ADC     C2+1,4
I 00141  1 02 00021   100          DPM     ARGSQ
00142  0 46 00014     101          XAB
00143  4 55 00167     102          ADD     C0,4
00144  0 46 00014     103          XAB
00145  4 57 00170     104          ADC     C0+1,4          COSINE AT 1
00146  4 51 00106     105          BRR     COS,4
00147  0 00 00000     106  CNVRT   PZE                     UNITS CONVERSION SUBR.
00150  0 71 00000     107          LDX     0               SAVE RETURN
00151  0 37 00016     108          STX     EXIT
00152  0 71 00015     109          LDX     TX
00153  0 77 40000     110          EAX     *0              GET UNITS ADDRESS
00154  0 46 00600     111          XXA
00155  0 72 00222     112          SKA     ONE             TEST FOR CIRCLES
00156  4 01 00163     113          BRU     $+5,4
00157  0 46 00600     114          XXA
I 00160  7 02 00215   115          DPM     FACTOR,6
00161  0 67 20001     116          LCY     1               SCALE AT -1
```

```
00162  4 51 00147      117            BRR     CNVRT,4
00163  0 46 00600      118            XXA
00164  4 51 00147      119            BRR     CNVRT,4
00165  20000000        120  P90       DATA    1*/(23-1)
00166  37777777        121  K1        DATA    037777777
00167  77776470        122  C0        DED     .99999999999*/(47-1)
00170  17777777
00171  24203210        123  C1        DED     .785398163379*/(47-0)
00172  31103755
00173  66205550        124  C2        DED     -.308425137530*/(47+1)
00174  54205414
00175  15564550        125  C3        DED     -.080745511815*/(47+2)
00176  65524206
00177  01644416        126  C4        DED     .015854344196*/(47+3)
00200  04036037
00201  64375113        127  C5        DED     .002490392478*/(47+4)
00202  01214656
00203  21370074        128  C6        DED     -.000359916857*/(47+5)
00204  77525054
00205  67534165        129  C7        DED     -.000036571417*/(47+6)
00206  77731515
00207  17301301        130  C8        DED     .000003590472 3*/(47+7)
00210  00007417
00211  50446636        131  C9        DED     .000000308563*/(47+8)
00212  00001226
00213  61565706        132  C10       DED     -.000000242663*/(47+9)
00214  77777627
00215  55620520        133  FACTOR    DED     .159154943092*/(47+2)
00216  24276301
00217  26575270        134               DED     .0027777777777*/(47+8)
00220  26602660
       00000015        135  TX        EQU     13
       00000016        136  EXIT      EQU     14
       00000017        137  ARG       EQU     15
       00000021        138  ARGSQ     EQU     17
00221  00000000        139  ZERO      DATA    0
00222  00000001        140  ONE       DATA    01
00223  77777777        141  ONES      DATA    -1
                       142               END
```

IDENTIFICATION: COSINE, Floating - CSF

AUTHOR: W. S. LaSor, SDS

ACCEPTED: 17 January 1964

COMPUTER CONFIGURATION: Any SDS 920/930 Computer

PURPOSE: To compute the floating point cosine of the contents of the A, B registers.

PROGRAMMED OPERATORS: SNF (SNF and CSF are contained in the same subroutine)

STORAGE: Instructions and constants: 331 oct, 217 dec

TIMING: Argument in circles: 4.35 - 4.65 m.s.
Argument in radians or degress: 5.15 - 5.45 m.s.

SOURCE LANGUAGE: SYMBOL

USE:

1. FUNCTION

The double precision floating point cosine of the contents of A, B replaces the contents of A, B. The contents of X are unchanged.

2. ARGUMENT

See Program Description for SNF.

METHOD: Use is made of the identity:

$$\text{Cos } X \equiv \text{Sin } (X + \frac{\pi}{2})$$

by adding the appropriate constant to the argument. The subroutine then exits through SNF. The error statement for SNF applies to CSF as well.

IDENTIFICATION: SINE, Floating Point - SNF

AUTHOR: W. S. LaSor, SDS

ACCEPTED: 17 January 1964

COMPUTER CONFIGURATION: Any SDS 920/930 Computer

PURPOSE: To compute the floating point sine of the contents of the A, B registers.

PROGRAMMED OPERATORS: DPM, FLM

STORAGE: Instructions and constants: 331 oct, 217 dec

TIMING: Argument in Circles: 4.2 - 4.5 m.s.
Argument in Radians or Degrees: 5.0 - 5.3 m.s.

SOURCE LANGUAGE: SYMBOL

USE:

1. FUNCTION

The floating point sine of the contents of the A, B registers replaces the contents of A, B. The contents of X are unchanged.

2. ARGUMENT

The argument in the A, B registers is in standard double precision floating point format. The units may be radians, degrees, or fractions of a circle. The address field of the programmed operator is used to define the units as follows:

ADDRESS	UNITS
00000	radians
00001	fractions of a circle
00002	degrees

METHOD: The argument is first reduced to the range $-\pi \leq X < \pi$ (one period). The result is then further reduced to the interval $\frac{-\pi}{4} \leq X \leq \frac{-\pi}{4}$ and used to evaluate one of two approximations as determined by the following identities:

Range	Identity
$-\pi \leq X < \frac{-3\pi}{4}$	$-\text{Sin}\,(X + \pi)$
$\frac{-3\pi}{4} \leq X < \frac{-\pi}{4}$	$-\text{Cos}\,(X + \frac{\pi}{2})$
$\frac{-\pi}{4} \leq X < \frac{\pi}{4}$	$\text{Sin}\,X$
$\frac{\pi}{4} \leq X < \frac{3\pi}{4}$	$\text{Cos}\,(X - \frac{\pi}{2})$
$\frac{3\pi}{4} \leq X < \pi$	$-\text{Sin}\,(X - \pi)$

The approximations for sine and cosine are:

$$\left.\begin{aligned} \text{Sin}\,\frac{\pi X}{4} &= \sum_{0}^{4} C_{2k+1} X^{2k+1} \\ \text{Cos}\,\frac{\pi X}{4} &= \sum_{0}^{5} C_{2k} X^{2k} \end{aligned}\right\} \quad -1 \leq X \leq 1$$

where:

$C_0 = .999,999,999,999,944$
$C_1 = .785,398,163,378,792$
$C_2 = -.308,425,137,530,000$
$C_3 = -.080,745,511,814,982$
$C_4 = .015,854,344,196,643$
$C_5 = .002,490,392,478,234$
$C_6 = -.000,325,991,685,657$
$C_7 = -.000,036,571,416,916$
$C_8 = .000,003,590,472,284$
$C_9 = .000,000,308,563,132$
$C_{10} = -.000,000,024,266,335$

METHOD: (cont.) The maximum relative computational error, E_R, satisfies:

$$\left|E_R\right| = \left|\frac{\sin X - \sin^* X}{\sin X}\right| < 6 \times 10^{-11},$$

where $\sin^* X$ denotes the computed value of sin X. In addition, however, error arises from loss of significance in the argument as X increases and as X approaches zeros of sin X (cos X). This error is due to the periodic nature of the sine (cosine) function and not to the computational method. For arguments exceeding 2^{39} in absolute value, all significance vanishes and the value zero will be returned.

```
                                    1  $CSF   P0PD    012700000
00000  0 37 00324                   2  CSF    STX     TX              SAVE X
00001  0 71 00000                   3         LDX     0               SAVE RETURN
00002  0 37 00322                   4         STX     EXIT
00003  4 43 00243                   5         BRM     CNVRT,4         CONVERT TO CIRCLES
00004  0 74 00330                   6         SKD     ONES
00005  4 01 00011                   7         BRU     $+4,4
00006  0 46 00102                   8         RCH     0102            CLEAR EXPONENT
00007  2 66 00000                   9         RSH     0,2             SCALE RIGHT
00010  4 01 00013                  10         BRU     $+3,4
00011  0 46 00102                  11         RCH     0102
00012  2 67 00000                  12         LSH     0,2             SCALE LEFT
00013  4 55 00261                  13         ADD     P90,4           ADD 90 DEGREES
00014  0 71 00330                  14         LDX     ONES            SET EXPONENT
00015  0 46 00140                  15         LDE
00016  4 01 00024                  16         BRU     SNF+5,4         EXIT THRU SNF
                                   17  $SNF   P0PD    012600000
00017  0 37 00324                  18  SNF    STX     TX              SAVE X
00020  0 71 00000                  19         LDX     0               SAVE RETURN
00021  0 37 00322                  20         STX     EXIT
00022  4 43 00243                  21         BRM     CNVRT,4         CONVERT TO CIRCLES
00023  0 46 00120                  22         RCH     0120            EXTRACT EXPONENT
00024  4 37 00323                  23         STX     EXP,4
00025  0 74 00330                  24         SKD     ONES            UNFLOAT ARG
00026  4 01 00044                  25         BRU     LEFT,4          GO SCALE LEFT
00027  0 46 00102                  26         RCH     0102            CLEAR EXPONENT
00030  4 62 00323                  27         XMA     EXP,4
00031  4 55 00260                  28         ADD     P3,4
00032  4 62 00323                  29         XMA     EXP,4
00033  0 35 00317                  30         STA     ARG+1
00034  0 36 00316                  31         STB     ARG
00035  2 66 00000                  32         RSH     0,2             SCALE RIGHT
00036  0 35 00320                  33  SAVE   STA     TA
00037  0 46 00022                  34         RCH     022             CBX + CLB
00040  0 67 20003                  35         LCY     3               LOCATE SUBINTERVAL
00041  0 46 00060                  36         XXB
00042  0 76 00320                  37         LDA     TA
00043  6 23 00053                  38         EXU     TABLE,6         GO TO PROPER SUBR.
```

```
00044  0 46 00102      39  LEFT   RCH    0102          CLEAR EXPONENT
00045  2 67 00000      40         LSH    0,2           SCALE LEFT
00046  0 35 00317      41         STA    ARG+1
00047  0 36 00316      42         STB    ARG
00050  4 71 00257      43         LDX    P2,4          SET EXPONENT TO 2
00051  4 37 00323      44         STX    EXP,4
00052  4 01 00036      45         BRU    SAVE,4
00053  4 01 00107      46  TABLE  BRU    INT3,4        BRANCH TABLE
00054  4 01 00122      47         BRU    INT4,4
00055  4 01 00122      48         BRU    INT4,4
00056  4 01 00063      49         BRU    INT1,4
00057  4 01 00063      50         BRU    INT1,4
00060  4 01 00072      51         BRU    INT2,4
00061  4 01 00072      52         BRU    INT2,4
00062  4 01 00107      53         BRU    INT3,4
00063  0 46 00014      54  INT1   XAB                  SUBTRACT 180 DEG.
00064  0 17 00330      55         EOR    ONES          NEGATE
00065  0 55 00326      56         ADD    ONE
00066  0 46 00014      57         XAB
00067  4 17 00256      58         EOR    K1,4
00070  0 57 00325      59         ADC    ZERO
00071  4 01 00111      60         BRU    INT3+2,4      EXIT THRU SINE LINK
00072  4 55 00261      61  INT2   ADD    P90,4         ADD 90 DEGREES
00073  4 43 00177      62         BRM    COS,4         EVALUATE COSINE
00074  0 46 00014      63         XAB                  NEGATE
00075  0 46 01000      64         CNA
00076  0 46 00014      65         XAB
00077  0 52 00330      66         SKB    ONES
00100  0 55 00326      67         ADD    ONE
00101  0 46 01000      68         CNA
00102  0 71 00326      69         LDX    ONE
00103  0 67 10002      70         NOD    2             NORMALIZE RESULT
00104  0 46 00140      71         LDE
00105  0 71 00324      72         LDX    TX            RESTORE X
00106  0 51 00322      73         BRR    EXIT
00107  0 76 00317      74  INT3   LDA    ARG+1         RECALL ARGUMENT
00110  0 75 00316      75         LDB    ARG
00111  4 71 00323      76         LDX    EXP,4
00112  0 67 10030      77         NOD    24            NORMALIZE
```

```
00113  0 46 00600      78          XXA
00114  0 73 00325      79          SKG     ZERO                TEST EXP GTR 0
00115  4 01 00125      80          BRU     SINE,4              GO COMPUTE SINE
00116  4 76 00263      81          LDA     ANS+1,4             SET TO SQRT HALF
00117  4 75 00262      82          LDB     ANS,4
00120  0 71 00324      83          LDX     TX                  RESTORE X
00121  0 51 00322      84          BRR     EXIT
00122  4 54 00261      85  INT4    SUB     P90,4               SUBTRACT 90 DEGREES
00123  4 43 00177      86          BRM     COS,4               EVALUATE COSINE
00124  4 01 00102      87          BRU     INT2+8,4            FLOAT AND EXIT
00125  0 46 00600      88  SINE    XXA
00126  0 35 00317      89          STA     ARG+1               SAVE ARGUMENT
00127  0 36 00316      90          STB     ARG
00130  4 37 00323      91          STX     EXP,4
00131  0 46 00200      92          CXA
00132  4 55 00323      93          ADD     EXP,4               EXPONENT OF ARGSQ
00133  0 46 01000      94          CNA
00134  0 46 00400      95          CAX                         SCALE TO X
00135  0 76 00317      96          LDA     ARG+1
00136  0 02 20001      97          ROV
I      00137  1 02 00316      98          DPM     ARG                 FORM ARGSQ
00140  0 40 20001      99          OVT                         TEST FOR -1 AT 0
00141  4 01 00240     100          BRU     LOAD,4
00142  0 35 00321     101  RET     STA     ARGSQ+1
00143  0 36 00320     102          STB     ARGSQ
I      00144  5 02 00312     103          DPM     C9,4                EVAL POLYNOMIAL
00145  2 66 00000     104          RSH     0,2
00146  0 46 00014     105          XAB
00147  4 55 00306     106          ADD     C7,4
00150  0 46 00014     107          XAB
00151  4 57 00307     108          ADC     C7+1,4
I      00152  1 02 00320     109          DPM     ARGSQ
00153  2 66 00000     110          RSH     0,2
00154  0 46 00014     111          XAB
00155  4 55 00302     112          ADD     C5,4
00156  0 46 00014     113          XAB
00157  4 57 00303     114          ADC     C5+1,4
I      00160  1 02 00320     115          DPM     ARGSQ
00161  2 66 00000     116          RSH     0,2
```

```
          00162  0 46 00014      117            XAB
          00163  4 55 00276      118            ADD     C3,4
          00164  0 46 00014      119            XAB
          00165  4 57 00277      120            ADC     C3+1,4
     I    00166  1 02 00320      121            DPM     ARGSQ
          00167  2 66 00000      122            RSH     0,2
          00170  0 46 00014      123            XAB
          00171  4 55 00272      124            ADD     C1,4
          00172  0 46 00014      125            XAB
          00173  4 57 00273      126            ADC     C1+1,4
     I    00174  1 02 00316      127            DPM     ARG
          00175  4 71 00323      128            LDX     EXP,4
          00176  4 01 00103      129            BRU     INT2+9,4        FLOAT AND EXIT
          00177  0 00 00000      130  COS       PZE                     COSINE SUBROUTINE
          00200  0 67 00001      131            LSH     1
          00201  0 35 00317      132            STA     ARG+1
          00202  0 36 00316      133            STB     ARG
     I    00203  1 02 00316      134            DPM     ARG             SQUARE ARGUMENT
          00204  0 35 00321      135            STA     ARGSQ+1
          00205  0 36 00320      136            STB     ARGSQ
     I    00206  5 02 00314      137            DPM     C10,4           EVAL POLYNOMIAL
          00207  0 46 00014      138            XAB
          00210  4 55 00310      139            ADD     C8,4
          00211  0 46 00014      140            XAB
          00212  4 57 00311      141            ADC     C8+1,4
     I    00213  1 02 00320      142            DPM     ARGSQ
          00214  0 46 00014      143            XAB
          00215  4 55 00304      144            ADD     C6,4
          00216  0 46 00014      145            XAB
          00217  4 57 00305      146            ADC     C6+1,4
     I    00220  1 02 00320      147            DPM     ARGSQ
          00221  0 46 00014      148            XAB
          00222  4 55 00300      149            ADD     C4,4
          00223  0 46 00014      150            XAB
          00224  4 57 00301      151            ADC     C4+1,4
     I    00225  1 02 00320      152            DPM     ARGSQ
          00226  0 46 00014      153            XAB
          00227  4 55 00274      154            ADD     C2,4
          00230  0 46 00014      155            XAB
```

```
00231  4 57 00275      156           ADC     C2+1,4
00232  1 02 00320      157           DPM     ARGSQ
00233  0 46 00014      158           XAB
00234  4 55 00270      159           ADD     C0,4
00235  0 46 00014      160           XAB
00236  4 57 00271      161           ADC     C0+1,4              COSINE AT 1
00237  4 51 00177      162           BRR     COS,4
00240  4 76 00256      163  LOAD     LDA     K1,4                NEGATE
00241  0 75 00330      164           LDB     ONES
00242  4 01 00142      165           BRU     RET,4
00243  0 00 00000      166  CNVRT    PZE                         UNITS CONVERSION SUBR
00244  0 71 00324      167           LDX     TX
00245  0 77 40000      168           EAX     *0                  GET UNITS ADDRESS
00246  0 46 00600      169           XXA
00247  0 72 00326      170           SKA     ONE                 TEST FOR CIRCLES
00250  4 01 00254      171           BRU     $+4,4
00251  0 46 00600      172           XXA                         RADIANS OR DEGREES
00252  7 03 00264      173           FLM     FACTOR,6            CONVERT
00253  4 51 00243      174           BRR     CNVRT,4
00254  0 46 00600      175           XXA
00255  4 51 00243      176           BRR     CNVRT,4
00256  37777777        177  K1       DATA    037777777
00257  00000002        178  P2       DATA    2
00260  00000003        179  P3       DATA    3
00261  20000000        180  P90      DATA    1*/(23-1)
00262  46376000        181  ANS      DED     .707106781187
00263  26501171
00264  55623776        182  FACTOR   DED     .159154943092
00265  24276301
00266  26604770        183           DED     .0027777777778
00267  26602660
00270  77777664        184  C0       DED     .999999999999*/(47-1)
00271  17777777
00272  24203210        185  C1       DED     .785398163379*/(47-0)
00273  31103755
00274  66205540        186  C2       DED     -.30842513753*/(47+1)
00275  54205414
```

```
                                187             PAGE
00276   43335132                188     C3      DED     -.080745511815*/(47-0)
00277   75325041
00300   01644416                189     C4      DED     .015854344196*/(47+3)
00301   04036037
00302   73217644                190     C5      DED     .002490392478*/(47-0)
00303   00050632
00304   21370074                191     C6      DED     -.0003259916857*/(47+5)
00305   77525054
00306   15675342                192     C7      DED     -.000036571417*/(47-0)
00307   77777315
00310   17301301                193     C8      DED     .0000035904723*/(47+7)
00311   00007417
00312   45521115                194     C9      DED     .000000308563*/(47-0)
00313   00000002
00314   61565706                195     C10     DED     -.0000000242663*/(47+9)
00315   77777627
00316                           196     ARG     BSS     2
00320                           197     ARGSQ   BSS     2
00322                           198     EXIT    BSS     1
00323                           199     EXP     BSS     1
00324                           200     TX      BSS     1
        00000320                201     TA      EQU     ARGSQ
00325   00000000                202     ZERO    DATA    0
00326   00000001                203     ONE     DATA    01
00327   40000000                204     SIGN    DATA    040000000
00330   77777777                205     ONES    DATA    -1
                                206             END
```

IDENTIFICATION: COS of A - COS

AUTHOR: W. S. LaSor, SDS

ACCEPTED: 19 February 1963

COMPUTER CONFIGURATION: Any 920/930 Computer

PURPOSE: To compute the cosine of an argument in the A register.

PROGRAMMED OPERATORS: SIN (COS and SIN are contained in the same subroutine)

STORAGE: Instructions and constants: 40 oct, 32 dec

Uses temporary storage locations 10 thru 12.

TIMING: 464 - 504 μsec

SOURCE LANGUAGE: SYMBOL

USE:

1. FUNCTION

The cosine of the contents of A replaces the contents of A. The address field is not used.

2. ARGUMENT

See program description for SIN of A.

METHOD: Use is made of the identity:

$$\cos X \equiv \sin (X + 90)$$

by adding 90 degrees to the argument. The subroutine then exits through SIN.

IDENTIFICATION: SIN of A - SIN

AUTHOR: W. S. LaSor, SDS

ACCEPTED: 19 February 1963

COMPUTER CONFIGURATION: Any 920/930 Computer

PURPOSE: To compute the sine of an argument in the A register.

PROGRAMMED OPERATORS: None

STORAGE: Instructions and constants: 40 oct, 32 dec

Uses temporary storage locations 10 thru 12.

TIMING: 448 - 488 μsec

SOURCE LANGUAGE: SYMBOL

USE:

1. FUNCTION

The sine of the contents of A replaces the contents of A. The address field is not used.

2. ARGUMENT

The argument is in fractions of a circle scaled at Q = -1. Thus:

40000000 = -180°

60000000 = -90°

00000000 = 0°

20000000 = 90°

37777777 = 179.99998°, etc.

USE: (Cont.) On exit, the sine is in A scaled at Q = 1. The B and X registers are unchanged.

METHOD: The argument is first reduced to either the first of fourth quadrants and the result used to evaluate:

$$\sum_{1}^{5} C_{2k-1} X^{2k-1}$$

where:

$C_1 = 1.5707963$

$C_3 = -.64596371$

$C_5 = .07968968$

$C_7 = -.00467377$

$C_9 = .00015148$

Maximum absolute error is less than 10^{-6}.

```
                                    1  $COS    POPD    012700000
00000   4 55 00030                  2  COS     ADD     P90,4           ADD 90 DEGREES
                                    3  $SIN    POPD    012600000
00001   0 36 00013                  4  SIN     STB     TB              SAVE B REGISTER
00002   4 73 00030                  5          SKG     P90,4           LOCATE QUADRANT
00003   4 73 00031                  6          SKG     M90,4
00004   4 01 00024                  7          BRU     REDUCE,4        2ND OR 3RD
00005   0 35 00012                  8  EVAL    STA     TA              1ST OR 4TH
00006   0 64 00012                  9          MUL     TA
00007   0 35 00014                 10          STA     ARGSQ
00010   4 64 00036                 11          MUL     C9,4            EVAL POLYNOMIAL
00011   4 55 00035                 12          ADD     C7,4
00012   0 64 00014                 13          MUL     ARGSQ
00013   4 55 00034                 14          ADD     C5,4
00014   0 64 00014                 15          MUL     ARGSQ
00015   4 55 00033                 16          ADD     C3,4
00016   0 64 00014                 17          MUL     ARGSQ
00017   4 55 00032                 18          ADD     C1,4
00020   0 64 00012                 19          MUL     TA
00021   0 67 00002                 20          LSH     2               SCALE AT 1
00022   0 75 00013                 21          LDB     TB              RESTORE B
00023   0 51 00000                 22          BRR     0
00024   0 02 20001                 23  REDUCE  EOM     020001          RESET OVFLO
00025   0 46 01000                 24          CNA                     REDUCE QUDRANT
00026   0 17 00037                 25          FOR     SIGN
00027   4 01 00005                 26          BRU     EVAL,4
00030   20000000                   27  P90     DATA    020000000
00031   57777777                   28  M90     DATA    057777777
00032   14441767                   29  C1      DATA    014441767
00033   53250420                   30  C3      DATA    053250420
00034   12146426                   31  C5      DATA    012146426
00035   75466632                   32  C7      DATA    075466632
00036   00236660                   33  C9      DATA    0236660
        00000012                   34  TA      EQU     10
        00000013                   35  TB      EQU     11
        00000014                   36  ARGSQ   EQU     12
00037   40000000                   37  SIGN    DATA    040000000
                                   38          END
```

IDENTIFICATION: Square Root, Fixed, Double - DSQ

AUTHOR: Richard S. Resnick, SDS

ACCEPTED: 22 March 1963

COMPUTER CONFIGURATION: Any SDS 920/930 Computer

PURPOSE: To extract the square root of the double precision argument in the A, B registers.

PROGRAMMED OPERATORS: None

STORAGE: Instructions and constants: 122 oct, 82 dec
Uses temporary storage locations 12 thru 15.

TIMING: 1.112 to 1.320 milliseconds

SOURCE LANGUAGE: SYMBOL

USE: The root of the argument replaces the argument in A, B. The argument is a double precision fixed point number in A, B. The sign and 23 most significant bits are in A (A_0 thru A_{23}) and the 24 least significant bits are in (B_0 thru B_{23}). The X register is unchanged.

METHOD: The argument is inspected for the interval in which it lies. The intervals are designated by subscipt i and are:

i = 0	$5/16 > A \geq 1/4$
i = 1	$3/8 > A \geq 5/16$
i = 2	$7/16 > A \geq 3/8$
i = 3	$1/2 > A \geq 7/16$
i = 4	$5/8 > A \geq 1/2$
i = 5	$3/4 > A \geq 5/8$
i = 6	$7/8 > A \geq 3/4$
i = 7	$1 > A \geq 7/8$

METHOD: (cont.) The root is then approximated on the i^{th} interval by the polynomial:

$$r_o = \sum_{J=0}^{3} CjiA^j \quad (i = 0 \text{ thru } 7)$$

where:

i = 0	C00 =	.1651632200
	C10 =	.8864482594
	C20 =	-.5275337877
	C30 =	.1879749603
	C01 =	.1828018288
	C11 =	.8010519649
	C21 =	-.3894905093
	C31 =	.1134708047
	C02 =	.1988250306
	C12 =	.7365302516
	C22 =	-.3027946859
	C32 =	.0745999492
	C03 =	.2136922437
	C13 =	.6853650700
	C23 =	-.2440563169
	C33 =	.0521059196
	C04 =	.23357606
	C14 =	1.25362715
	C24 =	-.74604543
	C34 =	.26583673
	C05 =	.25852082
	C15 =	1.13285855
	C25 =	-.55082276
	C35 =	.16047195
	C06 =	.28118105
	C16 =	1.04061107
	C26 =	-.42821635
	C36 =	.10550026
	C07 =	.30220646
	C17 =	.96925257
	C27 =	-.34514775
	C37 =	.07368889

METHOD: (cont.) r_o is accurate to at least 20 bits and represents a very good first approximation of the root. One Newton-Raphson interation is performed:

$$r_1 = 1/2 \left[\frac{A}{r_o} + r_o \right]$$

where:

r_o is the first approximation accurate to at least 20 bits.

A is the argument in A, B.

r_1 is the final root, accurate to at least 40 bits.

The polynomial renders the first approximation scaled at Q = 0. The argument is shifted right 1 bit for scaling purposes before the iteration is performed. A double precision divide is performed to form the quotient of:

$$\frac{A}{r_o}$$

If the argument is negative, overflow is set and the program exits. If the argument is zero the program exits immediately. The original argument can be found in locations 12 and 13.

```
                                1  $DSQ    P0PD    013500000
00000  0 37 00012               2  DSQ     STX     TX
00001  4 71 00056               3          LDX     RSH,4           SET RESCALE SHIFT
00002  0 67 10056               4          N0D     46              NORMALIZE N PLACES
00003  0 73 00117               5          SKG     =0
00004  4 01 00045               6          BRU     DSQ1,4          ARG LEQ 0
00005  0 35 00014               7          STA     AH
00006  0 36 00013               8          STB     AL
00007  0 46 00604               9          RCH     0604            XXA,CAB
00010  0 46 01000              10          CNA                     NEGATE A
00011  0 46 00104              11          RCH     0104            CABE
00012  0 66 20001              12          RCY     1               FORM N/2
00013  4 35 00042              13          STA     SCALE,4         STORE RSH N/2
00014  0 46 00001              14          CLA
00015  0 67 00005              15          LSH     5
00016  0 46 00600              16          XXA
00017  6 64 00067              17          MUL     C3-4,6          FIND RT
00020  6 55 00063              18          ADD     C2-4,6
00021  0 64 00014              19          MUL     AH
00022  6 55 00057              20          ADD     C1-4,6
00023  0 64 00014              21          MUL     AH
00024  6 55 00053              22          ADD     C0-4,6
00025  0 62 00014              23          XMA     AH              EXCHANGE AH AND RT
00026  0 75 00013              24          LDB     AL
00027  0 66 00001              25          RSH     1
00030  0 65 00014              26          DIV     AH              AH,AL/RT
00031  0 35 00015              27          STA     Q               Q EQU AH,AL/RT
00032  0 46 10012              28          BAC
00033  0 65 00014              29          DIV     AH              REMAINDER/RT
00034  0 64 00120              30          MUL     =1
00035  0 55 00015              31          ADD     Q               COMPLETE DP DIVIDE
00036  4 53 00042              32          SKN     SCALE,4         TEST N EVEN OR ODD
00037  0 67 00001              33          LSH     1               N IS EVEN
00040  0 55 00014              34          ADD     AH              COMPLETE N-R ITER
00041  0 02 20001              35          E0M     020001          RESET OVFL0
00042  0 66 00000              36  SCALE   RSH                     RESCALE RT
00043  0 71 00012              37          LDX     TX
00044  0 51 00000              38          BRR     0               EXIT
```

```
00045  0 35 00014      39  DSQ1   STA     AH
00046  0 46 00200      40         CXA
00047  0 46 01000      41         CNA
00050  4 55 00056      42         ADD     RSH,4
00051  0 62 00014      43         XMA     AH
00052  0 66 40014      44         RSH     *AH
00053  0 72 00121      45         SKA     =040000000
00054  4 51 00054      46         BRR     $,4             SET OVFLO
00055  4 01 00043      47         BRU     SCALE+1,4       LOAD X AND EXIT
00056  62327776        48  RSH    DATA    062327776
00057  07362716        49  CC     DATA    07362716
00060  10213464        50         DATA    010213464
00061  10776675        51         DATA    010776675
00062  11527263        52         DATA    011527263
00063  50073333        53  C1     DATA    050073333
00064  44200602        54         DATA    044200602
00065  41251603        55         DATA    041251603
00066  37010170        56         DATA    037010170
00067  50100625        57  C2     DATA    050100625
00070  56277244        58         DATA    056277244
00071  62230065        59         DATA    062230065
00072  64751063        60         DATA    064751063
00073  10403360        61  C3     DATA    010403360
00074  05105130        62         DATA    05105130
00075  03300410        63         DATA    03300410
00076  02267243        64         DATA    02267243
00077  05222017        65         DATA    05222017
00100  05663014        66         DATA    05663014
00101  06271431        67         DATA    06271431
00102  06655106        68         DATA    06655106
00103  34273443        69         DATA    034273443
00104  31504337        70         DATA    031504337
00105  27443240        71         DATA    027443240
00106  25735013        72         DATA    025735013
00107  57074706        73         DATA    057074706
00110  63422455        74         DATA    063422455
00111  66237006        75         DATA    066237006
00112  70141303        76         DATA    070141303
00113  06007620        77         DATA    06007620
```

```
00114  03503066          78            DATA    03503066
00115  02306176          79            DATA    02306176
00116  01525550          80            DATA    01525550
       00000012          81   TX       EQU     012
       00000013          82   AL       EQU     013
       00000014          83   AH       EQU     014
       00000015          84   Q        EQU     015
                         85            END
00117  00000000
00120  00000001
00121  40000000
```

IDENTIFICATION: Square Root, Floating, Double - FSQ

AUTHOR: Richard S. Resnick, SDS

ACCEPTED: 22 March 1963

COMPUTER CONFIGURATION: Any SDS 920/930 Computer

PURPOSE: To extract the square root of the double precision argument in the A, B registers.

PROGRAMMED OPERATORS: None

STORAGE: Instructions and constants: 107 oct, 71 dec
Uses temporary storage locations 12 thru 15.

TIMING: 1.056 to 1.080 milliseconds

SOURCE LANGUAGE: SYMBOL

USE: The root of the argument replaces the argument in A, B. The argument is a double precision floating point number in A, B with a 39 bit mantissa and 9 bit exponent field. The sign and 23 most significant bits of the mantissa are in A (A_0 thru A_{23}) and the 15 least significant bits of the mantissa are in B (B_0 thru B_{14}). The exponent field is also in B (B_{15} thru B_{23}). The X register is unchanged.

METHOD: The argument is inspected for the interval in which it lies. The intervals are designated by subscript i and are:

i = 0	$5/16 > A \geq 1/4$	i = 4	$5/8 > A \geq 1/2$
i = 1	$3/8 > A \geq 5/16$	i = 5	$3/4 > A \geq 5/8$
i = 2	$7/16 > A \geq 3/8$	i = 6	$7/8 > A \geq 3/4$
i = 3	$1/2 > A \geq 7/16$	i = 7	$1 > A \geq 7/8$

METHOD: (cont.) The root is then approximated on the i^{th} interval by the polynomial:

$$r_o = \sum_{J=0}^{3} CjiA^j \quad (i = 0 \text{ thru } 7)$$

where:

i = 0

C00 = .1651632200
C10 = .8864482594
C20 = -.5275337877
C30 = .1879749603

C01 = .1828018288
C11 = .8010519649
C21 = -.3894905093
C31 = .1134708047

C02 = .1988250306
C12 = .7365302516
C22 = -.3027946859
C32 = .0745999492

C03 = .2136922437
C13 = .6853650700
C23 = -.2440563169
C33 = .0521059196

C04 = .23357606
C14 = 1.25362715
C24 = -.74604543
C34 = .26583673

C05 = .25852082
C15 = 1.13285855
C25 = -.55082276
C35 = .16047195

C06 = .28118105
C16 = 1.04061107
C26 = -.42821635
C36 = .10550026

C07 = .30220646
C17 = .96925257
C27 = -.34514775
C37 = .07368889

METHOD: (Cont.) r_o is accurate to at least 20 bits and represents a very good first approximation of the root. One Newton-Raphson iteration is performed:

$$r_1 = 1/2 \left[\frac{A}{r_o} + r_o \right]$$

where:

r_o is the first approximation accurate to at least 20 bits.

A is the argument (mantissa) in A, B.

r_1 is the final root, accurate to at least 40 bits.

The exponent of the root is formed by the algorithm:

$$RE = 1/2 (AE + 1)$$

The polynomial renders the first approximation scaled at Q = 1. The argument is shifted right 2 bits for scaling purposes before the iteration is performed. A double precision divide is performed to form the quotient of:

$$\frac{A}{r_o}$$

If the argument is negative, overflow is set and the program exits. If the argument is zero the program exits immediately. The original argument can be found in locations 12 and 13.

```
                                1  $FSQ    POPD    014500000
00000   0 73 00107              2  FSQ     SKG     =0
00001   4 01 00043              3          BRU     FSQ1,4              SPECIAL CASE
00002   0 37 00014              4          STX     TX
00003   0 35 00013              5          STA     AH
00004   0 36 00012              6          STB     AL
00005   0 46 00414              7          RCH     0414                CAX,XAB
00006   0 67 20004              8          LCY     4
00007   4 14 00046              9          ETR     MC,4
00010   0 46 00600             10          XXA
00011   6 64 00063             11          MUL     C3,6                PERFORM CUBIC FIT
00012   6 55 00057             12          ADD     C2,6
00013   0 64 00013             13          MUL     AH
00014   6 55 00053             14          ADD     C1,6
00015   0 64 00013             15          MUL     AH
00016   6 55 00047             16          ADD     C0,6
00017   0 62 00012             17          XMA     AL
00020   0 46 00500             18          RCH     0500
00021   4 41 00022             19          BRX     $+1,4               AE + 1
00022   0 46 00600             20          XXA
00023   0 66 00001             21          RSH     1                   RE EQU (AE + 1)/2
00024   0 75 00013             22          LDB     AH
00025   0 46 00450             23          RCH     0450                CAX,CXB,CBA
00026   0 52 00110             24          SKB     =1
00027   0 66 00001             25          RSH     1                   AE ODD
00030   0 66 00002             26          RSH     2
00031   0 65 00012             27          DIV     AL
00032   0 35 00015             28          STA     Q
00033   0 46 10012             29          BAC
00034   0 65 00012             30          DIV     AL
00035   0 64 00110             31          MUL     =1
00036   0 55 00015             32          ADD     Q
00037   0 55 00012             33          ADD     AL
00040   0 46 00140             34          LDE
00041   0 71 00014             35          LDX     TX
00042   0 51 00000             36          BRR     0
00043   0 50 00107             37  FSQ1    SKE     =0
00044   4 51 00044             38          BRR     $,4                 NEG ARG - SET OVFLO
```

```
00045  0 51 00000      39          BRR     0               EXIT
00046  00000023        40    MC    DATA    023
00047  03571347        41    CC    DATA    03571347
00050  04105632        42          DATA    04105632
00051  04377337        43          DATA    04377337
00052  04653532        44          DATA    04653532
00053  24035556        45    C1    DATA    024035556
00054  22100301        46          DATA    022100301
00055  20524702        47          DATA    020524702
00056  17404074        48          DATA    017404074
00057  64040313        49    C2    DATA    064040313
00060  67137522        50          DATA    067137522
00061  71114033        51          DATA    071114033
00062  72364432        52          DATA    072364432
00063  04201570        53    C3    DATA    04201570
00064  02442454        54          DATA    02442454
00065  01540204        55          DATA    01540204
00066  01133522        56          DATA    01133522
00067  02511007        57          DATA    02511007
00070  02731406        58          DATA    02731406
00071  03134614        59          DATA    03134614
00072  03326443        60          DATA    03326443
00073  16135622        61          DATA    016135622
00074  14642160        62          DATA    014642160
00075  13621520        63          DATA    013621520
00076  12756406        64          DATA    012756406
00077  67436343        65          DATA    067436343
00100  71611227        66          DATA    071611227
00101  73117403        67          DATA    073117403
00102  74060542        68          DATA    074060542
00103  03003710        69          DATA    03003710
00104  01641433        70          DATA    01641433
00105  01143077        71          DATA    01143077
00106  00652664        72          DATA    0652664
       00000012        73    AL    EQU     012
       00000013        74    AH    EQU     013
       00000014        75    TX    EQU     014
       00000015        76    Q     EQU     015
00107  00000000        77          END
00110  00000001
```

IDENTIFICATION: Square Root of A - SQR

AUTHOR: Richard S. Resnick, SDS

ACCEPTED: 26 December 1962

COMPUTER CONFIGURATION: Any SDS 920/930 Computer

PURPOSE: To extract the square root from the 24 most significant bits of an argument in the A, B registers.

PROGRAMMED OPERATORS: None

STORAGE: Instructions plus constants: 124 oct, 84 dec
Uses temporary storage locations 12 and 13.

TIMING: 384 to 576 μsec

SOURCE LANGUAGE: SYMBOL

USE: The root of the argument replaces the first 24 significant bits of the argument in A, B. The argument is fixed point number of even Q in A, B. On exit the root is in A, B at Q/2. The X register is unchanged.

METHOD: The argument is normalized and inspected for the interval in which it lies. The intervals are designated by subscript i and are:

$i = 0 \quad 5/8 > A \geq 1/2$

$i = 1 \quad 3/4 > A \geq 5/8$

$i = 2 \quad 7/8 > A \geq 3/4$

$i = 3 \quad 1 > A \geq 7/8$

The root is then approximated on the i^{th} interval by the polynomial:

$$r = \sum_{j=0}^{3} CjiA^{j} \quad (i = 0, 1, 2, 3)$$

METHOD: (Cont.)

where:

i = 0	C00 =	.23357606
	C10 =	1.25362715
	C20 =	.74604543
	C30 =	.26583673
i = 1	C01 =	.25852082
	C11 =	1.13285855
	C21 =	.55082276
	C31 =	.16047195
i = 2	C02 =	.28118105
	C12 =	1.04161107
	C22 =	.42821635
	C33 =	.10550026
i = 3	C03 =	.30220646
	C13 =	.96925257
	C23 =	.34514775
	C33 =	.07368889

The root is then scaled at Q/2. Maximum absolute error for:

$i = 0$ is $1.09 \cdot 10^{-6}$

$i = 1$ is $6.38 \cdot 10^{-7}$

$i = 2$ is $3.58 \cdot 10^{-7}$

$i = 3$ is $5.73 \cdot 10^{-7}$

If the argument is negative, the results will be negative. The original argument can be found in location 12.

```
                                1  $SQR    POPD    014700000
00000   0 37 00013              2  SQR     STX     TX                  SAVE X
00001   4 71 00021              3          LDX     MLT,4               PRESET SCALE ADDRESS
00002   0 67 10060              4          NOD     48
00003   0 35 00012              5          STA     TA
00004   4 37 00016              6          STX     XEC,4               TO EXECUTE
00005   0 46 00402              7          RCH     0402                CAX + CLB
00006   0 67 20004              8          LCY     4
00007   0 46 00220              9          RCH     0220                CXA + CBX
00010   6 64 00032             10          MUL     C3-4,6              EVAL POLYNOMIAL
00011   6 55 00026             11          ADD     C2-4,6
00012   0 64 00012             12          MUL     TA
00013   6 55 00022             13          ADD     C1-4,6
00014   0 64 00012             14          MUL     TA
00015   6 57 00016             15          ADC     C0-4,6
00016   0 64 00000             16  XEC     MUL     0                   SCALE RESULT
00017   0 71 00013             17          LDX     TX                  RESTORE X
00020   0 51 00000             18          BRR     0
00021   4 64 00123             19  MLT     MUL     SC+48,4             RESCALE ARG
00022   07362716               20  C0      DATA    07362716
00023   10213464               21          DATA    010213464
00024   10776675               22          DATA    010776675
00025   11527262               23          DATA    011527262
00026   50073333               24  C1      DATA    050073333
00027   44200602               25          DATA    044200602
00030   41251603               26          DATA    041251603
00031   37010170               27          DATA    037010170
00032   50100625               28  C2      DATA    050100625
00033   56277244               29          DATA    056277244
00034   62230065               30          DATA    062230065
00035   64751063               31          DATA    064751063
00036   10403360               32  C3      DATA    010403360
00037   05105130               33          DATA    05105130
00040   03300410               34          DATA    03300410
00041   02267243               35          DATA    02267243
00042   40000000               36          DATA    040000000
00043   00000000               37  SC      DATA    0
00044   00000001               38          DATA    01
```

```
00045  00000001        39          DATA    01
00046  00000002        40          DATA    02
00047  00000002        41          DATA    02
00050  00000003        42          DATA    03
00051  00000004        43          DATA    04
00052  00000006        44          DATA    06
00053  00000010        45          DATA    010
00054  00000014        46          DATA    014
00055  00000020        47          DATA    020
00056  00000027        48          DATA    027
00057  00000040        49          DATA    040
00060  00000056        50          DATA    056
00061  00000100        51          DATA    0100
00062  00000133        52          DATA    0133
00063  00000200        53          DATA    0200
00064  00000266        54          DATA    0266
00065  00000400        55          DATA    0400
00066  00000553        56          DATA    0553
00067  00001000        57          DATA    01000
00070  00001325        58          DATA    01325
00071  00002000        59          DATA    02000
00072  00002651        60          DATA    02651
00073  00004000        61          DATA    04000
00074  00005521        62          DATA    05521
00075  00010000        63          DATA    010000
00076  00013241        64          DATA    013241
00077  00020000        65          DATA    020000
00100  00026502        66          DATA    026502
00101  00040000        67          DATA    040000
00102  00055203        68          DATA    055203
00103  00100000        69          DATA    0100000
00104  00132405        70          DATA    0132405
00105  00200000        71          DATA    0200000
00106  00265012        72          DATA    0265012
00107  00400000        73          DATA    0400000
00110  00552024        74          DATA    0552024
00111  01000000        75          DATA    01000000
00112  01324050        76          DATA    01324050
00113  02000000        77          DATA    02000000
```

```
00114  02650117          78          DATA    02650117
00115  04000000          79          DATA    04000000
00116  05520236          80          DATA    05520236
00117  10000000          81          DATA    010000000
00120  13240474          82          DATA    013240474
00121  17777777          83          DATA    017777777
00122  26501171          84          DATA    026501171
00123  37777777          85          DATA    037777777
       00000012          86  TA      EQU     012
       00000013          87  TX      EQU     013
                         88          END
```

IDENTIFICATION: Fixed-Floating Format Conversion Programmed Operator - FFF

AUTHOR: T. Marshall, SDS

ACCEPTED: February 11, 1964

COMPUTER CONFIGURATION: Any SDS 920/930 Computer

PURPOSE: Performs conversion between fixed point single precision, fixed point double precision, floating point single precision, and floating point double precision formats.

PROGRAMMED OPERATORS: None used.

STORAGE: Instructions and constants: 122 oct, 82 dec

TIMING: 280 to 632 microseconds, including programmed operator entry and exit.

SOURCE LANGUAGE: SYMBOL

USE: 1. INSTRUCTION CODING

The effective address of an FFF instruction specifies format conversion performed and scaling of fixed number (if any) as follows:

Octal Digit	Binary Bits	Significance
000X0000	10 - 11	Format converted from
0000X000	13 - 14	Format converted to
00000XXX	15 - 23	Scaling of fixed point number in two's compleme

USE: (Cont.) **Formats are numbered as follows:**

0 - Fixed point, single precision
1 - Floating point, single precision
2 - Fixed point, double precision
3 - Floating point, double precision

A scaling of 000 indicates that the binary point of the fixed point number is immediately right of the sign bit - i.e., that the number is a fraction less than or equal to one. A positive scaling other than zero specifies the number of binary places between the sign bit and the binary point to the right. A negative scaling specifies that the binary point is to the left of the sign. (However, the most significant bit of the number will still contain the sign.) Positive scaling may range from 000 through 377 octal, negative scaling from 777 through 400.

Scaling can alternately be conceived of as follows: The intended value of a fixed point number equals the value of the number as a fraction, times two raised to the S power, S being specified by scaling.

Example: A single register integral number would have scaling of 027 octal (23 decimal).

If both formats (conversion from and conversion to) are floating point, scaling has no significance.

2. FORMATS

1 - Fixed point single precision

A 24-bit number is contained in the A register in two's complement form.

2 - Floating point single precision

A 24-bit fraction is contained in the A register as a normalized, fractional two's complement number.

A 24-bit exponent is contained in the B register as an integral two's complement number.

3 - Fixed point double precision

The most significant 24 bits of the number is contained in the A register, the least significant 24 bits in the B register. The number is in two's complement form. Bit zero position of the B register contains a magnitude bit (not sign).

USE: (Cont.)

4 - Floating point double precision

A 39-bit **fraction** is contained in the A register and bit positions 0 through 14 of the B register, as a normalized, fractional two's complement number.

A 9 bit exponent is contained in bit positions 15 through 23 of the B register as an integral two's complement number.

3. FUNCTIONS

FFF 00 - No Operation

FFF 01 - Fixed point single precision to Floating point single precision

Zero becomes zero **fraction** and -2^8 exponent.

FFF 02 - Fixed point single precision to Fixed point double precision

B register is cleared. The contents of A register are not changed.

FFF 03 - Fixed point single precision to Floating point double precision.

Zero becomes zero fraction and -2^8 exponent.

FFF 10 - Floating point single precision to Fixed point single precision

Performs same operation as FFF 12. The result is double register.

FFF 11 - No Operation

FFF 12 - Floating point single precision to Fixed point double precision

If scaling specified is less than exponent overflow is set.

If difference between exponent and scaling exceeds 47, the result is set to zero.

FFF 13 - Floating point single precision to Floating point double precision

If exponent exceeds (2^8-1), overflow is set.

If exponent is less than -2^8, fraction is set to zero and exponent to -2^8.

Bit positions 0 through 14 of B register are set to zero.

USE: (Cont.)

FFF 20 - No Operation

FFF 21 - Fixed point double precision to
Floating point single precision

Zero (double register) becomes zero fraction and -2^8 exponent.

FFF 22 - No Operation

FFF 23 - Fixed point double precision to
Floating point double precision

Zero (double register) becomes zero fraction and -2^8 exponent.

FFF 30 - Floating point double precision to
Fixed point single precision

Performs same operation as FFF 32. The result is double register.

FFF 31 - Floating point double precision to
Floating point single precision

The sign of the exponent is extended left to bit zero of the B register.

FFF 32 - Floating point double precision to
Fixed point double precision

If scaling specified is less than exponent, overflow is set.

If difference between exponent and scaling exceeds 47, the result is set to zero.

FFF 33 - No Operation

METHOD:

Upon entry the effective address is obtained. The effective address is right shifted 9 bits and extracted with 00000033 to obtain format conversion specification. The format conversion specification becomes the table address of an indexed execute. The executed table operation performs a branch to subroutine, an operation, or a NOP, as a function of the format conversion specification.

METHOD: (Cont.) If a branch is not executed, the routine reloads original contents of registers and exits.

If fixed to floating conversion is specified, overflow is set as a flag if conversion is to floating single precision. If the fixed number is single precision, the B register is cleared. The number is always converted to floating double precision, then this in turn is changed to floating single precision if required.

A common subroutine performs the actual fixed to floating conversion. The scaling is extracted from the effective address. If the number is zero, the exponent is set to 400, otherwise the number is normalized and scaling is correspondingly decremented. The normalized number is truncated to 39 bits and becomes the fraction. The decremented scaling becomes the exponent. The subroutine then branches as a function of the flag; exiting if floating double precision is to be the final format.

If conversion to floating single precision is specified, the sign of the exponent (bit 15) is extended left throughout the B register.

If floating to fixed conversion is specified, the fraction is subtracted from the specified scaling. If this difference is negative, overflow is set. If the difference is greater than 57 (octal), the result is set to zero. Otherwise, the difference is placed in the X register and conditions an indexed right shift of the fraction. The result is the fixed point number.

If floating single to floating double conversion is specified, the exponent is tested. If the exponent is greater than 377 (octal) the overflow is turned on. If the exponent is less than 77777400 (octal), the fraction is set to zero and the exponent is set to 400. Otherwise the exponent is truncated to 9 bits and the low 15 bits of the fraction are set to zero.

```
                00000012          1  AH      EQU     012
                00000013          2  TEMP    EQU     013
                00000014          3  XREG    EQU     014
                00000024          4  ONE     EQU     024
                00000025          5  SIGN    EQU     025
                00000026          6  ONES    EQU     026
                                  7  $FFF    POPJ    015100000
00000   0 35 00012                8  FFF     STA     AH
00001   0 37 00014                9          STX     XREG
00002   0 77 40000               10          EAX     *0
00003   0 37 00013               11          STX     TEMP
00004   0 46 00222               12          RCH     0222
00005   0 66 20011               13          RCY     9
00006   0 14 00033               14          ETR     F33             020000033
00007   0 46 00440               15          RCH     0440
00010   2 23 00011               16          EXU     F11,2
00011   0 01 00044               17  F11     BRU     F44             END
00012   0 01 00047               18  F12     BRU     F47             FIX TO FLOAT
00013   0 46 00002               19          CLB
00014   0 51 00022               20          BRR     F22             SET OVERFLOW. FIX TO FLOAT
00015   0 76 00012               21  F15     LDA     AH              FLOATING
00016   0 52 00025               22          SKB     SIGN            DOUBLE
00017   0 55 00024               23          ADD     ONE             TO
00020   0 01 00064               24          BRU     F64             SINGLE
00021   0 01 00025               25          BRU     F25             LINKAGE TABLE
00022   0 20 00046               26  F22     NOP     F46
00023   0 01 00025               27          BRU     F25
00024   0 01 00103               28          BRU     F103
00025   0 46 00120               29  F25     RCH     0120
00026   0 46 00002               30          CLB
00027   0 01 00070               31          BRU     F70             FLOAT TO FIX+1
00030   00000400                 32  F30     DATA    0400
00031   0 20 00000               33          NOP                     LINKAGE TABLE
00032   0 01 00050               34          BRU     F50             FIX TO FLOAT+1
00033   0 20 00033               35  F33     NOP     27
00034   0 51 00012               36          BRR     F12             FIX TO FLOAT+1. SET OVERFLOW
00035   0 46 00502               37  F35     RCH     0502            FLOATING TO FIXED
00036   0 76 00012               38          LDA     AH
```

```
00037  2 66 00000      39          RSH     0,2         SCALE
00040  0 01 00045      40          BRU     F45         END+1
00041  0 01 00067      41          BRU     F67
00042  0 01 00015      42          BRU     F15
00043  0 01 00067      43          BRU     F67
00044  0 76 00012      44  F44     LDA     AH
00045  0 71 00014      45  F45     LDX     XREG
00046  0 51 00000      46  F46     BRR     0           RETURN
00047  0 46 00002      47  F47     CLB
00050  0 71 00013      48  F50     LDX     TEMP        SCALING
00051  0 76 00012      49          LDA     AH
00052  0 72 00026      50          SKA     0NES
00053  0 01 00060      51          BRU     F60         AH NOT = 0
00054  0 70 00026      52          SKM     0NES
00055  0 01 00060      53          BRU     F60         AL NOT = 0
00056  0 75 00030      54          LDB     F30
00057  0 01 00062      55          BRU     F62         ZERO EXIT
00060  0 67 10060      56  F60     NOD     48
00061  0 46 00140      57          RCH     0140
00062  0 40 20001      58  F62     SKS     020001      OVERFLOW
00063  0 01 00045      59          BRU     F45         END+1. FLOATING DOUBLE
00064  0 46 00120      60  F64     RCH     0120
00065  0 46 00040      61          CXB
00066  0 01 00045      62          BRU     F45
00067  0 46 00122      63  F67     RCH     0122
00070  0 46 00200      64  F70     CXA
00071  0 62 00013      65          XMA     TEMP        EXCHANGE SCALING AND AE
00072  0 46 00500      66          RCH     0500
00073  0 46 00200      67          CXA
00074  0 54 00013      68          SUB     TEMP
00075  0 72 00025      69          SKA     SIGN
00076  0 01 00110      70          BRU     F110        OVERFLOW
00077  0 73 00121      71          SKG     F121        00000057
00100  0 01 00035      72          BRU     F35         R = 0
00101  0 46 30003      73          CLR                 R NOT = 0
00102  0 01 00045      74          BRU     F45         END+1
00103  0 46 10012      75  F103    BAC
00104  0 73 00117      76          SKG     F117
00105  0 01 00114      77          BRU     F114        UNDERFLOW
```

```
00106   0 73 00120      78          SKG     F120
00107   0 01 00112      79          BRU     F112        NO OVERFLOW
00110   0 75 00120      80  F110    LDB     F120
00111   0 51 00113      81          BRR     F113        SET OVERFLOW TO END+1
00112   0 46 00104      82  F112    RCH     0104
00113   0 01 00044      83  F113    BRU     F44         TO END
00114   0 46 30003      84  F114    CLR
00115   0 75 00030      85          LDB     F30
00116   0 01 00045      86          BRU     F45         TO END+1
00117   77777400        87  F117    DATA    077777400
00120   00000377        88  F120    DATA    00000377
00121   00000057        89  F121    DATA    00000057
                        90          END
```

IDENTIFICATION: Floating Point, Single Precision Arithmetic Programmed Operator Package

AUTHOR: Richard S. Resnick, SDS

ACCEPTED: 5 February 1963

COMPUTER CONFIGURATION: Any SDS 920/930 Computer

PURPOSE: To provide the following single precision floating point arithmetic functions:

Mnemonic	Programmed Operator
FSA	Floating Add, Single Precision
FSS	Floating Subtract, Single Precision
FSM	Floating Multiply, Single Precision
FSD	Floating Divide, Single Precision
FSN	Floating Negate, Single Precision

PROGRAMMED OPERATORS: None

STORAGE: Instructions and constants: 174 oct, 124 dec

Uses standard subroutine temporary storage locations 12 through 14, and location 1.

TIMING: (microseconds)

FSA	352-480	FSD	464-480
FSS	368-480	FSN	64- 72
FSM	248-264		

SOURCE LANGUAGE: SYMBOL

USE:

1. FORMAT

Each floating, single precision number occupies two words. The fraction is contained in one word in two's complement form. Its value may be:

$$1/2 \leq \text{fraction} < 1$$

$$-1 \leq \text{fraction} < -1/2$$

$$\text{fraction} = 0$$

The exponent is contained in the other word as an integer in two's complement form scaled at 23. Its value may lie in the range:

$$2^{-8} \leq \text{exponent} < 2^{8}$$

Both A and B registers together constitute a "double accumulator" for floating single precision programmed operators. The fraction is contained in A register and the exponent in B register.

The contents of the two memory locations designated by the effective address of a programmed operator, and by the effective address plus one, constitute a "double operand". The exponent is contained in the location designated by the effective address; the fraction is contained in the following location.

2. FUNCTIONS

Except as noted, indexing and indirect addressing are possible on all instructions.

FLOATING ADD, SINGLE PRECISION FSA

The "floating operand" is added to the "floating accumulator", and the normalized sum appears in the "floating accumulator."

Registers Affected: A, B Timing: 44 - 60 cycles

FLOATING SUBTRACT, SINGLE PRECISION FSS

The "floating operand" is subtracted from the "floating accumulator" and the normalized difference appears in the "floating accumulator."

Registers Affected: A, B Timing: 46 - 60 cycles

USE (Cont.)

FLOATING MULTIPLY, SINGLE PRECISION FSM

The "floating operand" is multiplied by the contents of the "floating accumulator". The normalized product appears in the "floating accumulator."

Registers Affected: A, B Timing: 31 - 33 cycles

FLOATING DIVIDE, SINGLE PRECISION FSD

The contents of the "floating accumulator" are divided by the "floating operand". The normalized quotient appears in the "floating accumulator."

If the divisor is zero, overflow will be turned on.

Registers Affected: A, B Timing: 58 - 60 cycles

FLOATING NEGATE, SINGLE PRECISION FSN

The contents of the "floating accumulator" are negated. The address portion of the instruction is not used.

Indirect addressing and indexing are not used.

Registers Affected: A, B Timing: 8 - 9 cycles

3. ACCURACY

Precision of results is 23 bits plus sign. Results are not rounded.

If an exponent becomes larger than 255 or smaller than 256, the result will remain arithmetically correct. However, if such a result is converted to floating double precision, it will cause overflow or underflow. Probability of an exponent exceeding 2^{23} is negligibly small, therefore this condition is not detected.

METHOD: 1. NOMENCLATURE

Ah = Fractional portion of operand initially in accumulator (A register)

Ae = Exponent of operand initially in accumulator (B register)

Mh = Fractional portion of operand in memory (location M+1)

Me = Exponent of operand in memory (location M)

METHOD: (Cont.)Rh = Fractional portion of result (A register)

Re = Exponent of result (B register)

All fractions are normalized. All exponents are integers scaled at 23.

2. FLOATING ADD, SINGLE PRECISION

The exponents are compared. If their difference is greater than 32, the operand having the larger exponent is loaded into the accumulator and the subroutine exits. Otherwise, the fraction having the smaller exponent is shifted right a number of places equal to the difference, and added to the other fraction. The larger exponent becomes Re.

If the addition overflows, Rh is shifted right one position, the sign changed, and Re is incremented. Otherwise, the result is normalized.

3. FLOATING SUBTRACT, SINGLE PRECISION

The exponents are compared. If their difference is greater than 32, the operand having the large exponent is loaded into Ah and its exponent into Ae and the subroutine exits. Otherwise, if Ah is greater than Mh, Mh is negated and shifted right a number of places equal to the difference and added to Ah. If Mh is greater than Ah, Ah is shifted right a number of places equal to the difference and Mh is subtracted from Ah. The larger exponent becomes Re.

If overflow occurs, Rh is shifted right one bit position, the sign changed, and Re is incremented. Otherwise,the result is normalized.

4. FLOATING MULTIPLY, SINGLE PRECISION

The fractional parts are multiplied together and the result is normalized. If overflow is on, indicating multiplication of 40000000 times 40000000 occurred, Rh is set to 37777777. The exponents are added.

5. FLOATING DIVIDE, SINGLE PRECISION

The exponents are differenced and the result is incremented by one to compensate for a subsequent right shift of Ah. If Mh is zero, overflow is turned on, and subroutine exits. Otherwise,Ah is right shifted one position and normal division takes place and the result normalized.

6. FLOATING NEGATE, SINGLE PRECISION

If Ah is zero, the subroutine exits. Otherwise,the one's complement of Ah replaces Ah.

```
                                    1  $FSN    P0PD    014400000
00000  0 50 00173                   2  FSN     SKE     =0
00001  0 17 00172                   3          E0R     N1Q23           0NES COMPLIMENT
00002  0 51 00000                   4          BRR     0
                                    5  $FSD    P0PD    014300000
00003  0 37 00014                   6  FSD     STX     TX              SAVE X
00004  0 77 40000                   7          EAX     *0
00005  0 46 00014                   8          XAB
00006  2 54 00000                   9          SUB     ME,2            F0RM RE = AE - ME
00007  0 35 00012                  10          STA     RE
00010  0 46 10012                  11          BAC
00011  0 66 00001                  12          RSH     1               SUPPRESS 0VFL0
00012  2 65 00001                  13          DIV     MH,2            F0RM RH = AH/MH
00013  0 71 00012                  14          LDX     AE
00014  0 40 20001                  15          SKS     020001          TEST 0VFL0
00015  4 51 00167                  16          BRR     0VFL0,4         SET 0VFL0 AND EXIT
00016  4 41 00017                  17          BRX     $+1,4           INCREMENT RE
00017  0 67 10002                  18          N0D     2
00020  0 46 00040                  19          CXB
00021  0 71 00014                  20          LDX     TX              RESTORE X
00022  0 51 00000                  21          BRR     0
                                   22  $FSM    P0PD    014200000
00023  0 37 00014                  23  FSM     STX     TX              SAVE X
00024  0 77 40000                  24          EAX     *0
00025  0 36 00012                  25          STB     AE
00026  2 64 00001                  26          MUL     MH,2            F0RM RH=MH*AH
00027  2 71 00000                  27          LDX     ME,2
00030  0 40 20001                  28          SKS     020001          TEST 0VFL0
00031  0 17 00172                  29          E0R     N1Q23           MAKE P0S 1 AT 0
00032  0 67 10001                  30          N0D     1
00033  0 46 00204                  31          RCH     0204            CXA,CAB
00034  0 55 00012                  32          ADD     AE              F0RM RE = AE + ME
00035  0 46 00014                  33          XAB
00036  0 71 00014                  34          LDX     TX              RESTORE X
00037  0 51 00000                  35          BRR     0
                                   36  $FSS    P0PD    014100000
00040  0 37 00014                  37  FSS     STX     TX              SAVE X
00041  0 35 00013                  38          STA     AH              SAVE AH
```

```
00042  0 36 00012      39          STB     AE                  SAVE AE
00043  0 77 40000      40          EAX     *0
00044  0 46 00200      41          CXA
00045  2 74 00000      42          SKD     ME,2                FORM RE = AE-ME
00046  4 01 00066      43          BRU     FSS2,4              AH GTE MH
00047  0 46 00441      44          RCH     0441                CAX,CXB,CLA
00050  2 50 00001      45          SKE     MH,2                TEST MH=0
00051  4 52 00170      46          SKB     N1Q17,4             TEST RE GTE 32
00052  4 01 00107      47          BRU     FSS4,4              MH GTR AH SPEC CASE
00053  0 46 00014      48          XAB
00054  0 62 00013      49          XMA     AH                  EXCHANGE AH AND RE
00055  0 66 40013      50          RSH     *AH                 SCALE AH
00056  2 54 00001      51  FSS1    SUB     MH,2
00057  2 71 00000      52          LDX     ME,2
00060  0 40 20001      53          SKS     020001              TEST OVFL0
00061  4 01 00163      54          BRU     FIX,4               FIX OVFL0
00062  0 67 10030      55          NOD     24
00063  0 46 00040      56          CXB
00064  0 71 00014      57          LDX     TX                  RESTORE X
00065  0 51 00000      58          BRR     0                   EXIT
00066  0 46 00441      59  FSS2    RCH     0441                CAX,CXB,CLA
00067  0 50 00013      60          SKE     AH                  TEST AH=0
00070  4 52 00170      61          SKB     N1Q17,4             TEST RE GTE 32
00071  4 01 00112      62          BRU     FSS5,4              AH GTE MH SPEC CASE
00072  2 54 00001      63          SUB     MH,2
00073  0 40 20001      64          SKS     020001              TEST OVFL0
00074  0 17 00172      65          EOR     N1Q23               ONES COMPLIMENT
00075  0 46 00022      66          RCH     022                 CBX,CLB
00076  2 66 00000      67          RSH     0,2                 SCALE -MH
00077  0 55 00013      68  FSS3    ADD     AH
00100  0 71 00012      69          LDX     AE
00101  0 40 20001      70          SKS     020001              TEST OVFL0
00102  4 01 00163      71          BRU     FIX,4               FIX OVFL0
00103  0 67 10030      72          NOD     24
00104  0 46 00040      73          CXB
00105  0 71 00014      74          LDX     TX                  RESTORE X
00106  0 51 00000      75          BRR     0                   EXIT
00107  4 52 00170      76  FSS4    SKB     N1Q17,4
00110  4 01 00056      77          BRU     FSS1,4              RE GTE 32
```

```
00111  4 01 00077      78          BRU     FSS3,4          M=0
00112  4 52 00170      79  FSS5    SKB     N1Q17,4
00113  4 01 00077      80          BRU     FSS3,4          RE GTE 32
00114  4 01 00056      81          BRU     FSS1,4          A=0
                        82  $FSA    PBPD    014000000
00115  0 37 00014      83  FSA     STX     TX              SAVE X
00116  0 35 00013      84          STA     AH              SAVE AH
00117  0 36 00012      85          STB     AE              SAVE AE
00120  0 77 40000      86          EAX     *0
00121  0 46 00200      87          CXA
00122  2 74 00000      88          SKD     ME,2            FORM RE=AE-ME
00123  4 01 00136      89          BRU     FSA2,4          AH GTE MH
00124  0 46 00441      90          RCH     0441            CAX,CXB,CLA
00125  2 50 00001      91          SKE     MH,2            TEST MH=0
00126  4 52 00170      92          SKB     N1Q17,4         TEST RH GTE 32
00127  4 01 00155      93          BRU     FSA6,4          MH GTR AH SPEC CASE
00130  0 46 00014      94          XAB
00131  0 62 00013      95          XMA     AH              EXCHANGE RE AND AH
00132  0 66 40013      96          RSH     *AH             SCALE AH
00133  2 55 00001      97  FSA1    ADD     MH,2
00134  2 71 00000      98          LDX     ME,2
00135  4 01 00147      99          BRU     FSA4,4          TEST OVFLO
00136  0 46 00441     100  FSA2    RCH     0441            CAX,CXB,CLA
00137  0 50 00013     101          SKE     AH              TEST AH=0
00140  4 52 00170     102          SKB     N1Q17,4         TEST RH GTE 32
00141  4 01 00160     103          BRU     FSA7,4          AH GTE MH SPEC CASE
00142  2 76 00001     104          LDA     MH,2
00143  0 46 00022     105          RCH     022             CBX,CLB
00144  2 66 00000     106          RSH     0,2             SCALE MH
00145  0 55 00013     107  FSA3    ADD     AH
00146  0 71 00012     108          LDX     AE
00147  0 40 20001     109  FSA4    SKS     020001          TEST OVFLO
00150  4 01 00163     110          BRU     FIX,4           FIX OVFLO
00151  0 67 10030     111          NOD     24
00152  0 46 00040     112  FSA5    CXB
00153  0 71 00014     113          LDX     TX              RESTORE X
00154  0 51 00000     114          BRR     0               EXIT
00155  4 52 00170     115  FSA6    SKB     N1Q17,4
00156  4 01 00133     116          BRU     FSA1,4          RE GTE 32
```

```
00157  4 01 00145      117           BRU     FSA3,4              M=0
00160  4 52 00170      118  FSA7     SKB     N1Q17,4
00161  4 01 00145      119           BRU     FSA3,4              RE GTE 32
00162  4 01 00133      120           BRU     FSA1,4              A=0
00163  0 66 00001      121  FIX      RSH     1
00164  0 17 00171      122           EOR     N1Q0
00165  4 41 00152      123           BRX     FSA5,4              INCREMENT INDEX
00166  4 01 00152      124           BRU     FSA5,4
00167  4 00 00164      125  OVFLO    PZE     $-3,4               OVFLO EXIT
00170  77777740        126  N1Q17    DATA    077777740
       00000000        127  ME       EQU     0
       00000001        128  MH       EQU     01
       00000012        129  RE       EQU     012
       00000012        130  AE       EQU     012
       00000013        131  AH       EQU     013
       00000014        132  TX       EQU     014
00171  40000000        133  N1Q0     DATA    040000000
00172  77777777        134  N1Q23    DATA    -1
                       135           END
00173  00000000
```

IDENTIFICATION:	Double Precision Floating Point Programmed Operator Package
AUTHOR:	Francis F. Welsh, Jr., SDS
ACCEPTED:	July 28, 1964
COMPUTER CONFIGURATION:	Any SDS 920 or 930 computer.
PURPOSE:	To simulate the operation of floating point instructions on the SDS 920 or 930*.
PROGRAMMED OPERATORS:	None
STORAGE:	Instructions and constants: 320 oct, 208 dec
TIMING:	FLN 19 cycles average FLM 92 cycles average FLD 134 cycles average FLS 98 cycles average FLA 82 cycles average
SOURCE LANGUAGE:	SYMBOL

*This package supersedes Catalog N. 203023. While somewhat slower than 203023, this package detects the rare cases which 203023 ignores. Overflow/underflow correction is also somewhat different.

USE:

LDP: (M+1, M) → (A, B)
STD: (A, B) → (M+1, M)
FLA: (A, B) + (M+1, M) → (A, B)
FLS: (A, B) - (M+1, M) → (A, B)
FLM: (A, B) * (M+1, M) → (A, B)
FLD: (A, B) / (M+1, M) → (A, B)
FLN: −(A, B) → (A, B)

The floating point format is consistent with the description in the SDS 920/930 reference manual.

Upon completing floating point operations, the package checks for exponent overflow/underflow. In the case of overflow, the overflow trigger is set and the result is set to

3777 7777 7777 7377, result positive
4000 0000 0000 0377, result negative

When a division overflow occurs, the correct sign is intermediate. Underflows are set to zero and the overflow trigger is set.

METHOD:

The following notation is used below:

$$(A, B) = f_A \cdot 2^{E_A} = f_a \cdot 2^{E_A} + f_b \cdot 2^{E_A-24}$$

$$(M+1, M) = f_M \cdot 2^{E_M} = f_{m+1} \cdot 2^{E_M} + f_m \cdot 2^{E_M-24}$$

FLA(FLS)

$$(A, B) \underset{(-)}{+} (M+1, M) = \begin{cases} f_{A(-)}^{+} f_M \cdot 2^{E_M-E_A}, & |(A, B)| \geq |(M+1, M)| \\ (-)f_M + f_A \cdot 2^{E_A-E_M}, & |(A, B)| < |(M+1, M)| \end{cases}$$

FLM

$$(A, B) * (M+1, M) \doteq \left(f_a f_{m+1} + (f_b f_{m+1} + f_m f_a) \cdot 2^{-24}\right) \cdot 2^{E_A+E_M}$$

METHOD: (Cont)

FLD

$$(A, B) \Big/ (M+1, M) = \frac{f_a + f_b \cdot 2^{-24}}{f_{m+1} + f_m \cdot 2^{-24}} \cdot 2^{E_A - E_M}$$

$$= \frac{f_a + f_b \cdot 2^{-24}}{f_{m+1}\left(1 + \frac{f_m}{f_{m+1}} \cdot 2^{-24}\right)} \cdot 2^{E_A - E_M}$$

$$\doteq \frac{\left(f_a + f_b \cdot 2^{-24}\right)}{f_{m+1}} \left(1 - \frac{f_m}{f_{m+1}} \cdot 2^{-24}\right) \cdot 2^{E_A - E_M}$$

$$= \left(Q + \frac{R \cdot 2^{-24}}{f_{m+1}}\right) \left(1 - \frac{f_m}{f_{m+1}} \cdot 2^{-24}\right) \cdot 2^{E_A - E_M}$$

$$= \left(\left(Q + \left(R - \frac{f_m}{f_{m+1}}\right) \cdot 2^{-24}\right) \Big/ f_{m+1}\right) \cdot 2^{E_A - E_M}$$

```
                00000002          1  X2       EQU     2
00000  0 00 00000                 2  AREG     PZE     0
00001  0 00 00000                 3  XREG     PZE     0
00002  0 00 00000                 4  TEMPE    PZE     0
00003  0 00 00000                 5  TEMP     PZE     0
00004  0 00 00000                 6  FLAG     PZE     0
                                  7  $FLN     PSPD    014400000
00005                             8  FLN      BSS     0
00005  0 46 00014                 9           XAB                     TWOS COMPLEMENT FRACTION
00006  0 17 00277                10           EOR     =077777000
00007  0 55 00300                11           ADD     =01000
00010  0 46 00014                12           XAB
00011  0 17 00301                13           EOR     =-1             TWO RESULTS ARE UNNORMALIZED
00012  0 57 00302                14           ADC     =0
00013  0 50 00303                15           SKE     =060000000      ORIGINAL FRACTION = 1/2
00014  0 40 20001                16           OVT                     FINAL FRACTION =-1
00015  0 01 00017                17           BRU     $+2
00016  0 51 00000                18           BRR     0
00017  0 66 20011                19           RCY     9               FOLLOWING METHOD SETS OVERFLOW IF
00020  0 72 00304                20           SKA     =000020000      EXPONENT UNDERFLOWS OR OVERFLOWS
00021  0 01 00024                21           BRU     $+3
00022  0 55 00305                22           ADD     =00060000       FINAL F=-1. RIGHT NORMALIZE
00023  0 01 00025                23           BRU     $+2
00024  0 54 00306                24           SUB     =00120000       ORIGINAL F=-1. LEFT NORMALIZE
00025  0 67 20011                25           LCY     9
00026  0 40 20001                26           OVT
00027  0 01 00264                27           BRU     OUFLOW          OVER/UNDERFLOW
00030  0 51 00000                28           BRR     0               EXIT
                                 29  $FLM     PSPD    014200000
00031  0 37 00001                30           STX     XREG
00032  0 77 40000                31           EAX     *0
00033  0 35 00000                32           STA     AREG
00034  0 46 00200                33           CXA
00035  0 46 00122                34           STE
00036  0 37 00002                35           STX     TEMPE           SIGN-EXTENDED EXPONENT OF (A,B)
00037  0 46 00410                36           RCH     0410            CAX, CBA
00040  0 66 20002                37           RCY     2               FORCE +
00041  2 64 00001                38           MUL     1,X2
```

```
00042  0 35 00003      39          STA     TEMP
00043  2 75 00000      40          LDB     0,X2
00044  0 46 00200      41          CXA
00045  0 46 00122      42          STE
00046  0 46 00600      43          XXA                     SIGN-EXTENDED EXPONENT OF (M,M+1)
00047  0 63 00002      44          ADM     TEMPE           EXPONENT OF RESULT
00050  0 46 00010      45          CBA
00051  0 66 20002      46          RCY     2               FORCE +
00052  0 64 00000      47          MUL     AREG
00053  0 55 00003      48          ADD     TEMP            CROSS PRODUCT SUM
00054  0 64 00307      49          MUL     =2              EXTEND TO DOUBLE PRECISION
00055  0 36 00003      50          STB     TEMP
00056  0 62 00000      51          XMA     AREG
00057  2 64 00001      52          MUL     1,X2            MULTIPLY MORE SIGNIFICANT HALVES
00060  0 46 00014      53          XAB
00061  0 55 00003      54          ADD     TEMP            ADD CROSS PRODUCT SUM
00062  0 46 00014      55          XAB
00063  0 57 00000      56          ADC     AREG
00064  0 02 20001      57          ROV                     RESET OVERFLOW
00065  0 71 00002      58          LDX     TEMPE
00066  0 73 00310      59          SKG     =040000000      CHECK FOR (-1)*(-1)
00067  0 01 00072      60          BRU     FLMB            POSSIBLE
00070  0 67 10001      61          NOD     1               NORMALIZE RESULT
00071  0 01 00135      62          BRU     FLDB
00072  0 52 00301      63  FLMB    SKB     =-1             IF B=0, RESULT MUST BE RIGHT-NORMALIZ
00073  0 01 00135      64          BRU     FLDB
00074  0 66 20001      65          RCY     1               YES
00075  0 41 00135      66          BRX     FLDB
00076  0 01 00135      67          BRU     FLDB
                       68  $FLD    POPD    014300000
00077  0 37 00001      69          STX     XREG
00100  0 46 00122      70          STE                     BE TO X, 0 TO BE
00101  0 37 00002      71          STX     TEMPE           SIGN-EXTEND EXPONENT (A,B)
00102  0 71 00001      72          LDX     XREG
00103  0 77 40000      73          EAX     *0              PROCURE ARGUMENT ADDRESS
00104  0 66 00002      74          RSH     2
00105  2 65 00001      75          DIV     1,X2
00106  0 40 20001      76          OVT
00107  0 01 00270      77          BRU     DOFLO
```

```
00110  0 35 00003      78          STA     TEMP            MORE SIG. HALF OF QUOTIENT
00111  0 46 00010      79          CBA
00112  0 66 00001      80          RSH     1               SIGN-EXTEND REMAINDER
00113  0 35 00000      81          STA     AREG
00114  2 75 00000      82          LDB     0,X2
00115  0 46 00200      83          CXA
00116  0 46 00122      84          STE                     BE TO XE, 0 TO BE
00117  0 46 00600      85          XXA
00120  0 46 01000      86          CNA                     COMPUTE EXPONENT OF RESULT
00121  0 55 00307      87          ADD     =2              COMPENSATE FOR SHIFT
00122  0 63 00002      88          ADM     TEMPE
00123  0 46 00010      89          CBA
00124  0 66 20002      90          RCY     2
00125  0 46 01000      91          CNA
00126  0 64 00003      92          MUL     TEMP
00127  0 55 00000      93          ADD     AREG
00130  2 65 00001      94          DIV     1,X2            LESS SIG. HALF OF QUOTIENT
00131  0 64 00307      95          MUL     =2              EXTEND TO DOUBLE PRECISION
00132  0 55 00003      96          ADD     TEMP
00133  0 71 00002      97          LDX     TEMPE
00134  0 67 10004      98          NOD     4               NORMALIZE RESULT
00135  0 46 00600      99  FLDB    XXA
00136  0 73 00311     100          SKG     =0377
00137  0 73 00312     101          SKG     =077777377      UNDERFLOW
00140  4 51 00140     102          BRR     $,4
00141  0 46 00600     103          XXA                     NO
00142  0 50 00302     104          SKE     =0              CHECK FOR ZERO
00143  0 46 00140     105  FLAX    LDE                     NO. PACK RESULT
00144  0 71 00001     106          LDX     XREG            YES. LET EXPONENT BE ZERO
00145  0 40 20001     107          OVT
00146  0 01 00264     108          BRU     OUFLOW
00147  0 51 00000     109          BRR     0               EXIT
                       110  $FLS    POPD    014100000
00150  0 35 00004     111          STA     FLAG            SET FLAG NEGATIVE
00151  0 72 00310     112          SKA     =040000000
00152  0 01 00155     113          BRU     FLAPOP
00153  0 17 00301     114          EOR     =-1
00154  0 62 00004     115          XMA     FLAG            NEGATE IF POSITIVE
                       116  $FLA    POPD    014000000
```

```
00155  0 37 00001     117  FLAP0P STX     XREG
00156  0 35 00000     118         STA     AREG
00157  0 36 00003     119         STB     TEMP
00160  0 46 00122     120         STE                    BE TO X, 0 TO B
00161  0 37 00002     121         STX     TEMPE          SIGN-EXTEND EXPONENT (A,B)
00162  0 71 00001     122         LDX     XREG
00163  0 77 40000     123         EAX     *0             PROCURE ARGUMENT ADDRESS
00164  0 46 30003     124         CLR
00165  2 50 00001     125         SKE     1,X2           CHECK FOR M,M+1=0
00166  0 01 00170     126         BRU     $+2            NO
00167  0 01 00254     127         BRU     FLAZ           YES
00170  0 50 00000     128         SKE     AREG           CHECK FOR A,B=0
00171  0 01 00173     129         BRU     $+2            NO
00172  0 01 00210     130         BRU     FLAC           YES
00173  0 46 00200     131         CXA                    CONTINUE
00174  2 75 00000     132         LDB     0,X2
00175  0 46 00122     133         STE                    BE TO X, 0 TO BE
00176  0 46 00600     134         XXA                    SIGN-EXTEND EXPONENT (M,M+1)
00177  0 54 00002     135         SUB     TEMPE          COMPARE EXPONENTS
00200  0 73 00301     136         SKG     =-1
00201  0 01 00233     137         BRU     FLAGM          /A/>/M/
00202  0 72 00313     138         SKA     =077777700     /A/</M/. CHECK FOR DIFFERENCE LARGE
00203  0 76 00314     139         LDA     =39            YES. SET MAXIMUM SHIFT TO 39
00204  0 62 00000     140         XMA     AREG
00205  0 75 00003     141         LDB     TEMP           PROCURE SMALLER ARGUMENT
00206  0 66 40000     142         RSH     *AREG          ALIGN WITH LARGER
00207  0 46 00014     143         XAB
00210  0 53 00004     144  FLAC   SKN     FLAG           TEST WHETHER FLA OR FLS
00211  0 01 00220     145         BRU     FLAE           FLA
00212  0 16 00315     146         MRG     =0777          FLS
00213  2 54 00000     147         SUB     0,X2           PERFORM DPS
00214  0 17 00315     148         EOR     =0777
00215  0 46 00014     149         XAB
00216  2 56 00001     150         SUC     1,X2
00217  0 01 00224     151         BRU     FLAF
00220  0 46 00101     152  FLAE   RCH     0101           0 TO AE
00221  2 55 00000     153         ADD     0,X2           PERFORM DPA
00222  0 46 00014     154         XAB
00223  2 57 00001     155         ADC     1,X2
```

```
00224  2 46 00000      156  FLAF   RCH    0,2
00225  0 37 00004      157         STX    FLAG
00226  0 46 00122      158         STE
00227  0 40 20001      159         OVT                    CHECK FOR OVERFLOW
00230  0 01 00260      160         BRU    OFSET           YES
00231  0 67 10046      161         NOD    38              NO. NORMALIZE RESULT
00232  0 01 00135      162         BRU    FLDB
00233  0 46 01000      163  FLAGM  CNA                    /A/>/M/
00234  0 72 00313      164         SKA    =077777700      SET EXPONENT DIFFERENCE +
00235  0 76 00314      165         LDA    =39             LDA WITH MAX(EXP DIFF, 39)
00236  2 71 00001      166         LDX    1,X2
00237  0 46 00600      167         XXA
00240  2 66 00000      168         RSH    0,X2            ALIGN SMALLER WITH LARGER
00241  0 46 00102      169         RCH    0102            0 TO BE
00242  0 53 00004      170  FLAD   SKN    FLAG            TEST WHETHER FLA/FLS
00243  0 01 00253      171         BRU    FLAH            FLA
00244  0 62 00000      172         XMA    AREG            FLS
00245  0 46 00014      173         XAB
00246  0 62 00003      174         XMA    TEMP            EXCHANGE WITH MEMORY AND
00247  0 54 00003      175         SUB    TEMP            PERFORM DPS
00250  0 46 00014      176         XAB
00251  0 56 00000      177         SUC    AREG
00252  0 01 00224      178         BRU    FLAF
00253  0 46 00014      179  FLAH   XAB                    PERFORM DPA
00254  0 55 00003      180  FLAZ   ADD    TEMP
00255  0 46 00014      181         XAB
00256  0 57 00000      182         ADC    AREG
00257  0 01 00224      183         BRU    FLAF
00260  0 66 00001      184  OFSET  RSH    1               OVERFLOW SET. RIGHT NORMALIZE RESULT
00261  0 17 00310      185         EOR    =040000000
00262  0 41 00143      186         BRX    FLAX
00263  0 01 00135      187         BRU    FLDB
00264  0 52 00316      188  OUFLOW SKB    =0400
00265  0 01 00271      189         BRU    OFLO
00266  0 46 30003      190         CLR                    UNDERFLOW
00267  0 51 00000      191         BRR    0
00270  0 71 00001      192  DOFLO  LDX    XREG
00271  0 16 00317      193  OFLO   MRG    =037740000
00272  0 17 00303      194         EOR    =060000000
```

```
00273  0 66 00046     195          RSH     38                NEGATIVE. SET TO 4000C000.00000377
00274  0 17 00310     196          E0R     =040000000        PoSITIVE. SET TO 37777777.77777377
00275  4 51 00275     197          BRR     $.4
00276  0 51 00000     198          BRR     0
                       199          END

00277  77777000
00300  00001000
00301  77777777
00302  00000000
00303  60000000
00304  00020000
00305  00060000
00306  00120000
00307  00000002
00310  40000000
00311  00000377
00312  77777377
00313  77777700
00314  00000047
00315  00000777
00316  00000400
00317  37740000
```

IDENTIFICATION: Double Precision Arithmetic Programmed Operator Package

AUTHOR: Richard S. Resnick, SDS

ACCEPTED: 15 February 1963

COMPUTER CONFIGURATION: Any SDS 920/930 Computer

PURPOSE: Provides double precision fixed point programmed operators assembled as four separate subroutines.

	Mnemonic	Programmed Operators
1.	DPN	Double precision negate
2.	DPD	Double precision divide
	DPM	Double precision multiply
	DPS	Double precision subtract
	DPA	Double precision add
3.	LDP	Load double precision
4.	STD	Store double precision

Note: LDP, STD operate on single and double floating point numbers also.

PROGRAMMED OPERATORS: None

STORAGE: Instructions and constants:

DPN; 12 oct, 10 dec
DPD, DPM, DPS, DPA: 114 oct, 76 dec
LDP: 6 oct, 6 dec
STD: 6 oct, 6 dec

TIMING: (microseconds)

DPN	72 - 104
DPD	1016 - 1400
DPM	504
DPS	160
DPA	160
LDP	120
STD	160

All times include programmed operator entry and exit.

SOURCE LANGUAGE: SYMBOL

USE:

1. FORMAT

Each double precision number is in two's complement form. The sign bit and most significant 23 magnitude bits occupy the most significant word; the least significant 24 magnitude bits occupy the least significant word.

When multiplications are performed, the binary point can be consistently considered to be located at one of two places:

Immediately to the right of the sign bit. The double precision number is then interpreted as a fraction.

Immediately to the left of the least significant bit. The double precision number is then interpreted as a 46 bit plus sign integer. (The least significant bit is not used in double precision multiply and divide operations.)

For all arithmetic operations except multiplication, the binary point can be consistently considered to be located at any position.

Both A and B registers together constitute a "double accumulator" for double precision programmed operators. The most significant word is contained in the A register and the least significant word in the B register.

The contents of two memory locations, designated by the effective address of a programmed operator and by the effective address plus one, constitute a "double operand".

The least significant word is contained in the location designated by the effective address (M), the most significant word in the following location (M+1).

2. FUNCTIONS

LOAD, DOUBLE PRECISION LDP

The "double operand" is loaded into the "double accumulator", contents of M into B and contents of M+1 into A.

Registers Affected: A, B Timing: 15 cycles

STORE, DOUBLE PRECISION STD

The contents of the "double accumulator" are stored in the "double operand," contents of B into M and contents of A into M+1.

Registers Affected: M, M+1 Timing: 20 cycles

USE: (Cont.) DOUBLE PRECISION ADD DPA

The "double operand" is added to the "double accumulator" and the double precision sum appears in the "double accumulator."

Overflow occurs and the overflow indicator is turned on, if both numbers are of the same sign and the sign of the result is different.

Registers Affected: A, B Timing: 20 cycles

DOUBLE PRECISION SUBTRACT DPS

The "double operand" is subtracted from the "double accumulator" and the double precision difference appears in the "double accumulator".

Overflow occurs and the overflow indicator is turned on, if both numbers are of opposite sign, and the sign of the result does not agree with the original sign of the accumulator.

Registers Affected: A, B Timing: 20 cycles

DOUBLE PRECISION MULTIPLY DPM

The "double operand" is multiplied by the contents of the "double accumulator". The double precision product appears in the "double accumulator."

If the product equals minus one (40000000 00000000 octal) the overflow indicator is turned on. This result occurs if both numbers were minus one.

Registers Affected: A, B Timing: 63 cycles

DOUBLE PRECISION DIVIDE DPD

The contents of the "double accumulator" are divided by the "double operand". The double precision quotient appears in the "double accumulator".

Overflow will occur and the overflow indicator is turned on if:

$$1 \leq \frac{\text{Double Accumulator}}{\text{Double Operand}} < -1$$

In this case the results are not correct.

Registers Affected: A, B Timing: 127 - 175 cycles

USE: (Cont.) DOUBLE PRECISION NEGATE DPN

The two's complement of the contents of the "double accumulator" are placed in the "double accumulator".

The address portion of the instruction is not used.

Registers Affected: A, B Timing: 9 - 13 cycles

3. ACCURACY

Precision of DPA, DPS, and DPN is 47 bits plus sign.
Precision of DPM and DPD is 45 - 47 bits plus sign.

METHOD:

1. NOMENCLATURE

Ah = Most significant word initially in accumulator (A register).

Al = Least significant word initially in accumulator (B register).

Mh = Most significant word of operand in memory (location M+1)

Ml = Least significant word of operand in memory (location M).

Rh = Most significant word of result (A register).

Rl = Least significant word of result (B register).

2. DOUBLE PRECISION ADD

The least significant words are first added and the carry is stored in bit 0 of the index register. The most significant words are then added. The carry is simultaneously added to the sum.

Bit Zero Al	Bit Zero Ml	Bit Zero Rl	Carry
0	0	0	0
0	1	0	1
1	0	0	1
1	1	0	1
0	0	1	0
0	1	1	0
1	0	1	0
1	1	1	1

3. DOUBLE PRECISION SUBTRACT

The least significant words are subtracted and the borrow is stored in bit 0 of the index register. The most significant words are then subtracted. The borrow is simultaneously subtracted from the sum.

METHOD: (Cont.)

Bit Zero Al	Bit Zero Ml	Bit Zero Rl	Borrow
0	0	0	0
1	0	0	0
0	1	0	1
1	1	0	0
0	0	1	1
1	0	1	0
0	1	1	1
1	1	1	1

4. DOUBLE PRECISION MULTIPLY

The algorithm used is:

$$Rh,\ Rl = (Ah + Al)\ (Mh + Ml)$$

$$= (Ah\ Mh + Ah\ Ml + Mh\ al)$$

The term Ml Al is dropped because its significance is less than 2^{-46}.

Al is shifted right two positions with zero shifted into bits zero and one, and it is multiplied by Mh. Similarly Ml is shifted right two positions and multiplied by Ah. The cross products are added and shifted left two positions with sign extended left 23 positions. This result is added, double precision, to the product of Ah times Mh. If the final result is A=40000000, B=00000000; overflow is set.

5. DOUBLE PRECISION DIVIDE

The algorithm used is:

$$Rh,\ Rl = \frac{Ah + Al}{Mh + Ml} = \frac{Ah + Al}{Mh\left[1 + \frac{Ml}{Mh}\right]} = \frac{Ah + Al}{Mh}\left[1 - \frac{Ml}{Mh}\right]$$

The approximation of $\left[1 + \frac{Ml}{Mh}\right]^{-1}$ by $\left[1 - \frac{Ml}{Mh}\right]$ makes use of the taylor series $\left[1 + \frac{a}{b}\right]^{-1} = 1 - \frac{a}{b} + \left[\frac{a}{b}\right]^2 - \left[\frac{a}{b}\right]^3 + \ldots.$

The series can be truncated after the first term because $\frac{Ml}{Mh}$ is less than 2^{-23} and $\left[\frac{Ml}{Mh}\right]$ is then less than 2^{-46}.

METHOD: (Cont.) The algorithm can be further reduced to:

$$Rh,\ Rl = \frac{Ah + Al - \frac{Ah + Al}{Mh} Ml}{Mh}$$

Consider that:

$$\frac{Ah + Al}{Mh} = Q + R$$

then:

$$Rh,\ Rl = Q + \frac{\left[R - \frac{QMl}{Mh}\right]}{Mh}$$

Mh and Ml are first normalized. Ah and Al are shifted left an equal number of places, then right one place and divided by Mh. The quotient and remainder are saved with the remainder shifted right one place. Ml is shifted right two places, with zeros shifted into bits zero and one, negated and multiplied by the quotient. The product is added to the shifted remainder, divided by Mh, and shifted right 22 places with the sign extended. The quotient is then added to the "double accumulator" which is then shifted left one place.

Overflow occurs and the OVERFLOW FLIP FLOP is set if Ah, Al is greater in absolute magnitude than Mh, Ml, or if Mh, Ml equals zero.

6. DOUBLE PRECISION NEGATE

If the least significant word is zero the most significant word is two's complemented. Otherwise the least significant word is two's complemented and the most significant word is one's complemented.

```
                                    1  $IPN    P0P0    013400000
00000   0 52 00011                  2          SKB     =-1             TEST B EQU 0
00001   4 01 00004                  3          BRU     $+3,4
00002   0 46 01000                  4          CNA                     2S COMPLEMENT
00003   0 51 00000                  5          BRR     0               EXIT
00004   0 17 00011                  6          E0R     =-1             1S COMPLEMENT
00005   0 46 00014                  7          XAB
00006   0 46 01000                  8          CNA                     2S COMPLEMENT
00007   0 46 00014                  9          XAB
00010   0 51 00000                 10          BRR     0               EXIT
                                   11          END
00011   77777777
```

```
                                    1  $DPD    P0PD    013300000
00000  0 37 00014                   2  DPD     STX     TX
00001  0 77 40000                   3          EAX     *0
00002  0 35 00013                   4          STA     AH
00003  0 36 00012                   5          STB     AL
00004  2 76 00001                   6          LDA     1,2             MH
00005  2 75 00000                   7          LDB     0,2             ML
00006  2 46 00000                   8          RCH     0,2             CLX
00007  0 67 10060                   9          NOD     48
00010  0 46 00600                  10          XXA
00011  0 46 01000                  11          CNA
00012  0 46 00450                  12          RCH     0450            CAX,CXB,CBA
00013  0 62 00012                  13          XMA     AL
00014  0 46 00014                  14          XAB
00015  0 62 00013                  15          XMA     AH
00016  2 67 00000                  16          LSH     0,2             SCALE AH,AL EQU MH,ML
00017  0 66 00001                  17          RSH     1               AVOID OVFLO ERROR
00020  0 65 00013                  18          DIV     AH              AH,AL/MH
00021  0 35 00016                  19          STA     Q               SAVE QUOTIENT
00022  0 46 10012                  20          BAC
00023  0 66 00001                  21          RSH     1               SCALE REMAINDER
00024  0 62 00012                  22          XMA     AL              EXCHANGE Q AND ML
00025  0 66 20002                  23          RCY     2               SCALE ML RIGHT 2 BITS
00026  0 46 01000                  24          CNA
00027  0 64 00016                  25          MUL     Q               -ML*Q
00030  0 55 00012                  26          ADD     AL              R - ML*Q
00031  0 65 00013                  27          DIV     AH              R - ML*Q/MH
00032  4 64 00071                  28          MUL     1Q22,4          SHIFT 22 PLACES
00033  0 55 00016                  29          ADD     Q               ADD QUOTIENT
00034  0 67 00001                  30          LSH     1               SCALE RESULTS
00035  0 71 00014                  31          LDX     TX
00036  0 51 00000                  32          BRR     0               EXIT
                                   33  $DPM    P0PD    013200000
00037  0 37 00014                  34  DPM     STX     TX
00040  0 77 40000                  35          EAX     *0
00041  0 35 00013                  36          STA     AH
00042  0 46 10012                  37          BAC                     AL INTO A
00043  0 66 20002                  38          RCY     2
```

```
00044  2 64 00001      39            MUL     1,2             AL*MH
00045  0 35 00015      40            STA     R               R EQU AL*MH
00046  2 76 00000      41            LDA     0,2             ML
00047  06624002        42            DATA    06624002
00050  0 64 00013      43            MUL     AH              ML*AH
00051  0 55 00015      44            ADD     R               ML*AH + AL*MH
00052  4 64 00071      45            MUL     1Q22,4          SHIFT 22 PLACES
00053  0 36 00015      46            STB     R               RL
00054  0 62 00013      47            XMA     AH              EXCHANGE QH AND AH
00055  2 64 00001      48            MUL     1,2             AH*MH EQU PH,PL
00056  0 46 00014      49            XAB
00057  0 55 00015      50            ADD     R               RL + PL
00060  0 46 00014      51            XAB
00061  0 57 00013      52            ADC     AH              RH + PH
00062  0 02 20001      53            EOM     020001
00063  0 71 00014      54            LDX     TX
00064  0 73 00112      55            SKG     =040000000
00065  0 52 00113      56            SKB     =-1
00066  0 51 00000      57            BRR     0               EXIT
00067  4 51 00070      58            BRR     OVFLO,4         OVFLO EXIT
00070  4 00 00065      59  OVFLO     PZE     $-3,4           SET OVFLO AND EXIT
00071  00000002        60  1Q22      DATA    02
                       61  $DPS      POPD    013100000
00072  0 37 00014      62  DPS       STX     TX
00073  0 77 40000      63            EAX     *0
00074  0 46 00014      64            XAB
00075  2 54 00000      65            SUB     0,2             AL - ML
00076  0 46 00014      66            XAB
00077  2 56 00001      67            SUC     1,2             AH - MH - CARRY
00100  0 71 00014      68            LDX     TX
00101  0 51 00000      69            BRR     0               EXIT
                       70  $DPA      POPD    013000000
00102  0 37 00014      71  DPA       STX     TX
00103  0 77 40000      72            EAX     *0
00104  0 46 00014      73            XAB
00105  2 55 00000      74            ADD     0,2             AL + ML
00106  0 46 00014      75            XAB
00107  2 57 00001      76            ADC     1,2             AH + MH + CARRY
00110  0 71 00014      77            LDX     TX
```

```
00111   0 51 00000      78              BRR     0               EXIT
        00000012        79      AL      EQU     012
        00000013        80      AH      EQU     013
        00000014        81      TX      EQU     014
        00000015        82      R       EQU     015
        00000016        83      Q       EQU     016
        00004002        84      B4002   EQU     04002           LOGICAL RIGHT SHIFT 2
                        85              END
00112   40000000
00113   77777777
```

```
                        1       $LDP    POPD    013600000
00000   0 46 00040      2               CXB
00001   0 77 40000      3               EAX     *0
00002   2 76 00001      4               LDA     1,2
00003   2 71 00000      5               LDX     0,2
00004   0 46 00060      6               XXB
00005   0 51 00000      7               BRR     0
                        8               END
```

```
                        1       $STD    POPD    013700000
00000   0 37 00014      2               STX     TX
00001   0 77 40000      3               EAX     *0
00002   2 35 00001      4               STA     1,2
00003   2 36 00000      5               STB     0,2
00004   0 71 00014      6               LDX     TX
00005   0 51 00000      7               BRR     0
        00000014        8       TX      EQU     12
                        9               END
```

IDENTIFICATION:	Double Precision Multiply Programmed Operator - DPM
AUTHOR:	Richard S. Resnick, SDS
ACCEPTED:	29 April 1963
COMPUTER CONFIGURATION:	Any SDS 920/930 Computer
PURPOSE:	Provides Double Precision Multiply (DPM) as a programmed operator.
PROGRAMMED OPERATORS:	None
STORAGE:	Instructions and constants: 114 oct, 76 dec
TIMING:	504 microseconds including programmed operator entry and exit.
SOURCE LANGUAGE:	SYMBOL
METHOD:	See Catalog No. 203022-B for description of USE and of METHOD.

IDENTIFICATION:	Double Precision Subtract, Programmed Operator - DPS
AUTHOR:	Richard S. Resnick, SDS
ACCEPTED:	5 March 1963
COMPUTER CONFIGURATION:	Any SDS 920/930 Computer
PURPOSE:	Provides Double Precision Subtract (DPS) as a programmed operator.
PROGRAMMED OPERATORS:	None
STORAGE:	Instructions and constants: 114 oct, 76 dec
TIMING:	160 microseconds including programmed operator entry and exit.
SOURCE LANGUAGE:	SYMBOL
METHOD:	See Catalog No. 203022-B for description of USE and of METHOD.

IDENTIFICATION:	Double Precision Add, Programmed Operator - DPA
AUTHOR:	Richard S. Resnick
ACCEPTED:	6 March 1963
COMPUTER CONFIGURATION:	Any SDS 920/930 Computer
PURPOSE:	Provides Double Precision Add (DPA) as a programmed operator.
PROGRAMMED OPERATORS:	None
STORAGE:	Instructions and constants: 114 oct, 76 dec
TIMING:	160 microseconds including programmed operator entry and exit.
SOURCE LANGUAGE:	SYMBOL
METHOD:	See Catalog No. 203022-B for description of METHOD.

IDENTIFICATION: Load and Store Triple Precision Programmed Operator Package - LTP, STP

AUTHOR: Richard S. Resnick, SDS

ACCEPTED: 6 March 1963

COMPUTER CONFIGURATION: Any SDS 920/930 Computer

PURPOSE: Provides Load Triple Precision and Store Triple Precision Programmed Operators.

PROGRAMMED OPERATORS: None

STORAGE: Assembled as separate subroutines. LTP: 10 oct, 8 dec
STP: 11 oct, 9 dec

TIMING: LTP - 160 μsec STP - 216 μsec

SOURCE LANGUAGE: SYMBOL

USE:

1. FORMAT

The triple precision operand is stored in three consecutive memory locations in the following manner:

The least significant word in M

The next least significant word in M + 1

The most significant word in M + 2

The triple precision accumulator occupies the A register, the B register and location 2. The least significant word is in B, the next least is in A, and the most significant word is in location 2.

2. FUNCTION

Indexing and indirect addressing are possible in both instructions

USE: (Cont.)

LOAD, TRIPLE PRECISION LTP

The triple precision operand is loaded into the triple precision accumulator. The contents of M into the B register, the contents of M+1 into the A register, and the contents of M+2 into location 2.

Registers Affected: A, B, Location 2 Timing: 20

STORE, TRIPLE PRECISION STP

The contents of the triple precision accumulator are stored in the triple precision operand. The contents of B into M, A into M+1, and location 2 into M+2. A, B and location 2 are unchanged.

Registers Affected: M, M+1, M+2 Timing: 27

METHOD: Not applicable.

```
                                    1  $LTP    POPD    016600000
00000  0 46 00200                   2  LTP     CXA
00001  0 77 40000                   3          EAX     *0
00002  2 75 00002                   4          LDB     2,2
00003  0 36 00002                   5          STB     2
00004  2 75 00000                   6          LDB     0,2
00005  2 71 00001                   7          LDX     1,2
00006  0 46 00600                   8          XXA
00007  0 51 00000                   9          BRR     0
                                   10          END
```



```
                                    1  $STP    POPD    016700000
00000  0 37 00012                   2  STP     STX     TX
00001  0 77 40000                   3          EAX     *0
00002  2 35 00001                   4          STA     1,2
00003  2 36 00000                   5          STB     0,2
00004  0 76 00002                   6          LDA     2
00005  2 35 00002                   7          STA     2,2
00006  2 76 00001                   8          LDA     1,2
00007  0 71 00012                   9          LDX     TX
00010  0 51 00000                  10          BRR     0
       00000012                    11  TX      EQU     012
                                   12          END
```

IDENTIFICATION: Load and Store Quadruple Precision Programmed Operator Package - LQP, STQ

AUTHOR: Richard S. Resnick, SDS

ACCEPTED: 7 March 1963

COMPUTER CONFIGURATION: Any SDS 920/930 Computer

PURPOSE: Provides Load Quadruple Precision and Store Quadruple Precision Programmed Operators.

PROGRAMMED OPERATORS: None

STORAGE: Assembled as two separate subroutines. LQP: 12 oct, 10 dec
STQ: 13 oct, 11 dec

TIMING: LQP - 200 μsec SQP - 256 μsec

SOURCE LANGUAGE: SYMBOL

USE: 1. FORMAT

The quadruple precision operand is stored in four consecutive memory locations in the following manner:

The least significant word in M.

The next least significant word in M+1.

The next least significant word in M+2.

The most significant word in M+3.

The quadruple precision accumulator occupies the A register, the B register, location 2 and location 3. The least significant word is in B, the next least is in A, the next in location 2, and the most significant word is in location 3.

USE: (Cont.) 2. FUNCTION

Indexing and indirect addressing are possible in both instructions.

LOAD, QUADRUPLE PRECISION LQP

The quadruple precision operand is loaded in to the quadruple precision accumulator. The contents of M into the B register, the contents of M+1 into the A register, the contents of M+2 into location 2, and the contents of M+3 into location 3.

Registers Affected: A, B, location 2, location 3
Timing: 25

STORE, QUADRUPLE PRECISION STQ

The contents of the quadruple precision accumulator are stored in the quadruple precision operand. The contents of B into M, A into M+1, location 2 into M+2, and location 3 into M+3. A, B, location 2 and location 3 are unchanged.

Registers Affected: M, M+1, M+2, M+3 Timing: 32

METHOD: Not applicable

```
                                    1  $LQP    POPD    016600000
00000  0 46 00040                   2  LQP     CXB
00001  0 77 40000                   3          EAX     *0
00002  2 76 00003                   4          LDA     3,2
00003  0 35 00003                   5          STA     3
00004  2 76 00002                   6          LDA     2,2
00005  0 35 00002                   7          STA     2
00006  2 76 00001                   8          LDA     1,2
00007  2 71 00000                   9          LDX     0,2
00010  0 46 00060                  10          XXB
00011  0 51 00000                  11          BRR     0
                                   12          END
```

```
                                    1  $STQ    POPD    016700000
00000  0 37 00012                   2  STQ     STX     TX
00001  0 77 40000                   3          EAX     *0
00002  2 35 00001                   4          STA     1,2
00003  2 36 00000                   5          STB     0,2
00004  0 76 00002                   6          LDA     2
00005  2 35 00002                   7          STA     2,2
00006  0 76 00003                   8          LDA     3
00007  2 35 00003                   9          STA     3,2
00010  2 76 00001                  10          LDA     1,2
00011  0 71 00012                  11          LDX     TX
00012  0 51 00000                  12          BRR     0
       00000012                    13  TX      EQU     012
                                   14          END
```