3600 3800

COMPUTER SYSTEMS LIBRARY FUNCTIONS

CONTROL DATA CORPORATION

3600 3800

COMPUTER SYSTEMS LIBRARY FUNCTIONS

CONTROL DATA
CORPORATION

Additional copies of this manual may be obtained from the nearest Control Data Corporation Sales office listed on the back cover.

July, 1966
Pub. No. 60056400, B

CONTROL DATA CORPORATION
Documentation Department
3145 PORTER DRIVE
PALO ALTO, CALIFORNIA

© 1966, Control Data Corporation
Printed in the United States of America

CONTENTS

INTRODUCTION

This manual describes the Control Data® 3600 FORTRAN library routines. Library routines may be called directly by the COMPASS calling sequence listed, unless otherwise noted.

To test the accuracy of the double precision functions, several thousand arguments covering the necessary ranges were generated. The functions were evaluated, and the answers compared with the corresponding answers obtained from the CO-OP library routine A3 NBSB DOBLPREC. The latter routine does 91-bit interpretive arithmetic. Since, in general, different algorithms were used and the answers agreed to an average of 82 bits, the accuracy of the double precision functions was considered satisfactory.

Accuracy of the single precision functions was determined by comparing them with the double precision functions in a similar manner. In testing the timing of routines, several thousand appropriate arguments were chosen. The functions were evaluated for these arguments, and the time measured by the TIMEF function. The times quoted within this manual include the jump into and out of the routine, but do not include indexing, loads, and stores of the main programs.

Execution times for complex and double precision functions for the Control Data 3600 have been computed for several thousand arguments using the TIMEF routine. The arguments were chosen to cover uniformly a reasonable portion of the allowable ranges. The range from 0 to $\pi/2$ was timed separately for DSIN and DCOS because of a significant difference in execution time. In all cases, the time includes the jump into the routine and the jump back, but does not include any other instructions in the testing program.

PURPOSE: To obtain the absolute value of the real or integer argument in the A register.

COMPASS CALLING SEQUENCE:

Call by Value (ABSF)

```
            LDA      X or I
L           BRTJ     ($)ABSF, , * or ($)XABSF, ,*
L+1         SLJ      *+1
            00       DICT.
```

Call by Name (Q8QABSF)

```
L           BRTJ     ($)Q8QABSF, , * or ($)Q8QXABSF, ,*
L+1         SLJ      *+2
            01       DICT.
L+2         00       ($)X or ($)I
            00
```

FORTRAN FUNCTION: ABSF(X) or XABSF(I)

NORMAL RETURN: Result is left in the A register.

ERROR MESSAGE: None

STORAGE: 7 locations

ABSF and XABSF are generally compiled on-line.

Compiled coding sequence:

```
            LDA      X or I
+           AJP,PL   * + 1
            ROP,XOR  A,MZ,A
```

ACOSF

PURPOSE: To compute the single precision arccosine of a single precision argument X.

COMPASS CALLING SEQUENCE:

Call by Value (ACOSF)

```
            LDA        X
L           BRTJ       ($)ACOSF, , *
L+1         SLJ        *+1
            00         DICT.
```

Call by Name (Q8QACOSF)

```
L           BRTJ       ($)Q8QACOSF, , *
L+1         SLJ        *+2
            01         DICT.
L+2         00         ($)X
            NOP        0
```

FORTRAN FUNCTION: ACOSF(X)

NORMAL RETURN: Floating point result in the range 0 to π is left in the A register.

ERROR MESSAGE: ERROR DETECTED IN ASIN/ACO

A = (value)   Q = (value)

ARG GT 1.

If the absolute value of the argument is greater than 1.0, this message is printed on the standard output unit, and a normal exit is taken from ACOSF.

STORAGE: 96 locations

ACCURACY: Average 35 bits with a relative error less than 2^{-34} maintained throughout the domain of the argument. See ASINF.

TIMING: Average 178 microseconds for 5000 random values of X, $-1 \leq X \leq 1$.

MATHEMATICAL METHOD: The identity $ACOS(X) = \frac{\pi}{2} - ASIN(X)$ is used.
See ASINF for method description.

PURPOSE: To obtain the imaginary part of a complex number Z.

COMPASS CALLING SEQUENCE:

```
L        BRTJ     ($)AIMAG, , *
L+1      SLJ      *+2
         01       DICT.
L+2      00       ($)Z
         00
```

FORTRAN FUNCTION: AIMAG(Z)

NORMAL RETURN: The imaginary part of Z is left in the A register.

ERROR MESSAGE: None

STORAGE: 9 locations including REAL and CMPLX

COMMENT: In a FORTRAN program, Z must be declared type COMPLEX.

TIMING: 46 microseconds

AIMAG is generally compiled in-line.

Compiled coding sequence:

```
         LDA      Z+1
```

ALOG10

PURPOSE: To compute the common (base 10) logarithm of a floating point argument X.

COMPASS CALLING SEQUENCE:

```
L          BRTJ       ($)ALOG10,,*
L+1        SLJ        *+2
           01         DICT.
L+2        00         ($)X
           NOP        0
```

FORTRAN FUNCTION: ALOG10(X)

NORMAL RETURN: Returns with the result in the A register.

ERROR MESSAGE: Since ALOG10 calls LOGF, see LOGF for error messages of arguments less than or equal to zero.

STORAGE: 16 locations

ACCURACY: 34 bits, relative error increases for X close to 1.0

TIMING: 190 microseconds

MATHEMATICAL METHOD:

$$\text{Log}_{10}(X) = (\text{Log}_{10}(e)^* \text{ Ln } (X)$$

where $\text{Log}_{10}(e) = \text{Log}_{10}\ (2.7182818284)$

$= 0.4342944819032$

LOGF is called to compute Ln (X).

PURPOSE: To compute the single precision arcsine of a single precision argument X.

COMPASS CALLING SEQUENCE:

Call by Value (ASINF)

```
            LDA         X
L           BRTJ        ($)ASINF, , *
L+1         SLJ         *+1
            00          DICT.
L+2         Return
```

Call by Name (Q8QASINF)

```
L           BRTJ        ($)Q8QASINF, , *
L+1         SLJ         *+2
            01          DICT.
L+2         00          ($)X
            NOP         0
```

FORTRAN FUNCTION: ASINF(X)

NORMAL RETURN: Floating point result in the range $-\frac{\pi}{2}$ to $\frac{\pi}{2}$ is left in the A register.

ERROR MESSAGE: ERROR DETECTED IN ASIN/ACO

A = (value) Q = (value)

ARG GT 1.

If the absolute value of the argument is greater than 1.0, this message is printed on the standard output unit and a normal exit is taken from ASINF.

STORAGE: 96 locations

ACCURACY: ASINF was compared with a double precision arcsine routine for 10,000 random numbers between +1 and -1. The error distribution is as follows:

No. of bits in error	-4	-3	-2	-1	0	1	2	3	4
Frequency	0	24	293	2020	5234	2072	339	15	0

TIMING: Average 178 microseconds for 5000 random values of X, $-1 \leq X \leq 1$.

MATHEMATICAL METHOD:

Given an argument $|X|$ between 0. and 1. then

$$ASIN(X) = A + B\bar{X}(k_1 + \bar{X}^2(k_3 + \bar{X}^2(k_5 + \bar{X}^2(k_7 + \bar{X}^2(k_9 + \bar{X}^2(k_{11} + \bar{X}^2(k_{13} + \bar{X}^2(k_{15}\bar{X}^2)))))))$$

This is a relative minimax approximation of degree 15. The domain of X(0 to 1) is partitioned into 4 intervals and the values of A, B, and $\bar{X}$ are assigned as follows:

1. $0 \leq |X| \leq .5$ — $\bar{X} = X$, $A = 0$, $B = 1.$
2. $.5 < |X| \leq .866$ — $\bar{X} = 2X^2 - 1$, $A = \frac{\pi}{4}$, $B = .5$
3. $.866 < |X| \leq .966$ — $\bar{X} = 8X^4 - 8X^2 + 1$, $A = \frac{3\pi}{8}$, $B = .25$
4. $.966 < |X| < 1.$ — $\bar{X} = \left[\frac{1-|X|}{2}\right]^{1/2}$, $A = \frac{\pi}{2}$, $B = -2.$

The octal constants used are:

k_1	= 2000	7777	7777	7777
k_3	= 1775	5252	5253	1306
k_5	= 1774	4631	4506	5155
k_7	= 1773	5556	7611	7211
k_9	= 1772	7550	1757	0610
k_{11}	= 1772	6246	7740	7323
k_{13}	= 1770	7313	2650	2552
k_{15}	= 1773	4362	4544	0473

PURPOSE: To compute the floating point arctangent of the floating-point argument X.

COMPASS CALLING SEQUENCES:

Call by Value (ATANF):

```
            LDA         X
L           BRTJ        ($)ATANF, , *
L+1         SLJ         *+1
            00          DICT.
```

Call by Name (Q8QATANF):

```
L           BRTJ        ($)Q8QATANF, , *
L+1         SLJ         *+2
            01          DICT.
L+2         00          ($)X
            NOP
```

FORTRAN FUNCTION: ATANF(X)

NORMAL RETURN: Floating point result in the range $-\frac{\pi}{2}$ to $\frac{\pi}{2}$ is left in the A register.

ERROR MESSAGE: None

STORAGE: 64 locations

ACCURACY: 34 bits

TIMING: 125 microseconds

MATHEMATICAL METHOD: The Maehly formula for arctangents is used for most arguments. (Control Data Publication No. 516)

$$\tan^{-1}(X) = C + t\ D_0 + t^3\ D_1 + \cfrac{E_1 t^3}{t^2 + D_2 + \cfrac{E_2}{t^2 + D_3}}$$

$.0195 < \lvert X\rvert < \sqrt{2} - 1$	$t = X,\ C = 0$
$\sqrt{2} - 1 < \lvert X\rvert < \sqrt{2} + 1$	$t = \frac{\lvert X\rvert - 1}{\lvert X\rvert + t},\ C = \frac{\pi}{4}$
$\sqrt{2} + 1 < \lvert X\rvert < 3076$	$t = \frac{-1}{\lvert X\rvert},\quad C = \frac{\pi}{2}$

The formula is not used outside the above ranges. Instead, the following equations are used:

$0 \le \lvert X\rvert \le .0195$	$\tan^{-1} X = X\ \frac{-X^3}{3} + \frac{X^5}{5}$
$3076 \le \lvert X\rvert \le \infty$	$\tan^{-1} X = \pm\left(\frac{\pi}{2} - \frac{1}{\lvert X\rvert}\right)$

ATAN2

PURPOSE: To compute the angle when the tangent is given in terms of coordinates, X and Y.

COMPASS CALLING SEQUENCE:

```
L       BRTJ    ($)ATAN2, , *
L+1     SLJ     *+2
        02      DICT.
L+2     00      ($)Y
        00      ($)X
```

FORTRAN FUNCTION: ATAN2 (Y,X)

NORMAL RETURN: The floating point result in the range $-\pi$ to π of the angle in radians is left in the A register.

ERROR MESSAGE: ATAN (0/0) Undefined.

STORAGE: 44 locations including CANG and CANGQ8Q

ACCURACY: 33 bits

TIMING: 212 microseconds average

MATHEMATICAL METHOD: ATAN2 establishes the correct quadrant, and calls ATANF.

PURPOSE: To compute the magnitude or modulus of a complex number; real part in Z, imaginary part in Z + 1.

COMPASS CALLING SEQUENCE:

Call by Value (CMAGQ8Q) (callable from COMPASS only)

```
            DLDA        Z
L           BRTJ        ($)CMAGQ8Q, , *
L+1         SLJ         *+1
            00          DICT.
```

Call by Name (CABS)

```
L           BRTJ        ($)CABS, , *
L+1         SLJ         *+2
            01          DICT.
L+2         00          ($)Z
            NOP         0
```

FORTRAN FUNCTION: CABS (Z)

NORMAL RETURN: The magnitude is left in the A register.

ERROR MESSAGE: None

STORAGE: 18 locations

ACCURACY: 34-35 bits

TIMING: 203 microseconds

MATHEMATICAL METHOD: For the complex number Z, where Z = X + iY

CABS (Z) = |Z| = SQRTF (X*X+Y*Y)

CABS (Z) is evaluated as:

|X| *SQRTF (1. + (Y/X)**2) if |X| ≥ |Y|

or as |Y| *SQRTF (1. + (X/Y)**2) if |Y| > |X|

NOTE: Exponent overflow may occur if |X| or |Y| > $6.35 * 10^{307}$. If overflow occurs, no error indication is given, and CABS (Z) will be set to a very small number (of the order of 10^{-308}).

CANG

PURPOSE: To compute the argument or angle of a complex number, such that $-\pi < \theta \leqq \pi$, θ = argument of Z. The real part is in Z, the imaginary part in Z + 1.

COMPASS CALLING SEQUENCE:

Call by Value (CANGQ8Q) (Callable from COMPASS only)

```
        DLDA      Z
L       BRTJ      ($)CANGQ8Q, , *
L+1     SLJ       *+1
        00        DICT.
```

Call by Name (CANG)

```
L       BRTJ      ($)CANG, , *
L+1     SLJ       *+2
        01        DICT.
L+2     00        ($)Z
        NOP
```

FORTRAN FUNCTION: CANG (Z)

NORMAL RETURN: The angle in radians is left in the A register.

ERROR MESSAGE: None

STORAGE: 44 locations including ATAN2 TIMING: 213 microseconds

ACCURACY: Accuracy depends on ATANF and the arguments used.

MATHEMATICAL METHOD: For the complex number Z

$Z = X + iY = R^*e^{i\theta}$ X, Y, R are real

Argument of Z = $\theta = \text{Tan}^{-1}$ (Y/X) such that $-\pi < \theta \leqq \pi$

1.1 $X > 0$, $\theta = \text{Tan}^{-1} (Y/X)$

1.2 $X < 0$, $Y = 0$, $\theta = \pi$

1.3 $X < 0$, $Y > 0$, $\theta = \text{Tan}^{-1} (Y/X) + \pi$

1.4 $X < 0$, $Y < 0$ $\theta = \text{Tan}^{-1} (Y/X) - \pi$

1.5 $X = 0$

- 1.51 $Y = 0$ θ is undefined, set $\theta = 0$, exit
- 1.52 $Y > 0$ $\theta = \pi/2$
- 1.53 $Y < 0$ $\theta = -\pi/2$

where ATANF is used to compute Tan^{-1} (Y/X).

PURPOSE: To compute the arctangent of a complex number; real part in Z, imaginary part in Z + 1.

COMPASS CALLING SEQUENCE:

```
L       BRTJ    ($)CATAN , , *
L+1     SLJ     *+2
        01      DICT.
L+2     00      ($)Z
        NOP     0
```

FORTRAN FUNCTION: CATAN (Z)

NORMAL RETURN: The real part is left in the A register and the imaginary part in the Q register.

ERROR MESSAGES: See LOGF.

STORAGE: 33 locations

TIMING: 606 microseconds

ACCURACY: Accuracy depends on LOGF, ATANF, and the arguments used.

MATHEMATICAL METHOD:

For the complex number Z

$$u = \text{real part of } \left(\frac{i + Z}{i - Z}\right) \quad \text{u and w are real.}$$

$$w = \text{imaginary part of } \left(\frac{i + Z}{i - Z}\right) \text{ : i.e., } \left(\frac{i + Z}{i - Z}\right) = u + iw$$

$$\text{Tan}^{-1}(Z) = \frac{i}{2} \text{ Ln } \left(\frac{i + Z}{i - Z}\right) = \frac{i}{2} \text{ Ln } (u + iw) = \frac{i}{2} \left(\text{Ln}(R * e^{i\theta})\right)$$

$$\text{Tan}^{-1}(Z) = \frac{i}{2} \text{ Ln } (R) + \frac{i}{2} * i\theta = -\frac{\theta}{2} + \frac{i\text{Ln}(R)}{2}$$

$$\theta = \text{Tan}^{-1}\left(\frac{w}{u}\right) \text{ such that } -\pi < \theta \leq \pi$$

$$R = (u^2 + w^2)^{1/2}$$

LOGF and CANG are called to perform part of this computation.

CCOS

PURPOSE: To compute the cosine of a complex number; real part in Z, imaginary part of Z + 1.

COMPASS CALLING SEQUENCE:

```
L       BRTJ    ($)CCOS, , *
L+1     SLJ     *+2
        01      DICT.
L+2     00      ($)Z
        NOP     0
```

FORTRAN FUNCTION: CCOS (Z)

NORMAL RETURN: The real part is left in the A register and the imaginary part in the Q register.

ERROR MESSAGES: See SINF, COSF, and/or EXPF.

STORAGE: 36 locations including CSIN   TIMING: 624 microseconds

ACCURACY: Depends on SINF, COSF, and EXPF, and the arguments used.

MATHEMATICAL METHOD: For the complex number Z

$$\text{COS}(Z) = \text{SIN}(\pi/2 - Z)$$

See mathematical method of CSIN to compute SIN $(\pi/2 - Z)$.

PURPOSE: To compute the natural exponential of a complex number; the real part in Z, the imaginary part in Z + 1.

COMPASS CALLING SEQUENCE:

```
L       BRTJ    ($)CEXP, , *
L+1     SLJ     *+2
        01      DICT.
L+2     00      ($)Z
        NOP     0
```

FORTRAN FUNCTION: CEXP (Z)

NORMAL RETURN: The real part is left in the A register and the imaginary part in the Q register.

ERROR MESSAGES: See EXPF, SINF and COSF.

STORAGE: 21 locations

TIMING: 489 microseconds

ACCURACY: Depends on EXPF, SINF, COSF and the arguments used.

MATHEMATICAL METHOD: For the complex number Z

$Z = X + iY$, X, Y are real

$e^Z = e^{X+iY} = e^X * e^{iY} = e^X(\mathrm{COS}(Y) + i\ \mathrm{SIN}(Y))$

$e^Z = e^X \mathrm{COS}(Y) + ie^X\ \mathrm{SIN}(Y)$

EXPF, COSF, and SINF are called to perform this computation.

CLOG

PURPOSE: To compute the natural logarithm of a complex number; the real part in Z, the imaginary part in Z + 1.

COMPASS CALLING SEQUENCE:

```
L       BRTJ    ($)CLOG, , *
L+1     SLJ     *+2
        01      DICT.
L+2     00      ($)Z
        NOP     0
```

FORTRAN FUNCTION: CLOG (Z)

NORMAL RETURN: The real part is left in the A register and the imaginary part in the Q register.

ERROR MESSAGES: See LOGF.

STORAGE: 20 locations

ACCURACY: Depends on LOGF, SQRTF, ATANF, and the arguments used.

TIMING: 550 microseconds

MATHEMATICAL METHOD: For the complex number Z

$Z = X + iY = R*e^{i\theta}$ X, Y and R are real

$R = (1 + (Y/X)^2)1/2$

$\theta = \text{Tan}^{-1} (Y/X)$, such that $-\pi < \theta \leqq \pi$

$\text{Ln}(Z) = \text{Ln}(R*e^{i\theta}) = \text{Ln}(R) + i\theta$

CABS, LOGF, and CANG are called to compute R, Ln(R), and θ , respectively.

PURPOSE: To create a complex number from two real numbers, X and Y.

COMPASS CALLING SEQUENCE:

```
L       BRTJ    ($)CMPLX, , *
L+1     SLJ     *+2
        02      DICT.
L+2     00      ($)X
        00      ($)Y
```

FORTRAN FUNCTION: CMPLX (X,Y)

NORMAL RETURN: X is left in the A register, Y is left in the Q register.

ERROR MESSAGE: None

STORAGE: 9 locations including REAL and AIMAG

TIMING: 54 microseconds

CMPLX is generally compiled in-line.

Compiled coding sequence:

```
LDA     X
LDQ     Y
```

CONJG

PURPOSE: To compute the conjugate of a complex number Z; the real part is in Z, the imaginary part in Z + 1.

COMPASS CALLING SEQUENCE:

```
L       BRTJ    ($)CONJG, , *
L+1     SLJ     *+2
        01      DICT.
L+2     00      ($)Z
        NOP     0
```

FORTRAN FUNCTION: CONJG(Z)

NORMAL RETURN: The real part of the conjugate is left in the A register; the imaginary part in the Q register.

ERROR MESSAGE: None

STORAGE: 4 locations

TIMING: 50 microseconds

ACCURACY: Exact

MATHEMATICAL METHOD: For the complex number Z

$Z = X + i * Y$ X and Y are real

Conjugate of $Z = X - i * Y$

CONJG is generally compiled in-line.

Compiled coding sequence:

```
LDA     Z
LQC     Z+1
```

PURPOSE: To compute the cosine of a floating point argument X in radians.

COMPASS CALLING SEQUENCE:

Call by Value (COSF)

```
         LDA     X
L        BRTJ    ($)COSF, , *
L+1      SLJ     *+1
         00      DICT.
```

Call by Name (Q8QCOSF)

```
L        BRTJ    ($)Q8QCOSF, , *
L+1      SLJ     *+2
         01      DICT.
L+2      00      ($)X
         NOP     0
```

FORTRAN FUNCTION: COSF(X)

NORMAL RETURN: Floating point result is left in the A register.

ERROR MESSAGE: ERROR DETECTED IN SIN/COS

A = (value)   Q = (value)

ARG GT 2**36.

If the absolute value of the argument exceeds 2^{36}, this message is printed on the standard output unit, and the absolute value of the argument is returned in the A register.

STORAGE: 139 locations

ACCURACY: Average accuracy in the range $(-\frac{\pi}{2}, \frac{\pi}{2})$ exceeds 35 bits; the worst known case is 34 bits. When $|x| > \frac{\pi}{2}$, the relative error is less than 2^{-35}.

TIMING: 122 microseconds

MATHEMATICAL METHOD:

The method of partitioned polynomials, described in Control Data Technical Report 52 is used. Briefly, for $|x| >$ approximately 40°, a polynomial in x is evaluated; for approximately 40° $\angle |x| \angle 90°$, a polynomial in $(\frac{\pi}{2} - x)$ is evaluated; in both cases, the actual angle determines how many terms in each polynomial are evaluated.

COTF

PURPOSE: To compute the cotangent of the floating point argument X in radius.

COMPASS CALLING SEQUENCE:

Call by value (COTF)

```
            LDA         X
L           BRTJ        ($)COTF,,*
L+1         SLJ         *+1
            00          DICT.
```

Call by name (Q8QCOTF)

```
L           BRTJ        ($)Q8QCOTF,,*
L+1         SLJ         *+2
            01          DICT.
L+2         00          ($)X
            NOP         0
```

FORTRAN FUNCTION: COTF (X)

NORMAL RETURN: Floating point result is in the A register

ERROR MESSAGE: IF the argument is zero:

ERROR DETECTED IN COTF

A = (value) Q = (value)

ARG = 0

STORAGE: 19 locations

ACCURACY: See TANF

TIMING: 20 microseconds + TANF timing

MATHEMATICAL METHOD: The cotangent is evaluated by using the equation

$$COTF(X) = TANF(\pi/2-X)$$

PURPOSE: To compute the sine of a complex number; the real part is in Z, the imaginary part in Z + 1.

COMPASS CALLING SEQUENCE:

```
L      BRTJ   ($)CSIN, , *
L+1    SLJ    *+2
       01     DICT.
L+2    00     ($)Z
       NOP    0
```

FORTRAN FUNCTION: CSIN (Z)

NORMAL RETURN: The real part is left in the A register and the imaginary part in the Q register.

ERROR MESSAGE: See SINF, COSF and/or EXPF.

STORAGE: 36 locations including CCOS.

ACCURACY: Accuracy depends on SINF, COSF, and/or EXPF

TIMING: 550 microseconds

MATHEMATICAL METHOD: For the complex number Z

$$Z = X + iY \qquad X,\ Y \text{ are real}$$

$$\text{CSIN }(Z) = \frac{e^{Y} + e^{-Y}}{2} * \text{SIN }(X) + i\left(\frac{e^{Y} - e^{-Y}}{2}\right) * \text{COS }(X)$$

CSQRT

PURPOSE: To compute the square root of a complex number; the real part is in Z, the imaginary part in Z + 1.

COMPASS CALLING SEQUENCE:

```
L       BRTJ    ($)CSQRT, , *
L+1     SLJ     *+2
        01      DICT.
L+2     00      ($)Z
        NOP     0
```

FORTRAN FUNCTION: CSQRT (Z)

NORMAL RETURN: The real part is less than or equal to zero. It is left in the A register and the imaginary part in the Q register.

ERROR MESSAGE: None

STORAGE: 20 locations

ACCURACY: Approximately 35 bits

TIMING: 348 microseconds

MATHEMATICAL METHOD: For the complex number X, let Z = X + iY,

$$W = \sqrt{Z} \quad = U + iV$$

Compute $C = \sqrt{C\ |X| \ + \ \sqrt{X^2 + Y^2/2}} \qquad D = \frac{Y}{2C}$

Then U = C and V = D if $X \geq 0$

$U = |D|$ and $V = \text{sign}(Y) * C$ if $X < 0$

NOTE: Exponent overflow may occur if $|X|$ or $|Y| > 3.73 * 10^{307}$, CSQRT (Z) will be set to a number of the order $(10^{154}, 10^{154})$. If overflow occurs and if $|Z| < 8.98 * 10^{307}$, CSQRT (Z) is set to (0, 0).

PURPOSE: To compute the cube root of a real argument X.

COMPASS CALLING SEQUENCE:

Call by Value (CUBERTF)

```
            LDA       X
L           BRTJ      ($)CUBERTF, , *
L+1         SLJ       *+1
            00        DICT.
```

Call by Name (Q8QCUBER)

```
L           BRTJ      ($)Q8QCUBER, , *
L+1         SLJ       *+2
            01        DICT.
L+2         00        ($)X
            NOP       0
```

FORTRAN FUNCTION: CUBERTF (X)

NORMAL RETURN: The floating point result is left in the A register.

ERROR MESSAGE: None

STORAGE: 61 locations

ACCURACY: 35 bits average. Most cases are exact; worst known error is 2 bits

TIMING: 196 microseconds

MATHEMATICAL METHOD: $X^{1/3} = (2^{3n+k} * f)^{1/3}$ where $1/2 \leq f < 1$, n and k are like-signed integers, and $k = 0, \pm 1, \pm 2$. An initial approximation to $f^{1/3}$ is given $f_0 = A + Bf + Cf^2$, where A = .6937488349, B = .4944862032, and C = -.1886870774. Two Newtonian iterations of the form $f_{i+1}(2/3)\ (f_i + f/2f_i^2)$ are done, the first in floating point, and the second using fractional arithmetic to avoid a rounding error. f_2 is fractionally multiplied by 2^k, the new exponent is packed, and the result returned in the A register.

DABS

PURPOSE: To obtain the absolute value of the double precision argument X.

COMPASS CALLING SEQUENCE:

```
L        BRTJ     ($)DABS,,*
L+1      SLJ      *+2
         01       DICT.
L+2      00       ($)X
         NOP      0
```

FORTRAN FUNCTION: DABS (X)

NORMAL RETURN: Result is left in the AQ register.

ERROR MESSAGE: None

STORAGE: 5 location

TIMING: 40 microseconds

DABS is generally compiled in-line.

Compiled coding sequence:

```
         LDA        X
         AJP,PL     * + 2
+        ROP,XOR    A,MZ,A
         ROP,XOR    Q,MZ,Q
```

PURPOSE: To compute the double precision arctangent of a double precision argument X.

COMPASS CALLING SEQUENCE:

```
L       BRTJ    ($)DATAN, , *
L+1     SLJ     *+2
        01      DICT.
L+2     00      ($)X
        NOP     0
```

FORTRAN FUNCTION: DATAN (X)

NORMAL RETURN: The result is left in the A and Q registers.

ERROR MESSAGES: None

STORAGE: 107 locations

TIMING: 898 microseconds

ACCURACY: Average 83 bits, maintains at least 82 bits accuracy.

MATHEMATICAL METHOD:[1]

For $0 \leqq x \leqq 0.057$, a continued fraction expansion is used:

1) $$\arctan x = x/b_o + \sum_{i=1}^{M} x^2/b_i$$

where M=6 for $0 \leqq |x| < 0.024$ and M=7 for $0.024 \leqq |x| < 0.057$

For $0.057 \leqq |x|$, the identity used is:

2) $$\arctan y = -\arctan y_o + \arctan \frac{y + y_o}{1 - yy_o}$$

For $0.057 \leqq |x| < 1$, let y=x. For $1 \leqq |x|$, let y=1/x, then $\arctan x = \frac{\pi}{2} - \arctan y$.

If $0 \leqq |y| < \sqrt{2} - 1$, let $y_o = -\tan(\pi/16)$.

If $\sqrt{2} - 1 \leqq |y| \leqq 1$, let $y_o = -\tan(3\pi/16)$.

DATAN

Let $W = \frac{y+y_o}{1-yy_o}$. Then

$$\arctan W = \frac{W}{/b_o} + \sum_{i=1}^{13} \frac{W^2}{/b_i} = \cfrac{W}{b_o + \cfrac{W^2}{b_1 + \cfrac{W^2}{b_2 + \cfrac{W^2}{b_3 + \cdots}}}}$$

[1] See "Some Basic 1604 Mathematical Sub-Routines, with Analysis of Chebyshev Polynomials Modified Taylor Series, and Continued Fractions", Control Data Corporation publication 61 for a discussion of the subdivisions of the range; and Ralston and Wilf's "Mathematical Method For Digital Computers" p. 30, for a discussion of continued fractions; also D. Teichroew's "Use of Continued Fractions in High Speed Computers", NBS report 1514, NBS Project 3011-60-0002, March 1952.

The octal constants are:

b_0 =	2001400000000000	0000000000000000
b_1 =	2002600000000000	0000000000000000
b_2 =	2001500000000000	0000000000000002
b_3 =	2002616161616161	6161616161627261
b_4 =	2001504000000000	0000000015365211
b_5 =	2002620376671352	3035475047030314
b_6 =	2001505000000000	0013510547076322
b_7 =	2002621173230435	1112472750662701
b_8 =	2001505310000213	2616107512023450
b_9 =	2002621444355714	6265443712724534
b_{10} =	2001505457754524	5260666322154061
b_{11} =	2002622437536621	6257645425724336
b_{12} =	2001533551505605	0740405770000000

PURPOSE: To compute the angle when the tangent is given in terms of the double precision arguments, X and Y.

COMPASS CALLING SEQUENCE:

```
L        BRTJ     ($)DATAN2,,*
L+1      SLJ      *+2
         02       DICT.
L+2      00       ($)Y
         00       ($)X
```

FORTRAN FUNCTION: DATAN2 (Y, X)

NORMAL RETURN: The double precision result of the angle in radians is left in the AQ register.

ERROR MESSAGE: ARCTAN(0/0) UNDEFINED.

Written on the standard output unit; a normal exit is taken from DATAN2 when both arguments are zero since the arctangent (0/0) is undefined.

STORAGE: 57 locations

ACCURACY: 82 bits

MATHEMATICAL METHOD: To compute

$TAN^{-1}(Y/X)$ such that $-\pi < \theta \leqq \pi$

1.1 $X > 0, \theta = Tan^{-1}(Y/X)$

1.2 $X < 0, Y = 0, \theta = \pi$

1.3 $X < 0, Y > 0, \theta = Tan^{-1}(Y/X) + \pi$

1.4 $X < 0, Y < 0, \theta = Tan^{-1}(Y/X) - \pi$

1.5 $X = 0$

1.51 $Y = 0$, θ is undefined, take error exit

1.52 $Y > 0, \theta = \pi/2$

1.53 $Y < 0, \theta = \pi/2$

DATAN is used to compute $Tan^{-1}(Y/X)$.

DBLE

PURPOSE: To convert a single precision floating point argument X, to double precision.

COMPASS CALLING SEQUENCE:

Call by value (DBLE)

```
L         LDA       X
L+1       BRTJ      ($)DBLE,,*
L+2       SLJ       *+1
```

Call by name (Q8QDBLE)

```
L         BRTJ      ($)Q8QDBLE,,*
L+1       SLJ       *+2
          01        DICT.
L+2       00        ($)X
          NOP       0
```

FORTRAN FUNCTION: DBLE (X)

NORMAL RETURN: The double precision result is left in the AQ register.

ERROR MESSAGE: None

STORAGE: 24 locations including SNGL

ACCURACY: 84 bits

DBLE is generally compiled in-line.

Compiled coding sequence:

```
          LDA       X
          LRS       48
          LLS       48
```

PURPOSE: To compute the double precision cosine of a double precision argument X in radians.

COMPASS CALLING SEQUENCE:

```
L       BRTJ    ($)DCOS, , *
L+1     SLJ     *+2
        01      DICT.
L+2     00      ($)X
        NOP     0
```

FORTRAN FUNCTION: DCOS (X)

NORMAL RETURN: The result is left in the A and Q registers.

ERROR MESSAGE: ERROR DETECTED IN DSIN/DCO

A = (value)   Q = (value)

X GR THAN 2**83.

When the argument is greater than 2^{83}, this message is written on standard output and a normal exit is taken from DCOS.

STORAGE: 122 locations including DSIN

ACCURACY: Average 83 bits; worst known relative error is 4×2^{-84}.

TIMING: DCOS (0 to $\pi/2$) 518 microseconds

DCOS ($\pi/2$ and up) 645 microseconds

MATHEMATICAL METHOD: The cosine is computed using the identity Cos R=Sin($\frac{\pi}{2}$ - R).

Refer to method in the DSIN description.

DCUBRT

PURPOSE: To compute the double precision cube root of a double precision argument X.

COMPASS CALLING SEQUENCE:

```
L        BRTJ     ($)DCUBRT, , *
L+1      SLJ      *+2
         01       DICT.
L+2      00       ($)X
         NOP      0
```

FORTRAN FUNCTION: DCUBRT (X)

NORMAL RETURN: The result is left in the A and Q registers.

ERROR MESSAGE: None

STORAGE: 55 locations

ACCURACY: Average 83 bits

TIMING: 331 Microseconds

MATHEMATICAL METHOD: For $|X| = 2^{3n} * f \qquad 1/8 \leqq f < 1$

The coefficients for the united Chebyshev approximation are those generated by Stoer's version of the Remez algorithm[1]:

$$Y_0 = A + B * f + \frac{C}{D+f}$$

A	0.8898	7969
B	0.2438	0614
C	- 0.1698	8794
D	0.2795	9498

The Newtonian iteration is of the form

$$Y_{i+1} = \frac{Y_i}{2} + \frac{3f}{4Y_i^2 + \frac{f}{Y_i/2}}$$

Y_0 and Y_1 are evaluated with single precision; Y_2 with double precision

Then $\sqrt[3]{X} = 2^n * Y_2 * \text{sign}(X)$

NOTE: Refer to CO-OP routine B4 ANL ANL B453 DCUBRT.

---

[1] J. Stoer, "A Direct Method for Chebyshev Approximation by Rational Functions, JACM, Vol. 11, No. 1 (Jan. 1964), pp. 59-69.

PURPOSE: To compute the double precision natural exponential of a double precision argument X.

COMPASS CALLING SEQUENCE:

```
L       BRTJ    ($)DEXP, , *
L+1     SLJ     *+2
        01      DICT.
L+2     00      ($)X
        NOP     0
```

FORTRAN FUNCTION: DEXP (X)

NORMAL RETURN: The result is left in the A and Q registers.

ERROR MESSAGE: ERROR DETECTED IN DEXP

A = (value)   Q = (value)

ARG GR THAN 709.

When argument is greater than 709.089, EXP(X) will exceed the largest floating point number. This message is printed on standard output and a normal exit is taken from DEXP.

STORAGE: 88 locations   TIMING: 482 microseconds

ACCURACY: Average 82 bits; worst known relative error is 4×2^{-84}.

MATHEMATICAL METHOD: $X = M\,Ln2 + R$ M = nearest integer to X/ Ln2.

$|R| \leq Ln2/2$

$$\frac{X}{Ln2/2} = 2M + \frac{R}{Ln2/2}$$

$$e^X = 2^M e^R$$

where: $e^R = \frac{P+S}{P-S}$

$$P = A_0 + R^2 \quad (A_2 + R^2 (A_4 + R^2 (A_6 + R^2 * A_8)))$$

$$S = R(A_1 + R^2 (A_3 + R^2 (A_5 + R^2 * A_7)))$$

This rational approximation is a Pade approximation, see Ralston and Wilf, "Mathematical Methods for Digital Computers", p. 13.

This approximation is also a special case of the Modified Taylor Series of Obrechkoff, see "Some Basic 1604 Mathematical Subroutines, With Analysis of Chebyshev Polynomials, Modified Taylor Series, and Continued Fractions", Control Data pub. 61, p. AA19.

The octal constants used are:

A_0 =	20357567	02100000	00000000	00000000
A_1 =	20345270	14252363	22265012	00634545
A_2 =	20316736	51040243	74724732	70047333
A_3 =	20265374	55112340	05736560	14746154
A_4 =	20225667	32357466	45736172	43317057
A_5 =	20164251	50626015	02447135	14134622
A_6 =	60114273	66467055	13742110	25554071
A_7 =	20035421	67776704	21366464	13166547
A_8 =	17736644	04650356	37521044	40013400

PURPOSE: To determine the difference between two floating point numbers, X_1 and X_2.

COMPASS CALLING SEQUENCE:

Call by Value (DIMF)

```
            LDA      X1
            LDQ      X2
L           BRTJ     ($)DIMF, , *
L+1         SLJ      *+1
            00       DICT.
```

Call by Name (Q8QDIMF)

```
L           BRTJ     ($)Q8QDIMF, , *
L+1         SLJ      *+2
            00       DICT.
L+2         00       ($)X1
            00       ($)X2
```

FORTRAN FUNCTION: DIMF(X_1,X_2)

NORMAL RETURN: If $X_1 > X_2$, the floating point difference is left in the A register.

If $X_1 \leq X_2$, zero is left in the A register.

ERROR MESSAGE: ERROR DETECTED IN DIMF

A = (value) Q = (value)

OVERFLOW ERROR.

This message is written on the standard output unit and a normal exit is taken when the exponent of the difference, $X_1 - X_2$, exceeds 308.

STORAGE: 33 locations

ACCURACY: Depends on relative magnitude of X_1 and X_2

DIMF is generally compiled in-line.

Compiled coding sequence:

```
            LDA      X1
            FSB      X2
+           AJP, PL  *+1
            ENA      0
```

DLOG

PURPOSE: To compute the double precision natural logarithm of a double precision floating point argument X.

COMPASS CALLING SEQUENCE:

L	BRTJ	($)DLOG, , *
L+1	SLJ	*+2
	01	DICT.
L+2	00	($)X
	NOP	0

FORTRAN FUNCTION: DLOG (X)

NORMAL RETURN: The result is left in the A and Q registers.

ERROR MESSAGE: ERROR DETECTED IN DLOG

A = (value)　　Q = (value)

ARG = 0/NEG.

This message is written on standard output and a normal exit is taken from DLOG when the argument is zero or negative.

STORAGE: 124 locations　　TIMING: 442 microseconds

ACCURACY: Average 82 bits; close to 1.0 (from .88 to 1.2), an absolute error of 2^{-82} is maintained but the relative error may be larger in this range.

MATHEMATICAL METHOD: $X = 2^n * f \qquad 1/2 \leq f < 1$

$$Ln\ f = (A_1 + A_2 t^3 + A_3 t^5 + A_4 t^7 + A_5 t^9 + A_6 t^{11}) - \alpha * Ln2$$

$$t = (f - B) / (f + B)$$

$$Ln\ X = Ln\ f + n*Ln2$$

The range is subdivided into eight intervals yielding:

$$\text{for} \left(\frac{1}{2}\right)^{(I-1)/8} \geq f > \left(\frac{1}{2}\right)^{(I/8)}, \text{ then } \alpha = \frac{2*I-1}{16} \text{ and } B = \left(\frac{1}{2}\right)^{\alpha}$$

This makes use of logarithmic subdivision with n=8, see Maehly, Hans J. "Approximations for Control Data 1604". The approximation above is a Taylor power series of degree 15 telescoped into an 11th degree series by the use of Chebyshev polynomials. See "Some Basic 1604 Mathematical Subroutines with analysis of Chebyshev Polynomials, Modified Taylor Series, and Continued Fractions", Control Data pub. 61, p. AA20. Also see Hammings "Numerical Analysis".

The octal constants used are:

A_1 =	20017777	77777777	77777777	77777703
A_2 =	20005252	52525252	52525253	05377105
A_3 =	17766314	63146314	63113252	20625157
A_4 =	17764444	44444452	33021367	57217246
A_5 =	17757070	70570326	44106520	16001122
A_6 =	17755646	70115712	66255342	54525145

DLOG10

PURPOSE: To compute the common (base 10) logarithm of a double precision argument X.

COMPASS CALLING SEQUENCE:

```
L        BRTJ     ($)DLOG10,,*
L+1      SLJ      *+2
         01       DICT.
L+2      00       ($)X
         NOP      0
```

FORTRAN FUNCTION: DLOG10 (X)

NORMAL RETURN: The double precision result is left in the AQ register.

ERROR MESSAGE: Since DLOG10 calls DLOG, see DLOG for error messages for arguments less than or equal to zero.

STORAGE: 17 locations

ACCURACY: Average 82 bits, close to 1.0 (.88 to 1.2) an absolute error of 2^{-82} is maintained but the relative error may be larger in this range.

MATHEMATICAL METHOD: LOG_{10} (X) = (Log_{10} (e)) * Log_e (X)

Log_{10} (e) = Log_{10} (2.71828...)

= 0.4342944819032518276511289...

DLOG is called to compute Log_e (X).

PURPOSE: To determine maximum value of the double precision arguments X_1, $X_2, \ldots,$ X_n.

COMPASS CALLING SEQUENCE:

L	BRTJ	($)DMAX1, , *
L+1	SLJ	*+m
	n	DICT.
L+2	00	($)$X_1$
	00	($)$X_2$
	.	
	.	
	.	
	00	($)$X_n$

$m = (n + 1)/2 + 1$, n = number of parameters

FORTRAN FUNCTION: DMAX1 (X_1, $X_2, \ldots,$ X_n)

NORMAL RETURN: The largest double precision argument is left in the AQ register.

ERROR MESSAGE: None

STORAGE: 47 locations for DMAX1 and DMIN1

PURPOSE: To determine the minimum value of the double precision arguments X_1, $X_2, \ldots,$ X_n1

COMPASS CALLING SEQUENCE:

```
L       BRTJ    ($)DMIN1,,*
L+1     SLJ     *+m
        n       DICT.
L+2     00      ($)X1
        00      ($)X2
        .       .
        .       .
        .       .
        00      ($)Xn
```

m + (n + 1)/2 + 1, n = number of parameters

FORTRAN FUNCTION: DMIN1 $(X_1, X_2, \ldots\ldots, X_n)$

NORMAL RETURN: Returns with the minimum double precision argument in the AQ register.

ERROR MESSAGE: None

STORAGE: 47 locations for DMIN1 and DMAX1

PURPOSE: To compute the value of double precision argument X_1 modulo X_2.

COMPASS CALLING SEQUENCE:

L	BRTJ	($)DMOD, , *
L+1	SLJ	*+2
	02	DICT.
L+2	00	($)$X_1$
	00	($)$X_2$

FORTRAN FUNCTION: DMOD (X_1, X_2)

NORMAL RETURN: The value of X_1 modulo X_2 is left in the AQ register. The result is negative only if X_1 and X_2 have unlike signs.

ERROR MESSAGE: ARG2 = 0, ARG1/0 UNDEFINED.

Written on the standard output unit if $X_2 = 0$ and a normal exit is taken.

STORAGE: 55 locations

ACCURACY: Depends on the size of the arguments and the subtraction, $X_1 - \left[X_1/X_2\right] * X_2$.

MATHEMATICAL METHOD: The result is computed as $X_1 - \left[X_1/X_2\right] * X_2$, where only the integer part of the bracketed quantity is used.

DSIGN

PURPOSE: To transfer the sign of double precision argument X_2 to the absolute value of double precision argument X_1.

COMPASS CALLING SEQUENCE:

L	BRTJ	($)DSIGN, , *
L+1	SLJ	*+2
	02	DICT.
L+2	00	($)$X_1$
	00	($)$X_2$

FORTRAN FUNCTION: DSIGN(X_1, X_2)

NORMAL RETURN: Result is returned in the AQ register.

ERROR MESSAGE: None

STORAGE: 19 locations

ACCURACY: Exact

PURPOSE: To compute the double precision sine of a double precision argument X in radians.

COMPASS CALLING SEQUENCE:

L	BRTJ	($)DSIN, , *
L+1	SLJ	*+2
	01	DICT.
L+2	00	($)X
	NOP	0

FORTRAN FUNCTION: DSIN(X)

NORMAL RETURN: The result is left in A and Q registers.

ERROR MESSAGE: ERROR DETECTED IN DSIN/DCO

A = (value) Q = (value)

X GR THAN 2**83.

When the argument is greater than 2^{83}, this message is written on standard output and a normal exit is taken from DSIN.

STORAGE: 122 locations including DCOS

ACCURACY: Average 83 bits, worst known relative error is $4 * 2^{-84}$.

TIMING: DSIN (0 to $\pi/2$) 529 microseconds; DSIN ($\pi/2$ and up) 652 microseconds

MATHEMATICAL METHOD: The sine is computed by using the approximation:

$$SIN(R) = R(S_1 + R^2(S_2 + R^2(S_3 + R^2(S_4 + R^2(S_5 + R^2(S_6 + R^2(S_7 + R^2(S_8 + R^2(S_9 + R^2(S_{10} + R^2(S_{11} + R^2 * S_{12})))))))))))$$

where $0 < |R| < \pi/2$.

This is a power series of order 27 telescoped into a series of order 23. The telescoping is done through the use of Chebyshev Polynomials. See "Some Basic 1604 Mathematical Subroutines with analysis of Chebyshev Polynomials, Modified Taylor Series, and Continued Fractions", Control Data pub. 61, p. AA20. Also see Hammings "Numerical Analysis".

The octal constants used are:

S_1	= 20014000	00000000	00000000	00000000
S_2	= 60022525	25252525	25252525	25252544
S_3	= 17714210	42104210	42104210	42076135
S_4	= 60141377	13771377	13771377	20310533
S_5	= 17555616	74351253	30714671	43064642
S_6	= 60311214	67246005	27316212	12561732
S_7	= 17375411	10604724	13221602	12742761
S_8	= 60501214	01406156	31461116	43664522
S_9	= 17176251	30731453	04614442	22665460
S_{10}	= 60703205	55051622	44471070	34065063
S_{11}	= 16765614	61541561	06010527	52104350
S_{12}	= 61122233	10216717	44131274	67045655

PURPOSE: To compute the double precision square root of a double precision argument X.

COMPASS CALLING SEQUENCE:

```
L        BRTJ     ($)DSQRT, , *
L+1      SLJ      *+2
         01       DICT.
L+2      00       ($)X
         NOP      0
```

FORTRAN FUNCTION: DSQRT(X)

NORMAL RETURN: The result is left in the A and Q registers.

ERROR MESSAGE: ERROR DETECTED IN DSQRT

A = (value)   Q = (value)

NEG ARG.

When the argument is negative this message is written on standard output and a normal exit is taken from DSQRT.

STORAGE: 37 locations   TIMING: 212 microseconds

ACCURACY: Average 83 bits; worst known relative error is 2×2^{84}

MATHEMATICAL METHOD: $X = 2^{2N}\,\overline{X}$, $\frac{1}{2} \leqq \overline{X} < 2$

$\sqrt{X} = 2^N * \sqrt{\overline{X}}$

$\sqrt{\overline{X}}$ is computed using the Newtonian approximation.

The initial value is a Chebyshev approximation due to R. F. King[1]:

$$Y_o = \frac{\overline{X}}{2} + 0.464841464$$

Two single precision iterations follow:

$$Y_i = 1/2 \left(Y_{i-1} + \frac{\overline{X}}{Y_{i-1}} \right) \qquad i = 1,2$$

Two double precision iterations follow, using the same expression with i = 3,4, and giving

$$Y_4 \approx \sqrt{\overline{X}}$$

NOTE: Refer to CO-OP routine B4 ANL ANL B451 DSQRT.

---

[1] Library Subroutine ANL B451 DSQRT Argonne Natl. Laboratory Argonne, Illinois

EXPF

PURPOSE: To compute the floating point exponential of a floating point argument X.

COMPASS CALLING SEQUENCE:

Call by Value (EXPF)

```
        LDA     X
L       BRTJ    ($)EXPF, , *
L+1     SLJ     * + 1
        00      DICT.
```

Call by Name (Q8QEXPF)

```
L       BRTJ    ($)Q8QEXPF, ,*
L+1     SLJ     *+2
        01      DICT.
L+2     00      ($)X
        NOP
```

FORTRAN FUNCTION: EXPF(X)

NORMAL RETURN: The floating point result is left in the A register.

ERROR MESSAGE: ERROR DETECTED IN EXPF

A = (value)   Q = (value)

ARG GT 709.

If the argument is greater than 709.089 this message is written on the standard output unit and a normal exit is taken from EXPF.

STORAGE: 42 locations

ACCURACY: 34 bits, accuracy decreases as arguments get large

TIMING: 140 microseconds

MATHEMATICAL METHOD:

$$X = M * Ln2 + R * Ln2$$

M = nearest integer to X/Ln2
$|R| \le 1/2$

$$e^X = 2^M * e^{R\,Ln2}$$

Let $u = R\,Ln2$; $e^u = \frac{P+u}{P-u} = \frac{P/Ln2+R}{P/Ln2-R}$

$$P = A + u^2\left(B + \frac{C}{D+u^2}\right)$$

$$P/Ln2 = A/Ln2 + R^2 * \left(B * Ln2 + \frac{C/Ln2}{D/Ln2^2 + R^2}\right)$$

In octal:

A/Ln2 = 2002561250731226

B*Ln2 = 1773433546344025

C/Ln2 = 2003704560677012

D/(Ln2*Ln2) = 2007535620560567

FLOATF

PURPOSE: To convert a fixed point number I to floating point.

COMPASS CALLING SEQUENCE:

Call by Value (FLOATF)

```
            LDA         I
L           BRTJ        ($)FLOATF, , *
L+1         SLJ         *+1
            00          DICT.
```

Call by Name (Q8QFLOAT)

```
L           BRTJ        ($)Q8QFLOAT, , *
L+1         SLJ         *+2
            01          DICT.
L+2         00          ($)I
            NOP         0
```

FORTRAN FUNCTION: FLOATF(I)

NORMAL RETURN: The floating point result is left in the A register.

ERROR MESSAGE: None

STORAGE: 12 locations

FLOATF is generally compiled in-line.

Compiled coding sequence:

```
                LDA         X
                ENO         * (if necessary)
                AJP, ZR     *+3
+               LRS         48
                SCM         =O2124000000000000
                DFAD        =DO0
```

PURPOSE: To convert a double precision argument X to fixed point.

COMPASS CALLING SEQUENCE:

L	BRTJ	($)IDINT, , *
L+1	SLJ	*+2
	01	DICT.
L+3	00	($)X
	NOP	0

FORTRAN FUNCTION: IDINT (X)

NORMAL RETURN: The integer result is left in the A register.

ERROR MESSAGE: ERROR DETECTED IN IDINT

A = (value)   Q = (value)

INTEGER TOO BIG

This message is written on the standard output unit and a normal exit is taken from IDINT when the argument is too large to convert to fixed point.

STORAGE: 24 locations

ACCURACY: 35-36 bits

INTF

PURPOSE: To truncate a floating point argument to the nearest floating point integer where magnitude is less than or equal to magnitude of the argument.

COMPASS CALLING SEQUENCE:

Call by Value (INTF)

```
            LDA     X
L           BRTJ    ($)INTF, , *
L+1         SLJ     *+1
            00      DICT.
```

Call by Name (Q8QINTF)

```
L           BRTJ    ($)Q8QINTF, ,*
L+1         SLJ     *+2
            01      DICT.
L+2         00      ($)X
            00
```

FORTRAN FUNCTION: INTF(X)

NORMAL RETURN: Floating point integer result is left in the A register.

ERROR MESSAGE: None

STORAGE: 8 locations

INTF is generally compiled in-line.

Compiled coding sequence:

```
            LDA     X
            ENO     * (if necessary)
            FAD,UR  =O2044000000000000
```

PURPOSE: To raise an integer number to an integer power: I**J.

COMPASS CALLING SEQUENCE:

Call by Value (ITOJ)

```
        LDA     I
        LDQ     J
L       BRTJ    ($)ITOJ, , *
L+1     SLJ     *+1
        00      DICT.
```

Call by Name(Q8QITOJ)

```
L       BRTJ    ($)Q8QITOJ, , *
L+1     SLJ     *+2
        02      DICT.
L+2     00      ($)I
        00      ($)J
```

FORTRAN FUNCTION: ITOJ(I,J)

NORMAL RETURN: The integer result is left in the A register.

ERROR MESSAGE: ERROR DETECTED IN ITOJ

A = (value)   Q = (value)

Error Code

Error Codes

I = 0, J = 0/NEG.	I is zero and J is zero or negative
J GT 47.	J is greater than 47 and $\|I\| > 1$
OVERFLOWED INTG.	$\|I^J\|$ is greater than $2^{47}-1$

If one of these error conditions occurs, the associated error message is printed and a normal exit taken.

ACCURACY: Exact

STORAGE: 66 locations including Q2Q07000

MATHEMATICAL METHOD:

Since $J = P*2^0 + Q*2^1 + R*2^2 + S*2^3 + T*2^4 + U*2^5$

(P, Q, R, S, T, U, = 1 or 0)

Then $I^J = I^{P*2^0} * I^{Q*2^1} * I^{R*2^2} * I^{S*2^3} * I^{T*2^4} * I^{U*2^5}$

Compute $I^2, I^4, I^8, I^{16}, I^{32}$ and multiply the powers of I that are present as indicated by the bit pattern of J.

ITOX

PURPOSE: To raise an integer to a floating point power: I**X.

COMPASS CALLING SEQUENCE:

Call by value (ITOX)

```
            LDA         I
            LDQ         X
L           BRTJ        ($)ITOX, , *
L+1         SLJ         *+1
            00          DICT.
```

Call by name (Q8QITOX)

```
L           BRTJ        ($)Q8QITOX, , *
L+1         SLJ         *+2
            02          DICT.
L+2         00          ($)I
            00          ($)X
```

FORTRAN FUNCTION: ITOX(I,X)

NORMAL RETURN: Floating point result is left in the A register.

ERROR MESSAGE: See POWRF for error messages.

STORAGE: 24 locations including Q2Q07101

ACCURACY: 34 bits, relative error increases for large arguments

MATHEMATICAL METHOD: Convert I to floating point and call POWRF.

PURPOSE: To compute the natural logarithm of a floating point argument X.

COMPASS CALLING SEQUENCE:

Call by Value (LOGF)

```
        LDA     X
L       BRTJ    ($)LOGF, , *
L+1     SLJ     *+1
        00      DICT.
```

Call by Name (Q8QLOGF)

```
L       BRTJ    ($)Q8QLOGF, , *
L+1     SLJ     *+2
        01      DICT.
L+2     00      ($)X
        NOP
```

FORTRAN FUNCTION: LOGF (X)

NORMAL RETURN: The floating point result is left in the A register.

ERROR MESSAGE: ERROR DETECTED IN LOGF

A = (value)   Q = (value)

ARG=0/NEG.

If the argument is zero or negative, this message is written on the standard output unit and a normal exit is taken from LOGF.

STORAGE: 60 locations

ACCURACY: Exact 75% of the time. Worst known error is 3 in last octal digit.

TIMING: 154 microseconds

MATHEMATICAL METHOD:[1]

$X = 2^n f$ where $2^{**}(-3/8) < f \leq 2^{**}(5/8)$

$LnX = nLn2 + Lnf$

$$Lnf = t\ (A*Ln2 + \frac{B*Ln2}{C + t^2}) - \alpha * Ln2$$

Divide the range into four intervals:

where $2^{(2i-5)/8} < f \leq 2^{(2i-3)/8}$ (for i = 1, 2, 3, 4),

set $\beta = 2^{-i/4}$ and $\alpha = - Ln\beta / Ln2$

Let $t = (f - \beta)/(f + \beta)$.

Then $Lnf = t\ (ALn2 + \frac{BLn2}{C + t^2}) - \alpha Ln2$

ALn2 = 0.88924967148818

BLn2 = -1.85064969453155

C = -1.66612572331613

---

[1] Reference: COOP routine B3 ANL ANLB350 LOGF

PURPOSE: To determine the maximum value of the fixed point arguments $I_1, I_2, \ldots, I_n$.

COMPASS CALLING SEQUENCE:

L	BRTJ	($)MAX0F, , *
L+1	SLJ	*+m
	n	DICT.
L+2	00	($)$I_1$
	00	($)$I_2$
	.	
	.	
	.	
	00	($)$I_n$

m = (n+1)/2 + 1, n = number of parameters

FORTRAN FUNCTION: MAX0F(i_1, i_2, . . .)

NORMAL RETURN: The maximum fixed point argument, converted to floating point is left in the A register.

ERROR MESSAGE: None

STORAGE: 81 locations for all MIN and MAX functions

MAX1F

PURPOSE: To determine the maximum value of the floating point arguments $X_1, X_2, \ldots, X_n$.

COMPASS CALLING SEQUENCE:

```
L        BRTJ     ($)MAX1F, , *
L+1      SLJ      *+m
         n        DICT.
L+2      00       ($)X1
         00       ($)X2
         .
         .
         .
         00       ($)Xn
```

$m = (n + 1)/2 + 1$, n = number of parameters

FORTRAN FUNCTION: MAX1F $(X_1, X_2, \ldots)$

NORMAL RETURN: The largest floating point argument is left in the A register.

ERROR MESSAGF: None

STORAGE: 81 locations for all MIN and MAX functions

PURPOSE: To determine the minimum value of the fixed point arguments $I_1, I_2, \ldots, I_n$, and return the result in floating point.

COMPASS CALLING SEQUENCE:

L	BRTJ	($)MIN0F, , *
L+1	SLJ	*+m
	n	DICT.
L+2	00	($) I_1
	00	($) I_2
	.	
	.	
	.	
	00	($) I_n

m = (n+1)/2 + 1, n = number of parameters

FORTRAN FUNCTION: MIN0F ($I_1, I_2, \ldots, I_n$)

NORMAL RETURN: Floating point value of the minimum argument is left in the A register.

ERROR MESSAGE: None

STORAGE: 81 locations for MIN and MAX functions

MIN1F

PURPOSE: To determine the minimum value of the floating point arguments $X_1, X_2, \ldots, X_n$.

COMPASS CALLING SEQUENCE:

```
L       BRTJ    ($)MIN1F, , *
L+1     SLJ     *+m
        n       DICT.
L+2     00      ($)X1
        00      ($)X2
        .
        .
        .
        00      ($)Xn
```

$m = (n + 1)/2 + 1$, n = number of parameters

FORTRAN FUNCTION: MIN1F($X_1, X_2, \ldots, X_n$)

NORMAL RETURN: The minimum floating point argument is left in the A register.

ERROR MESSAGE: None

STORAGE: 81 locations for MIN and MAX functions

PURPOSE: To compute the value of the floating point argument X_1 modulo X_2.

COMPASS CALLING SEQUENCE:

Call by Value (MODF)

```
            LDA     X1
            LDQ     X2
L           BRTJ    ($)MODF, , *
L+1         SLJ     *+1
            00      DICT.
```

Call by Name (Q8QMODF)

```
L           BRTJ    Q8QMODF,,*
L+1         SLJ     *+2
            02      DICT.
L+2         00      ($)X1
            00      ($)X2
```

FORTRAN FUNCTION: MODF(X_1, X_2)

NORMAL RETURN: The value of X_1 modulo X_2 is left in the A register. If X_1 is zero or the integer portion of the quotient X_1/X_2 exceeds 2^{37} -1, zero is left in the A register. The result is negative only if X_1 and X_2 have unlike signs.

ERROR MESSAGE: ERROR DETECTED IN MODF

A = (value)   Q = (value)

Q = ZERO.

This message is written on the standard output unit if X_2=0 and a normal exit is taken.

STORAGE: 38 locations

ACCURACY: Depends on the size of the arguments and the subtraction, $X_1 - [X_1/X_2] * X_2$.

MATHEMATICAL METHOD: The result is computed as $X_1 - [X_1/X_2] * X_2$, where only the integer part of the bracketed quantity is used.

POWRF

PURPOSE: To perform general exponentiation for A^B where A and B are floating point arguments.

COMPASS CALLING SEQUENCE:

Call by Value (POWRF)

```
            LDA         A
            LDQ         B
L           BRTJ        ($)POWRF, , *
L+1         SLJ         *+1
            00          DICT.
```

Call by Name (Q8QPOWRF)

```
L           BRTJ        ($)Q8QPOWRF, , *
L+1         SLJ         *+2
            02          DICT.
L+2         00          ($)A
            00          ($)B
```

FORTRAN FUNCTION: POWRF (A,B)

NORMAL RETURN: Floating point result is left in the A register.

ERROR MESSAGE: ERROR DETECTED IN POWRF

A = (value)   Q = (value)

Error Code

Error Codes

BASE LT ZERO.	Base is negative
B*LN(A) GT 709.	Maximum is exceeded
EXP=0/NEG.	Power is zero or negative and base is zero

This message is written on the standard output unit and a normal exit is taken from POWRF.

STORAGE: 115 locations

ACCURACY: 34 bits, as arguments get large relative error increases

TIMING: 292 microseconds

METHOD: A^B = Exponential (B*Ln(A)) which is computed within the routine.

PURPOSE: To perform mixed mode conversion for complex mode load and arithmetic operations.

COMPASS CALLING SEQUENCE:

Two word operands

```
L        BRTJ      ($)Q1Qop3s0,,*
L+1      DLDA      ($)V,b
L+2      Return
```

One word operands

```
L        BRTJ      ($)Q1Qop3s0,,*
L+1      ENO       $V
         LDA       V,b
L+2      Return
```

Partial word operands

```
L        BRTJ      ($)Q3Qop340,,*
L+1      01        ca,b
         pof       ($)V
L+2      Return
```

V	Variable location
b	index register
ca	constant addend
pof	parameter offset

s	mode of v
0	integer
1	real
2	double
3	complex

op	Operation
00	load
01	load complement
02	add
03	subtract
04	multiply
05	divide

Routines included in this package

Q1Q00300	Load, integer to complex
Q1Q01300	Load complement, integer to complex
Q1Q02300	Add, integer to complex
Q1Q03300	Subtract, integer from complex
Q1Q04300	Multiply, complex by integer
Q1Q05300	Divide, complex by integer
Q1Q00310	Load, real to complex
Q1Q01310	Load complement, real to complex

Routines included in this package (continued)

Q1Q02310	Add, real to complex
Q1Q03310	Subtract, real from complex
Q1Q04310	Multiply, complex by real
Q1Q05310	Divide, complex by real
Q1Q00320	Load, double to complex
Q1Q01320	Load complement, double to complex
Q1Q02320	Add, double to complex
Q1Q03320	Subtract, double from complex
Q1Q04320	Multiply, complex by double
Q1Q05320	Divide, complex by double
Q0Q06300	Complement complex AQ register
Q1Q02330	Add, complex to complex
Q1Q03330	Subtract, complex from complex
Q1Q04330	Multiply complex by complex
Q1Q05330	Divide, complex by complex
Q3Q00340	Load, logical to complex
Q3Q01340	Load complement, logical to complex
Q3Q02340	Add, logical to complex
Q3Q03340	Subtract, logical from complex
Q3Q04340	Multiply, complex by logical
Q3Q05340	Divide, complex by logical

NORMAL RETURN: Complex result in AQ register

ERROR MESSAGE: None

STORAGE: 193 locations

Q1Q00310, Q1Q01310, and Q0Q06300
are generally compiled in-line.

PURPOSE: To perform mixed mode conversion for double precision load and arithmetic operations.

COMPASS CALLING SEQUENCE:

Full word

L	BRTJ	($)Q1Qop2s0, ,*
L+1	ENO	$V
	LDA	V,b
L+2	Return	

Partial word

L	BRTJ	($)Q3Qop240, ,*
L+1	01	ca,b
	pof	($)V
L+2	Return	

V	Variable location
b	index register
ca	constant addend
pof	parameter offset

s	mode of V
0	integer
1	real

op	Operation
00	load
01	load complement
02	add
03	subtract
04	multiply
05	divide

Routines included in this package

Q1Q00200	Load, integer to double
Q1Q01200	Load complement, integer to double
Q1Q02200	Add, integer to double
Q1Q03200	Subtract, integer from double
Q1Q04200	Multiply, double by integer
Q1Q05200	Divide, double by integer
Q1Q00210	Load, real to double
Q1Q01210	Load complement, real to double

Routine included in this package (continued)

Q1Q02210	Add, real to double
Q1Q03210	Subtract, real from double
Q1Q04210	Multiply, double by real
Q1Q05210	Divide, double by real
Q3Q00240	Load, logical to double
Q3Q01240	Load complement, logical to double
Q3Q02240	Add, logical to double
Q3Q03240	Subtract, logical from double
Q3Q04240	Multiply, double by logical
Q3Q05240	Divide, double by logical
Q0Q06200	Complement double precision AQ register

NORMAL RETURN: The double precision result is left in the AQ register.

ERROR MESSAGE: None

STORAGE: 120 locations

Q1Q00210, Q1Q01210, and Q0Q06200
are generally compiled in-line.

PURPOSE: To perform conversion from integer to real and from logical to real or integer for load and arithmetic operations.

COMPASS CALLING SEQUENCES:

Integer to real conversion

L	BRTJ	($)Q1Qop100, ,*
L+1	ENO	$V
	LDA	V,b
L+2	Return	

Logical to real or integer

L	BRTJ	($)Q3Qopm40, ,*
L+1	01	ca,b
	pof	($)V
L+2	Return	

V variable location

b index register

pof parameter offset

ca constant addend

op	Operation
00	load
01	load complement
02	add
03	subtract
04	multiply
05	divide

m	mode
0	integer
1	real

Routines included in this package

Q1Q00100	Load, integer to real
Q1Q01100	Load complement, integer to real
Q1Q02100	Add, integer to real
Q1Q03100	Subtract, integer from real
Q1Q04100	Multiply, integer by real
Q1Q05100	Divide, real by integer
Q3Q00140	Load, logical to real
Q3Q01140	Load complement, logical to real
Q3Q02140	Add, logical to real

Routines included in this package (continued)

Q3Q03140	Subtract, logical from real
Q3Q04140	Multiply, real by logical
Q3Q05140	Divide, real by logical
Q3Q00040	Load, logical to integer
Q3Q01040	Load complement, logical to integer
Q3Q02040	Add, logical to integer
Q3Q03040	Subtract, logical from integer
Q3Q04040	Multiply, integer by logical
Q3Q05040	Divide, integer by logical

NORMAL RETURN: The result is left in the A register.

ERROR MESSAGE: None

STORAGE: 112 locations

Q1Q00100 and Q1Q01100
are generally compiled in-line.

PURPOSE: To perform and store mixed mode conversion.

COMPASS CALLING SEQUENCE:

Store full word

L	BRTJ	($)Q1Q10ms0, ,*
L+1	ENO	$V
	STA	V,b
L+2	Return	

Store two full words

L	BRTJ	($)Q1Q10ms0, ,*
L+1	DSTA	($)V,b
L+2	Return	

Store partial word

L	BRTJ	($)Q3Q10m40, ,*
L+1	01	ca,b
	pof	($)V
L+2	Return	
V	Storage location of result	
b	index register	
ca	constant addend	
pof	parameter offset	

m,s	mode
0	integer
1	real
2	double
3	complex
4	logical

Routines included in this package

Q1Q10100	Store, real to integer
Q1Q10200	Store, double to integer
Q1Q10300	Store, complex to integer
Q1Q10400	Store, logical to integer
Q1Q10010	Store, integer to real
Q1Q10210	Store, double to real
Q1Q10310	Store, complex to real
Q1Q10410	Store, logical to real
Q1Q10020	Store, integer to double

Routines included in this package (continued)

Q1Q10120	Store, real to double
Q1Q10320	Store, complex to double
Q1Q10420	Store, logical to double
Q1Q10030	Store, integer to complex
Q1Q10130	Store, real to complex
Q1Q10230	Store, double to complex
Q1Q10430	Store, logical to complex
Q3Q10040	Store, integer to logical
Q3Q10140	Store, real to logical
Q3Q10240	Store, double to logical
Q3Q10340	Store, complex to logical
Q3Q10440	Store, logical to logical

NORMAL RETURN: The result is left in the A register.

ERROR MESSAGE: None

STORAGE: 119 locations

Q1Q10010, Q1Q10310, Q1Q10120, and Q1Q10130 are generally compiled in-line.

PURPOSE: To evaluate I**J.

COMPASS CODING SEQUENCE:

```
L        BRTJ    ($)Q2Q07000,,*
L+1      SLJ     *+3
         02      DICT.
L+2      LDA     ($)I
L+3      LDA     ($)J
```

FORTRAN FUNCTION: This routine is called whenever I**J is encountered in a FORTRAN statement, and I and J are both type integer.

NORMAL RETURN: The integer result is left in the A register.

ERROR MESSAGE: See ITOJ for error messages.

ACCURACY: Exact

STORAGE: 66 locations including ITOJ.

MATHEMATICAL METHOD: See ITOJ.

PURPOSE: To evaluate I**X

COMPASS CODING SEQUENCE:

```
L       BRTJ    ($)Q2Q07101,,*
L+1     SLJ     *+3
        02      DICT.
L+2     LDA     ($)I
L+3     LDA     ($)X
```

FORTRAN FUNCTION: This routine is called whenever I**X is encountered in a FORTRAN statement, and I is type integer and X is type real.

NORMAL RETURN: Floating point result is left in the A register.

ERROR MESSAGE: See POWRF for error messages.

STORAGE: 24 locations including ITOX.

ACCURACY: 34 bits, relative error increases for large amounts.

MATHEMATICAL METHOD: Convert I to floating point and call POWRF.

PURPOSE: To evaluate X**I

COMPASS CODING SEQUENCE:

L	BRTJ	($)Q2Q07110,,*
L+1	SLJ	*+3
	02	DICT.
L+2	LDA	($)X
L+3	LDA	($)I

FORTRAN FUNCTION: This routine is called whenever X**I is encountered in a FORTRAN statement, and X is type real and I is type integer.

NORMAL RETURN: Floating point result is left in the A register.

ERROR MESSAGE: See XTOI for error messages.

STORAGE: 87 locations including XTOI.

ACCURACY: 36 bits for $|I| \leq 1023$, 34 bits for $|I| > 1023$

MATHEMATICAL METHOD: See XTOI.

PURPOSE: To compute I**D where I is type integer and D is type double.

COMPASS CALLING SEQUENCE:

L	BRTJ	($)Q2Q07202, ,*
L+1	SLJ	*+3
	02	DICT.
L+2	LDA	($)I
L+3	DLDA	($)D

FORTRAN FUNCTION: This routine is called whenever I**D is encountered in a FORTRAN statement.

NORMAL RETURN: The double precision result is left in the A and Q registers.

ERROR MESSAGES: Since Q2Q07212 is called, error messages may appear from DLOG and/or DEXP.

STORAGE: 17 locations

ACCURACY: 82 bits average

MATHEMATICAL METHOD: I is converted to floating point and Q2Q07212 (R**D) is called.

PURPOSE: To compute D**D, D**R, or R**D where R is type real and D is type double.

COMPASS CALLING SEQUENCE:

U = location of base V = location of power

For D**D

```
L       BRTJ    ($)Q2Q07222, ,*
L+1     SLJ     *+3
        02      DICT.
L+2     DLDA    ($)U
L+3     DLDA    ($)V
```

For D**R

```
L       BRTJ    ($)Q2Q07221, ,*
L+1     SLJ     *+3
        02      DICT.
L+2     DLDA    ($)U
L+3     LDA     ($)V
```

For R**D

```
L       BRTJ    ($)Q2Q07212, ,*
L+1     SLJ     *+3
        02      DICT.
L+2     LDA     ($)U
L+3     DLDA    ($)V
```

FORTRAN FUNCTION: The appropriate Q2Q07...routine is called when a U**V is encountered in a FORTRAN statement where U and/or V are type double.

NORMAL RETURN: The double precision result is left in the AQ register.

STORAGE: 88 (for DEXP) + 124 (for DLOG) + 47 (for DPOWER) = 259 locations

ACCURACY: 82 bits average

MATHEMATICAL METHOD: U^V is computed as exp(V*LnU), DLOG is called to compute LnU and DEXP is called to compute exp(V*LnU).

PURPOSE: To compute D**I where D is type double and I is type integer.

COMPASS CALLING SEQUENCE:

```
L       BRTJ    ($)Q2Q07220
L+1     SLJ     *+3
        02      DICT.
L+2     DLDA    ($)D
L+3     LDA     ($)I
```

FORTRAN FUNCTION: This routine is called whenever D**I is encountered in a FORTRAN statement.

NORMAL RETURN: The double precision result is left in the A and Q registers.

ERROR MESSAGE: ERROR DETECTED IN Q2Q07220

A = (value) Q = (value)

D = 0, EXP = 0/NEG.

This message is written on standard output when the base is zero and the power is less than or equal to zero. The most significant part of the base is in the A register and the power is in the Q register.

Since Q2Q07220 calls Q2Q07221 for powers greater than 1023, see Q2Q07221 for other error messages.

STORAGE: 94 locations

ACCURACY: 83 bits for $|I| \leq 1023$, 82 bits average for $|I| > 1023$.

MATHEMATICAL METHOD: The method is the same as that used for XTOI (X^I) except the computations are in double precision. See METHOD heading of XTOI.

For powers greater than 1023, I is converted to floating point and Q2Q07221 is called.

PURPOSE: To compute I**Z where I is type integer and Z is type complex; the real part is in Z; the imaginary part in Z + 1.

COMPASS CALLING SEQUENCE:

```
L       BRTJ    ($)Q2Q07303, ,*
L+1     SLJ     *+3
        02      DICT.
L+2     LDA     ($)I
L+3     DLDA    ($)Z
```

FORTRAN FUNCTION: This routine is called whenever I**Z is encountered in a FORTRAN statement and I is type integer and Z is type complex.

NORMAL RETURN: The real part is left in the A register and the imaginary part in the Q register.

ERROR MESSAGES: See LOGF, POWRF, SINF, and COSF.

STORAGE: 40 locations including Q2Q07313

ACCURACY: Depends on POWRF, LOGF, SINF, and COSF and the arguments used.

MATHEMATICAL METHOD: I is converted to floating point and Q2Q07313 (R** Z) is used. (Q2Q07303 and Q2Q07313 are entries in the same routine.)

Q2Q07313

PURPOSE: To compute R**Z when R is type real and Z is type complex; the real part is in Z; the imaginary part in Z + 1.

COMPASS CALLING SEQUENCE:

L	BRTJ	($)Q2Q07313, ,*
L+1	SLJ	*+3
	02	DICT.
L+2	LDA	($)R
L+3	DLDA	($)Z

FORTRAN FUNCTION: This routine is called whenever R**Z is encountered in FORTRAN statement and R is type real and Z is type complex.

NORMAL RETURN: The real part is left in the A register and the imaginary part in the Q register.

ERROR MESSAGES: See LOGF, POWRF, SINF, and COSF.

STORAGE: 40 locations including Q2Q07303

ACCURACY: Depends on POWRF, LOGF, SINF, and COSF and the arguments used.

MATHEMATICAL METHOD: For the complex number Z

$Z = X + iY$ X, Y, R are real

$R{**}Z = R^X * \cos(Y*\mathrm{Ln}(R)) + i*R^X*\sin(Y*\mathrm{Ln}(R))$

POWRF, COSF, LOGF, SINF are called to compute R^X, cos(Y*Ln(R)), Ln(R), and sin (Y*Ln(R)) respectively.

PURPOSE: To compute D**Z where D is type double and Z is type complex.

COMPASS CALLING SEQUENCE:

L	BRTJ	($)Q2Q07323, ,*
L+1	SLJ	*+3
	02	DICT.
L+2	DLDA	($)D
L+3	DLDA	($)Z

FORTRAN FUNCTION: This routine is called whenever D**Z is encountered in a FORTRAN statement and D is type double and Z is type complex.

NORMAL RETURN: The real part is left in the A register and the imaginary part in the Q register.

ERROR MESSAGES: See DLOG, EXPF, SINF, and COSF.

STORAGE: 46 locations

ACCURACY: Depends on EXPF, SINF, and COSF and the arguments used.

MATHEMATICAL METHOD: For the complex number Z

$Z = X + iY$ X, Y are real

$D^{**}Z = D^{X} * \cos(Y * \mathrm{Ln}(D)) + i * D^{X} * \sin(Y * \mathrm{Ln}(D))$

where $D^{X} = \mathrm{EXP}(X * \mathrm{Ln}(D))$

DLOG, EXPF, SINF, and COSF are called to compute Ln(D), EXP(X*Ln(D)), SIN(Y*Ln(D)), and COS(Y*Ln(D)) respectively.

Q2Q07330

PURPOSE: To compute Z**I where Z is complex and I is integer. The real part is in Z; the imaginary part in Z + 1.

COMPASS CALLING SEQUENCE:

L	BRTJ	($)Q2Q07330, ,*
L+1	SLJ	*+3
	02	DICT.
L+2	DLDA	($)Z
L+3	LDA	($)I

FORTRAN FUNCTION: This routine is called whenever Z**I is encountered in a FORTRAN statement and Z is type complex and I is type integer.

NORMAL RETURN: The real part is left in the A register and the imaginary part in the Q register.

ERROR MESSAGES: See XTOI, COSF, and SINF.

STORAGE: 35 locations

ACCURACY: Depends on XTOI, COSF, SINF, and CABS and the arguments used.

MATHEMATICAL METHOD: For the complex number Z

$Z = X+iY$ X and Y are real

$Z^{**}N = |Z|^N (COS(N^*\theta)) + i\,SIN(N^*\theta))$

$|Z| = (X^2 + Y^2)^{1/2}$

$\theta = Tan^{-1} (Y/X)$ such that $-\pi < \theta \leqq \pi$

XTOI, COSF, SINF, CABS, CANG are called to perform

$|Z|^N$, $COS(N^*\theta)$, $SIN(N^*\theta)$, $|Z|$, and to get θ, respectively.

If N is even, compute $|Z|^N = (X^2 + Y^2)^{N/2}$

Routines Z**R, Z**D, and Z**Z have not been implemented for the complex number Z, the real number R, the integer number I, and the double precision number D, because, according to strict mathematical definition, these function are either multivalued or indeterminate depending upon whether the exponent is rational or irrational. Although computer representation can always be expressed as the rational fraction M/N, there are still N legitimate values for the answer.

To define a unique principal value could lead to frequent discontinuities, and (Z**R)**(I/R) would in general be drastically different from (Z**(I/R))**R.

Therefore, if one of these routines is called, the following message is written on standard output and a normal return is taken:

ERROR MESSAGE: ERROR DETECTED IN C**S

A = (value) Q = (value)

ILLEGAL POWER.

The values of the A and Q registers are meaningless.

STORAGE: 11 locations

Q2QDLDA

PURPOSE: To get the arguments for a routine which computes A**B where A and/or B is a two word variable (complex or double).

COMPASS CALLING SEQUENCE:

The Q2Q type calling sequence is used for all exponentiation, no matter what the types of the arguments.

```
L         BRTJ      ($)Q2Q07mst, ,*
L+1       SLJ       *+3
          02        DICT.
L+2       XXX       ($)BASE,b
L+3       YYY       ($)POWER,b
```

XXX = LDA or DLDA,   YYY = LDA or DLDA

	m,s,t	type
m = type of result	0	integer
s = type of base	1	real
	2	double
t = type of power	3	complex

Then Q2Q07mst may appear:

```
Q2Q07mst    UBJP    (*)**
L+1         XMIT    (*)*-1, ($)Q8QDICT.
L+2         XMIT    (*)*-2, (*)DICT.
L+3         BRTJ    ($)Q2QDLDA, ,*
L+4         STA     GETBASE
            STQ     GETPOWER
GETBASE     OCT     0
              .
              .
              .
GETPOWER    OCT     0
```

NORMAL RETURN: (XXX ($) BASE,b) is left in the A register and (YYY ($) POWER,b) is in the Q register. These instructions can then be stored and executed to get the actual parameters when needed.

STORAGE: 7 locations

USAGE: Q2QLDA is used in all routines processing A**B where A and/or B are type complex and/or double.

PURPOSE: To get the arguments A and B to the accumulator for a routine which computes A**B, where A and B are single word variables (integer or real).

COMPASS CALLING SEQUENCE:

The Q2Q type calling sequence is used for R**R, R**I, I**R, I**I. I is type integer. R is type real.

```
L       BRTJ    ($)Q2Q07mst, ,*
L+1     SLJ     *+3
        02      DICT.
L+2     LDA     ($)BASE,b
L+3     LDA     ($)POWER,b
```

		m,s,t	type
where	m = type of result	0	integer
	s = type of base	1	real
		2	double
	t = type of power	3	complex

Then Q2Q07mst may appear:

```
Q2Q07mst    UBJP     (*)**
A+1         XMIT     (*)*-1, ($)Q8QDICT.
A+2         XMIT     (*)*-2, (*)DICT.
A+3         BRTJ     (*)Q2QLOADA, ,*
A+4         Return from Q2QLOADA
```

NORMAL RETURN: Upon return from Q2QLOADA, the base will be in the A register and the power will be in the Q register. Q2QLOADA executes the actual instructions L + 2, L + 3 in the calling sequence to Q2Q07mst to get the base and the power.

USAGE: Q2QLOADA is used in Q2Q07000 (I**I), Q2Q07111 (R**R), Q2Q07101 (I**R), and Q2Q07110 (R**I).

STORAGE: 10 locations

Q7QLODLC

PURPOSE: To load a logical variable and convert it to floating point.

COMPASS CALLING SEQUENCE:

L	BRTJ	($)Q7QLODLC
L+1	Return	

FORTRAN FUNCTION: This routine is called by a Q3Q. . .routine in a Q1Q. . .package to load the logical value specified in a logical mixed mode expression.

NORMAL RETURN: The floating point result, 0.0 or 1.0, is left in the A register, and the Q register is cleared.

ERROR MESSAGE: none

STORAGE: 26 locations

PURPOSE: To pick up arguments from successive locations in a Q8Q type of calling sequence.

COMPASS CALLING SEQUENCE:

A subroutine with a call by name entry and a two word parameter.(Double, Complex or some other 2-word type), with error tracing; may be written as:

NAME	UBJP	(*)**
L+1	XMIT	(*)*-1, ($)Q8QDICT.
L+2	XMIT	(*)*-2, (*)DICT.
L+3	BRTJ	($)Q8QDLDA, ,*
L+4	Normal return from Q8QDLDA	

where Q8QDICT. is an external symbol.

The same subroutine with no error tracing may be written as:

NAME	UBJP	(*)**
L+1	XMIT	(*)*-1, ($)Q8QDCONS
L+2	BRTJ	($)Q8QDLODA, ,*
L+3	Normal return for Q8QDLODA	

when Q8QDCONS is an external symbol.

NAME has the Q8Q type calling sequence. Q8QDLDA uses Q8QDICT. to get the relative address of the argument. Q8QDLODA uses Q8QDCONS to get the relative address of the argument.

NORMAL RETURN: The double precision argument is left in the AQ register.

STORAGE: 14 locations

USAGE: Q8QDLDA is used to get the double precision argument for DLOG, DSQRT, DSIN/DCOS, DEXP. It is also used to get the complex argument for CATAN, CEXP, CSIN/CCOS, and CLOG.

Q8QDLODA is used to get the double precision argument for DCUBRT, and DATAN. It is also used to get the complex argument for CANG, CABS, and CSQRT..

Q8QENTRY

PURPOSE: 1. To initialize storage for the execution of FORTRAN compiled programs by the following steps:

Clears Q8QHIST. tables

Determines and saves available memory

Writes EXECUTION STARTED AT hhmmss on standard output unit

2. To terminate input/output operations by calling EXIT.

Calls IOP. to check and close all input/output tapes

Returns control to SCOPE

COMPASS CALLING SEQUENCE:

Call for initialization after loading a program for execution

L	BRTJ	($)Q8QENTRY,,*	
L+1	SLJ	*+1	
	00	DICT.	DICT. will be set to contain transfer to EXIT

Call by EXIT

L	BRTJ	($)EXIT,,*
L+1	SLJ	*+1
	00	DICT.

NORMAL RETURN: Subsequent calls to Q8QENTRY act as a NOP.

ERROR MESSAGE: None.

STORAGE: 51 locations for Tape SCOPE version.

46 locations for Drum SCOPE version.

PURPOSE: To print error diagnostics on the standard output unit. The error diagnostics include:

Error message

Contents of A and Q registers

Name of routine in which error occurred

A trace of the chain of subprogram calls back to the main program

COMPASS CALLING SEQUENCE:

L	BRTJ	($)Q8QERROR, , *	
L+1	SLJ	*+2	
	02	DICT.	
L+2	00	($)K	K Location of error key
	00	($)M	M Location of BCD error message

Alternate entry to preset fatal option

L	BRTJ	($)Q8QERSET, , *
L+1	SLJ	*+2
	01	DICT.
L+2	00	($)N
	00	0

N = 0, all entries to Q8QERROR are considered fatal
N ≠ 0, all entries to Q8QERROR are treated as specified by (K)

NORMAL RETURN: If (K) is non-zero, the error message is printed, A, Q registers restored, and the next statement is executed.

FORTRAN FUNCTION: CALL Q8QERSET (N); CALL Q8QERROR (K, M)

ERROR MESSAGE: (K) = 0, fatal error. The standard printout is followed by:

EXECUTION DELETED. Control returns to SCOPE. A zero field may replace ($)K.

(K) ≠ 0, non-fatal error. The standard printout is followed by a normal return.

ERROR RETURN: If (K) is zero, job is abandoned.

STORAGE: 133 locations

PURPOSE: To interrogate the Interrupt Register to determine if a divide fault, an arithmetic overflow, or an exponent overflow has occurred.

COMPASS CALLING SEQUENCE:

```
L       BRTJ     ($)name, , *
L+1     SLJ      *+1
        00       DICT.
L+2     AJP,MI   n1
        SLJ      n2
```

name	condition
Q8QIFDIV	divide { fault / check
Q8QIFOVF	arithmetic overflow
Q8QIFEXP	exponent overflow

FORTRAN FUNCTIONS: IF DIVIDE {FAULT / CHECK} n_1, n_2

IF OVERFLOW FAULT n_1, n_2

IF EXPONENT FAULT n_1, n_2

NORMAL RETURN: Control transfers to statement n_1 and the fault condition is cleared if the fault has occurred, or to statement n_2 if the fault has not occurred.

ERROR MESSAGE: None

STORAGE: 15 locations

PURPOSE: To interrogate a designated sense switch and branch to one of two given statements, depending on the status of the sense switch.

COMPASS CALLING SEQUENCE:

	ENA	i	if the argument is an integer constant, or
			LDA I if the argument is an integer variable
L	BRTJ	($)Q8QIFSSW, , *	
L+1	SLJ	*+1	
	00	DICT.	
L+2	AJP, MI	n_1	
	SLJ	n_2	

FORTRAN FUNCTION: IF (SENSE SWITCH i) n_1, n_2

NORMAL RETURN: If the corresponding sense switch is on, control transfers to statement n_1; if the corresponding sense switch is off, control transfers to statement n_2.

ERROR MESSAGE: SENSE SWITCH OUT OF RANGE

If the integer i is outside the range, $1 \leq i \leq 6$, this message is printed on the standard output unit and control transfers to n_2.

STORAGE: 20 locations

Q8QLOADA and Q8QLODA

PURPOSE: To pick up an argument.

COMPASS CALLING SEQUENCE:

A subroutine with a call by name entry (Q8Qname) and a call by value entry (name), with error tracing may be written as:

```
Q8Qname     UBJP     (*)**
L+1         XMIT     (*)*-1, ($)Q8QDICT.
L+2         XMIT     (*)*-2, ($)DICT.
L+3         BRTJ     ($)Q8QLOADA
name        UBJP     (*)**
L+5         XMIT     (*)*-1, ($)Q8QDICT.
L+6         XMIT     (*)*-2, ($)DICT.
L+7         Normal return from Q8QLOADA
```

where Q8QDICT. is an external symbol.

The same subroutine with no error tracing:

```
Q8Qname     UBJP     (*)**
L+1         XMIT     (*)*-1, ($)Q8QLDCON
L+2         BRTJ     ($)Q8QLODA
name        UBJP     (*)**
L+4         Normal return from Q8QLODA
```

Q8Qname has the Q8Q type calling sequence. Q8QLOADA uses Q8QDICT. to get the relative address of the argument. Q8QLODA uses Q8QLDCON to get the relative address of the argument.

Q8QLOADA returns to *+4 and Q8QLODA returns to *+2.

NORMAL RETURN: The argument is left in the A register.

STORAGE: 18 locations

PURPOSE: To stop operation, displaying an octal constant n on the output comment medium (OCM).

1. For PAUSE, a normal return is taken when a character is received from the input comment medium (ICM)
2. For STOPS, a transfer to EXIT in Q8QENTRY is taken when a character is received from the ICM. No output occurs on OCM when n is zero.

COMPASS CALLING SEQUENCE:

```
            ENA       n
L           BRTJ      ($)Q8QPAUSE,,* or ($)Q8QSTOPS,,*
L+1         SLJ       *+1
            00        DICT.
```

FORTRAN FUNCTION: PAUSE n
STOP n   n may be absent

NORMAL RETURN: For Q8QPAUSE, operation resumes with the next statement with n in the accumulator. For Q8QSTOPS operation resumes, transferring control back to EXIT, and the job is terminated.

ERROR MESSAGE: none

STORAGE: 32 locations for both Tape SCOPE and Drum SCOPE versions

Q8QRESID

PURPOSE: To initialize actual address and bank parameters in a called subroutine from a parameter substitution list.*

COMPASS CALLING SEQUENCE:

L	BRTJ	($)Q8QRESID, , *
L+1	SLJ	Return
	n	DICT.
L+2	r_1t_i	ca
	0m	P_1
L+3	r_2t_i	P_2
	.	.
	.	.
	.	.
	77	77777,7
	77	77777,7

n number of arguments

m argument ordinal number

r	reference
4	first reference to argument m
0	subsequent reference to same argument

ca constant addend to address a + P_i

77. . .7 end of parameter list

t	address
0	upper address of P_i
1	upper 18 bits of P_i
2	upper 24 bits of P_i
3	lower address of P_i, upper AUG bank field of P_i
4	lower address of P_i
5	lower 18 bits of P_i
6	lower 24 bits of P_i
7	upper address of P_i + 1, lower AUG bank field of P_i

NORMAL RETURN: The index registers and A,Q registers are restored.

ERROR MESSAGE: None

STORAGE: 55 locations

*Not advisable for handcoded routines.

PURPOSE: 1. To turn off all sense lights, or to turn on a designated sense light.

2. To interrogate a designated sense light and branch to one of two given statements, depending on the status of the sense light.

COMPASS CALLING SEQUENCE:

Call by SENSE LIGHT i

```
            ENA      i   if i is an integer constant, or LDA I if i is an integer variable
L           BRTJ     ($)Q8QSENLT, , *
L+1         SLJ      *+1
            00       DICT.
```

Call by IF(SENSE LIGHT i) n_1, n_2

```
            ENA      i  if i is an integer constant, or LDA I if i is an integer variable
L           BRTJ     ($)Q8QIFSLT, , *
L+1         SLJ      *+1
            00       DICT.
L+2         AJP,MI   n1
            SLJ      n2
```

FORTRAN FUNCTIONS: SENSE LIGHT I

IF (SENSE LIGHT I) n_1, n_2

NORMAL RETURN: For SENSE LIGHT I:

If I = 0, all sense lights are turned off; if $1 \leq I \leq 4$, the corresponding sense light is turned on.

For IF (SENSE LIGHT I) n_1, n_2:

If the corresponding sense light is on, it is turned off, and control transfers to statement n_1; if the corresponding sense light is off, control transfers to statement n_2.

ERROR MESSAGE: SENSE LIGHT OUT OF RANGE

If the integer i is outside the range $1 \leq i \leq 4$, this message is printed on the standard output unit and a normal return is taken.

STORAGE: 39 locations

Q9QEVAL

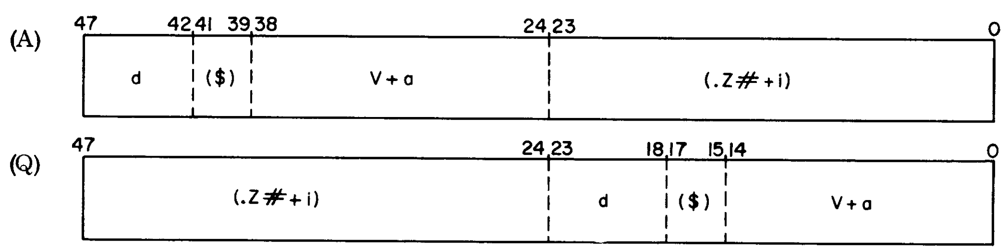
PURPOSE: To compute the offset and operand address of an element in a logical or byte array and use this information to position the element in a called subroutine parameter list.

COMPASS CALLING SEQUENCE:

L	BRTJ	name	(Return to L+2)
L+1	n	ca,b	
	f	($)V	
L+2	STA	.Z#+i	for upper half word or STQ .Z#+i for lower half word

name	Q9QEVALL for logical elements Q9QEVALB for byte elements
n	byte size (number of bits--01 for logical)
ca	constant addend
b	index register
f	bit offset
($)V	base address of array V

NORMAL RETURN: Upon return, the following information is left in the A and Q registers:



ERROR RETURN: None

STORAGE: 44 locations

PURPOSE: Repeated use of RANF generates a uniformly distributed sequence of random numbers in either fixed or floating point format. If floating point is used, the numbers range from 0 to 1; if fixed point is used, the range is 0 to 1 (if interpreted as a fraction) or 1 to 2^{47} -1 (if interpreted as an integer). RANFGET allows extract ion of the current generative number, and RANFSET allows it to be replaced.

COMPASS CALLING SEQUENCE:

Call by Value (RANF)

```
        ENA ± 1
L       BRTJ      ($)RANF, ,*
L+1     SLJ       *+1
        00        DICT.
```

A positive parameter to RANF requests a fixed point response, a negative parameter requests floating point.

Call by Name (RANFGET or RANFSET)

```
L       BRTJ      ($)RANFGET, ,* or ($)RANFSET, ,*
L+1     SLJ       *+2
        01        DICT.
L+2     00        ($)R
        NOP       0
```

FORTRAN FUNCTIONS: X = RANF(±1)

CALL RANFGET(R)

CALL RANFSET(R)

STORAGE: 33 locations

ACCURACY: Tests described in COOP routine G5 ANL G550 RANF were performed.

TIMING: Floating point, approximately 34 microseconds
Fixed point, approximately 27 microseconds

MATHEMATICAL METHOD: Given the random integer X_i, compute $T = 5^{15} * X$. Then $X_{i+1} = T$ modulo 2^{47}. Since the low order bits of X_{i+1} have a very short period, the fixed point number returned by RANF consists of bits 47-11 of X_{i+1} followed by bits 58-48 of T. The floating point number returned is $X_i + 1/2^{47}$. X_i is 11060471625_8 initially.

REAL

PURPOSE: To obtain the real part of a complex number Z.

COMPASS CALLING SEQUENCE:

```
L          BRTJ       ($)REAL, , *
L+1        SLJ        *+2
           01         DICT.
L+2        00         ($)Z
```

FORTRAN FUNCTION: REAL(Z)

NORMAL RETURN: The real part of Z is left in the A register.

ERROR MESSAGE: None

STORAGE: 9 locations including AIMAG and CMPLX

TIMING: 46 microseconds

REAL is generally compiled in-line.

Compiled coding sequence:

```
           LDA        Z
```

PURPOSE: To transfer the sign of argument X_2 to the absolute value of argument X_1.

COMPASS CALLING SEQUENCE:

Call by Value (SIGNF or XSIGNF)

	LDA	X_1
	LDQ	X_2
L	BRTJ	($)SIGNF, , * or ($)XSIGNF, , *
L+1	SLJ	*+1
	00	DICT.

Call by Name

L	BRTJ	($)Q8QSIGNF, , * or ($)Q8QXSIGNF, , *
L+1	SLJ	*+2
	02	DICT.
L+2	00	($)$X_1$
	00	($)$X_2$

FORTRAN FUNCTION: SIGNF(X_1,X_2) and XSIGN F(X_1,X_2)

NORMAL RETURN: With SIGNF, the sign is transferred between floating point arguments; with XSIGNF, the sign is transferred between fixed point arguments. In both cases the result is left in the A register.

ERROR MESSAGE: None

STORAGE: 8 locations

ACCURACY: Exact

SIGNF and XSIGNF are generally compiled in-line.

Compiled coding sequence:

	LDA	X1
	LDQ	X2
+	QJP, MI	*+2
	AJP, PL	*+3
	ROP, XOR	A, MZ, A
	SLJ	*+2
	AJP, MI	*+1
	ROP, XOR	A, MZ, A

SINF

PURPOSE: To compute the sine of a floating point argument X in radians.

COMPASS CALLING SEQUENCE:

Call By Value (SINF)

```
        LDA     X
L       BRTJ    ($)SINF, ,*
L+1     SLJ     *+1
        00      DICT.
```

Call by Name (Q8QSINF)

```
L       BRTJ    ($)Q8QSINF, ,*
L+1     SLJ     *+2
        01      DICT.
L+2     00      ($)X
        NOP     0
```

FORTRAN FUNCTION: SINF(X)

NORMAL RETURN: The value of SINF(X) is in the A register.

ERROR MESSAGE: ERROR DETECTED IN SIN/COS

A = (value)   Q = (value)

ARG GT 2**36.

If the absolute value of the argument exceeds 2^{36}, the message is printed on the standard output unit, and the absolute value of the argument is returned in the A register.

STORAGE: 138 locations

ACCURACY: Average accuracy in the range $(-\frac{\pi}{2}, \frac{\pi}{2})$ exceeds 35 bits; the worst known case is 34 bits. When $|x| > \frac{\pi}{2}$, the relative error is less than 2^{-35}.

TIMING: 124 microseconds

MATHEMATICAL METHOD:

The method of partitioned polynomials, described in Control Data Technical Report 52 is used. Briefly, for $|x|$ less than approximately 50°, a polynomial in x is evaluated; for approximately 50° < $|x|$ < 90°, a polynomial in $(\frac{\pi}{2} - x)$ is evaluated, in both cases, the actual angle determines how many terms in each polynominal are evaluated.

PURPOSE: To convert a double precision argument X, to single precision floating point.

COMPASS CALLING SEQUENCE:

Call by value (SNGL)

	LDA	X
L	BRTJ	($)SNGL, ,*
L+2	SLJ	*+1
	00	DICT.

Call by name (Q8QSNGL)

L	BRTJ	($)Q8QSNGL, ,*
L+1	SLJ	*+2
	01	DICT.
L+2	00	($)X
	NOP	0

FORTRAN FUNCTION: SNGL (X)

NORMAL RETURN: Returns with the single precision floating point result in the A register.

ERROR MESSAGE: None

STORAGE: 24 locations including DBLE

ACCURACY: 37th bit is rounded into 36th bit.

SQRTF

PURPOSE: To compute the square root of a real argument X.

COMPASS CALLING SEQUENCE:

Call by Value (SQRTF)

```
         LDA       X
L        BRTJ      ($)SQRTF, , *
L+1      SLJ       *+1
         00        DICT.
```

Call by Name (Q8QSQRTF)

```
L        BRTJ      ($)Q8QSQRTF, , *
L+1      SLJ       *+2
         01        DICT.
L+2      00        ($)X
         NOP       0
```

FORTRAN FUNCTION: SQRTF(X)

NORMAL RETURN: Return with the result in the A register.

ERROR MESSAGE: ERROR DETECTED IN SQRTF

A = (value)   Q = (value)

NEG ARG.

If the argument is negative, this message is printed on the standard output unit and a normal exit is taken from SQRTF.

STORAGE: 31 locations

ACCURACY: 35 bits average

TIMING: 107 microseconds

NOTE: Refer to CO-OP routine B4 ANL ANL B450 SQRTF.

MATHEMATICAL METHOD:

$$X = 2^{2N} * \bar{X} \qquad 1/2 \leqq \bar{X} < 2$$

$$\sqrt{X} = 2^{N} * \sqrt{\bar{X}}$$

$\sqrt{\bar{X}}$ is computed using the Newtonian approximation.

The initial value is a Chebychev approximation.[1]

$$Y_0 = A - \frac{B}{A+\bar{X}}$$

$$A = 3.0903156$$
$$B = 8.5500505$$

Two iterations follow:

$$Y_i = 1/2\left(Y_{i-1} + \frac{\bar{X}}{Y_{i-1}}\right) \qquad i = 1, 2$$

Then $Y_2 \approx \sqrt{\bar{X}}$

---

[1] W. J. Cody, "Double Precision Square Root for the Control Data Corporation 3600, AMD Tech. Memo No. 55, p. 18, Applied Mathematics Division, ANL (unpublished).

TANF

PURPOSE: To compute the tangent of the floating point in radians of X.

COMPASS CALLING SEQUENCE:

Call by Value (TANF)

```
               LDA            ($)X
L              BRTJ           ($)TANF, , *
L+1            SLJ            *+1
               00             DICT.
```

Call by Name (Q8QTANF)

```
L              BRTJ           ($)Q8QTANF, , *
L+1            SLJ            *+2
               01             DICT.
L+2            00             ($)X
               00
```

FORTRAN FUNCTION: TANF(X)

NORMAL RETURN: Floating point result is left in the A register.

ERROR MESSAGE: If the argument is greater than 2^{36}:

ERROR DETECTED IN TANF

A = (value)   Q = (value)

ARG GT MAX.

STORAGE: 74 locations

ACCURACY: TANF (X) was evaluated for 5000 arguments between -5 and +10 radians. The average spacing between the arguments was .003; actual spacing was a random number between .002 and .004. TANF (X) was compared with SINF (X)/COSF (X), which is known to have an average accuracy of 1 part in 2^{35}, by doing an integer subtraction.

TIMING:	
$-\pi/4 < R < \pi/4$	140 microseconds
$\pi/4 < \|R\| < \pi/2$	149 microseconds
$\pi/2 < \|R\|$	187 microseconds

The differences are as follows:

Difference	-3	-2	-1	0	1	2	3
Occurrences	202	483	897	1180	869	431	215

The difference exceeded 3 on 673 samples. Sizable differences occurred only when X was approximately equal to $k\pi/2$, for $k = \pm 2, \pm 3, \ldots$

In this case, when $2^{-n-1} < \left|\frac{k\pi}{2} - X\right| < 2^{-n}$, an upper bound on the magnitude of the relative error is approximately $\frac{2^n}{3} \cdot 2^{-35}$.

MATHEMATICAL METHOD[1]:

The approximation for computing the tangent of the angle R in radians is given by the following equations:

For $-\pi/4 \leqq R \leqq \pi/4$

$$f(W) = \tan R = \frac{W}{S(W^2)}$$

$$W = R * 2/\pi$$

$$S(W^2) = A + W^{2}* \left[B + \frac{C}{D + W^2 + \frac{E}{F + W^2}} \right]$$

A = .63661 97723 67596

B = -.07531 94869 91705

C = 3.88560 57227 68290

D = -14.87026 86251 97861

E = -57.81869 13873 68667

F = -9.32191 89536 46030

For $|R| > \pi/4$, the argument is reduced to its basic equivalent angle by division by $\pi/4$. Giving R the value of the remainder of this division, the following table shows the values of W and $f(W)$.

Octant	W	TANF
First or fifth	$(2/\pi) * R$	$f(W)$
Second or sixth	$(2/\pi) * (\pi/4 - R)$	$1/f(W)$
Third or seventh	$-(2/\pi) * R$	$f(W)$
Fourth or eighth	$(2/\pi) * (R - \pi/4)$	$1/f(W)$

[1] Maehly, Hans J., Approximations for the Control Data 1604, Control Data Corporation publication number 516.

TANHF

PURPOSE: To compute the single precision hyperbolic tangent of a single precision argument X.

COMPASS CALLING SEQUENCE:

Call by Value (TANHF)

```
            LDA         X
L           BRTJ        ($)TANHF, , *
L+1         SLJ         *+1
            00          DICT.
```

Call by Name (Q8QTANHF)

```
L           BRTJ        ($)Q8QTANHF, , *
L+1         SLJ         *+2
            01          DICT.
L+2         00          ($)X
            NOP         0
```

FORTRAN FUNCTION: TANHF(X)

NORMAL RETURN: Floating point result is left in the A register.

ERROR MESSAGE: None

STORAGE USED: 71 locations

ACCURACY: Average 35 bits; worst known relative error 2^{-34}

MATHEMATICAL METHOD:

$$E^{2X} = 2^{I}2^{f} = 2^{I} * E^{f*\mathrm{Ln}\,2}$$

1. $\mathrm{TANH}(X) = \left[(E^{2X} - 1.)/(E^{2X} + 1.)\right]$ for $.3 < |X| < 12.48$

2. $\mathrm{TANH}(X) = X - \frac{X^3}{3} + \frac{2X^5}{15} - \frac{17X^7}{315} + \frac{62X^9}{2835} - .00886323553X^{11} + .00357X^{13}$

   for $0 < |x| < .3$

3. $\mathrm{TANH}(0) = 0$

4. $\mathrm{TANH}(X) = \pm 1.0$ for $|x| > 12.48$

PURPOSE: To obtain the current reading of the clock in milliseconds.

COMPASS CALLING SEQUENCE:

```
L       BRTJ    ($)TIMEF,,*
L+1     SLJ     *+2
        01      DICT.
L+2     00      ($)X
        NOP     0
```

X is a dummy variable

FORTRAN FUNCTION: TIMEF(X)

NORMAL RETURN: Floating point result is left in the A register.

ERROR MESSAGE: None

STORAGE: 5 locations

XDIMF

PURPOSE: To determine the difference between two fixed point numbers, I_1 and I_2.

COMPASS CALLING SEQUENCE:

Call by Value (XDIMF)

```
            LDA      I1
            LDQ      I2
L           BRTJ     ($)XDIMF, , *
L-1         SLJ      *+1
            00       DICT.
```

Call by Name (Q8QXDIMF, , *)

```
L           BRTJ     ($)Q8QXDIMF, , *
L-1         SLJ      *+2
            02       DICT.
L-2         00       ($)I1
            00       ($)I2
```

FORTRAN FUNCTION: XDIMF(I_1,I_2)

NORMAL RETURN: If $I_1 > I_2$, the fixed point difference is left in the A register.
If $I_1 \leqq I_2$, zero is left in the A register.

ERROR MESSAGE: ERROR DETECTED IN XDIMF

A = (value) Q = (value)

OVERFLOW ERROR.

This message is written on the standard output unit and a normal exit is taken from XDIMF when $(I_1-I_2)| > 2^{47}-1$.

STORAGE: 22 locations

ACCURACY: Exact

XDIMF is generally compiled in-line.

Compiled coding sequence:

```
            LDA      X1
            SUB      X2
+           AJP,PL   *+1
            ENA      0
```

PURPOSE: To convert a floating point number X to the nearest fixed point integer with a magnitude less than or equal to the absolute value of X.

COMPASS CALLING SEQUENCE:

Call by Value (XFIXF)

```
        LDA     X
L       BRTJ    ($)XFIXF, , *
L+1     SLJ     *+1
        00      DICT.
```

Call by Name (Q8QXFIXF)

```
L       BRTJ    ($)Q8QXFIXF, , *
L+1     SLJ     *+2
        01      DICT.
L+2     00      ($)X
        NOP     0
```

FORTRAN FUNCTION: XFIXF(X)

NORMAL RETURN: Fixed point value is left in the A register.

ERROR MESSAGE: ERROR DETECTED IN XFIXF

A = (value)   Q = (value)

INTEGER TOO BIG.

This message is printed on the standard output unit and a normal exit is taken from XFIXF when the floating point argument is too large to be converted to an integer.

STORAGE: 26 locations

ACCURACY: Exact

XINTF

PURPOSE: To convert a floating point argument X to the nearest fixed point integer with a magnitude less than or equal to the absolute value of X.

COMPASS CALLING SEQUENCE:

Call by Value (XINTF)

```
            LDA         X
L           BRTJ        ($)XINTF, , *
L+1         SLJ         *+1
            00          DICT.
```

Call by Name (Q8QXINTF)

```
L           BRTJ        ($)Q8QXINTF, , *
L+1         SLJ         *+2
            01          DICT.
L+2         00          ($)X
            NOP         0
```

FORTRAN FUNCTION: XINTF(X)

NORMAL RETURN: The fixed point result is left in the A register.

ERROR MESSAGE: ERROR DETECTED IN XFIXF

A = (value) Q = (value)

INTEGER TOO BIG.

This message is written on the standard output unit and a normal exit is taken from XINTF when X is too large to convert to an integer.

STORAGE: 26 locations

ACCURACY: Exact

NOTE: XINTF is the same routine as XFIXF.

PURPOSE: To determine the maximum value of the fixed point arguments $I_1, I_2, \ldots, I_n$.

COMPASS CALLING SEQUENCE:

L	BRTJ	($)XMAX0F, , *
L+1	SLJ	*+m
	n	DICT.
L+2	00	($)$I_1$
	00	($)$I_2$
	.	
	.	
	.	
	00	($)$I_n$

m = (n + 1)/2 + 1, n = number of parameters

FORTRAN FUNCTION: XMAX0F($I_1, I_2, \ldots, I_n$)

NORMAL RETURN: Fixed point value of the largest argument is left in the A register.

ERROR CODE: None

STORAGE: 81 locations for MIN and MAX functions

XMAX1F

PURPOSE: To determine the maximum value of the floating point arguments $X_1, X_2, \ldots, X_n$.

COMPASS CALLING SEQUENCE:

```
L        BRTJ     ($)XMAX1F, , *
L+1      SLJ      *+m
         n        DICT.
L+2      00       ($)X1
         00       ($)X2
         .
         .
         .
         .
         00       ($)Xn
```

$m = (n + 1)/2 + 1$, n = number of parameters.

FORTRAN FUNCTION: $XMAX1F(X_1, X_2, \ldots, X_n)$

NORMAL RETURN: Fixed point value of the largest argument is left in the A register.

ERROR RETURN: This return is taken if the largest floating point argument results in an absolute fixed point value greater than $2^{47} -1$.

ERROR CODES: See XFIXF.

STORAGE: 81 locations for MIN and MAX functions

PURPOSE: To determine the minimum of the fixed point arguments $I_1, I_2, \ldots, I_n$.

COMPASS CALLING SEQUENCE:

```
L       BRTJ    ($)XMIN0F, , *
L+1     SLJ     *+m
        n       DICT.
L+2     00      ($)I1
        00      ($)I2
        .
        .
        .
        00      ($)In
```

m = (n + 1)/2 + 1, n = number of parameters

FORTRAN FUNCTION: XMIN0F($I_1, I_2, \ldots, I_n$)

NORMAL RETURN: Fixed point value of the smallest argument is left in the A register.

ERROR CODES: None

STORAGE: 81 locations for MIN and MAX functions

XMIN1F

PURPOSE: To determine the minimum value of the floating point arguments $X_1, X_2, \ldots, X_n$.

COMPASS CALLING SEQUENCE:

```
L       BRTJ    ($)XMIN1F, , *
L+1     SLJ     *+m
        n       DICT.
L+2     00      ($)X1
        00      ($)X2
        .
        .
        .
        00      ($)Xn
```

$m = (n + 1)/2 + 1$, n = number of parameters

FORTRAN FUNCTION: XMIN1F($X_1, X_2, \ldots, X_n$)

NORMAL RETURN: Fixed point value of the smallest argument is left in the A register.

ERROR RETURN: This return is taken if the smallest floating point argument results in an absolute fixed point value greater than $2^{47}-1$.

ERROR CODES: See XFIXF.

STORAGE: 81 locations for MIN and MAX functions

PURPOSE: To compute the value of the fixed point argument I modulo J.

COMPASS CALLING SEQUENCE:

Call by Value (XMODF)

	LDA	I
	LDQ	J
L	BRTJ	($)XMODF, , *
L+1	SLJ	*+1
	00	DICT.

Call by Name (Q8QXMODF)

L	BRTJ	($)Q8QMODF, , *
L+1	SLJ	*+2
	02	DICT.
L+2	00	($)I
	00	($)J

FORTRAN FUNCTION: XMODF (I,J)

NORMAL RETURN: The value of I modulo J is left in the A register. The result is negative only if I and J have unlike signs.

ERROR MESSAGE: ERROR DETECTED IN XMODF

A = (value) Q = (value)

DIVISOR IS ZERO.

This message is printed on the standard output unit and a normal exit is taken from XMODF when J is zero.

STORAGE: 21 locations

ACCURACY: Exact

PURPOSE: To raise a floating point number to an integer power: X**I.

COMPASS CALLING SEQUENCE:

Call by Value (XTOI)

```
            LDA         X
            LDQ         I
L           BRTJ        ($)XTOI, , *
L+1         SLJ         *+1
            00          DICT.
```

Call by Name (Q8QXTOI)

```
L           BRTJ        ($)Q8QXTOI, , *
L+1         SLJ         *+2
            02          DICT.
L+2         00          ($)X
            00          ($)I
```

FORTRAN FUNCTION: XTOI(X,I)

NORMAL RETURN: Return with the result in the A register.

ERROR MESSAGE: ERROR DETECTED IN XTOI

A = (value)   Q = (value)

Error Code

Error Code

X = 0, I = 0 OR NEG.	If X is zero and I is zero or negative
EXP OVERFLOW.	If $\|X^I\| \geq 10^{308}$.

XTOI calls POWRF for powers greater than 1023. See POWRF for other error codes.

STORAGE: 87 locations including Q2Q07110

ACCURACY:

36 bits for $|I| \leq 1023$, 34 bits for $|I| > 1023$

MATHEMATICAL METHOD:

If $|I| > 1023$ call POWRF.

If not, since $I = P*2^0 + Q*2^1 + R*2^2 + S*2^3 + T*2^4 + U*2^5 + V*2^6 + W*2^7 + Y*2^8 + Z*2^9$
(P,Q,R,S,T,U,V,W,Y,Z = 1 or 0)

then $X^I = X^{P*2^0} * X^{Q*2^1} * X^{R*2^2} * X^{S*2^3} * X^{T*2^4} * X^{U*2^5} * X^{V*2^6} * X^{W*2^7} * X^{Y*2^8} * X^{Z*2^9}$

Compute X^2, X^4, X^8, X^{16}, X^{32}, X^{64}, X^{128}, X^{256}, X^{512} and
multiply the powers of X that are present as indicated by the bit pattern of I.

CONTROL DATA

CORPORATION

COMMENT AND EVALUATION SHEET

3600/3800 Computer Systems
Library Functions

Pub. No. 60056400, B July, 1966

THIS FORM IS NOT INTENDED TO BE USED AS AN ORDER BLANK. YOUR EVALUATION OF THIS MANUAL WILL BE WELCOMED BY CONTROL DATA CORPORATION. ANY ERRORS, SUGGESTED ADDITIONS OR DELETIONS, OR GENERAL COMMENTS MAY BE MADE BELOW. PLEASE INCLUDE PAGE NUMBER REFERENCE.

FROM NAME: ______________________________

BUSINESS
ADDRESS: ______________________________

NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

FOLD ON DOTTED LINES AND STAPLE

STAPLE STAPLE

FOLD FOLD

FIRST CLASS
PERMIT NO. 8241

MINNEAPOLIS, MINN.

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION
Documentation Department
3145 PORTER DRIVE
PALO ALTO, CALIFORNIA

FOLD FOLD

STAPLE STAPLE

3600/3800 LIBRARY FUNCTIONS

CONTROL DATA
CORPORATION

CORPORATE HEADQUARTERS, 8100 34th AVE. SO., MINNEAPOLIS, MINN. 55440
SALES OFFICES AND SERVICE CENTERS IN MAJOR CITIES THROUGHOUT THE WORLD

Pub. No. 60056400

Litho in U.S.A.