the input table and the corresponding Y values, Y_2 and Y_1, are extracted. The difference in Y values is divided by the difference in X values. Y as a function of X is obtained by use of expression (1). CUTIE is stepped by one and the subprogram returns to the user subprogram.

d. Expressions.

$$y = \left(\frac{X - X_1}{X_2 - X_1}\right)\left(Y_2 - Y_1\right) + Y_1 \qquad (1)$$

2-95. SUBPROGRAM D50 (INTRØG). INTRØG interrogates a simulated sense switch. The FORTRAN II reference statement is CALL INTRØG (N).

a. Inputs. The input is the argument N which is the number of the simulated sense switch to be interrogated by INTRØG.

b. Outputs. The output is the setting of ITRØG. If switch N is set ØN, ITRØG is set to one. If N is set ØFF, ITRØG is cleared to zero. If neither of these conditions exist, INTRØG will print and write the statement: ILLEGAL CONTROL REGISTER SETTING. CONTENTS OF SW XX IS XXXXXX.

c. Program Logic. FD D50. Switch N is examined. If ØFF, ITRØG is cleared to zero; if ØN, ITRØG is set to one. Control is returned to the user subprogram. If switch N is neither ØN nor ØFF, BAREA reads the B subprograms and the B Common into core, if not already there. The output statement is printed and written. The subprogram exits to RLLBCK for return to the previous check point. If U16 (Common Merger) is being performed, the output statement is written and the subprogram exits to RLLBCK for return to the previous check point.

2-96. SUBPROGRAM D04(LCTØLG). LCTØLG computes geographic latitude. The FORTRAN II reference statement is CALL LCTØLG.

a. Inputs. The duplexed input of geocentric latitude is expressed in degrees and decimal parts of a degree. Geocentric latitude is defined GCLAT, a variable of dimension two. A conversion factor from radians to degrees in FRTØD is also an input to LCTØLG.

b. Outputs. The duplexed output of a geographic latitude is expressed in degrees and decimal parts of a degree. Geographic latitude is defined GGLAT, a variable of dimension two.

c. Program Logic. FD D04

(1) Steps 1-3. The input geocentric latitude (1) is examined. If zero, the duplexed output geographic latitudes are set to 0.0 and control is returned to the user subprogram; if not zero, the subprogram continues at step 4.

(2) Steps 4-7. The GRASE area 1 to 10 is set up with the coefficients used in expression (1). Expression (1) evaluates geographic latitude (2) as a function of geocentric latitude. If the input geocentric latitude (2) is negative the sign of the computed geographic latitude is reversed. The subprogram continues at step 8.

(3) Steps 8-12. Geocentric latitude (1) is examined. If equal to 90.0, geographic latitude (1) is set to 90.0 and the subprogram continues at step 13. If not equal to 90.0, its value is converted to radians. Expression (2) evaluates geographic latitude (1) which is then converted to degrees. The subprogram continues at step 13.

(4) Steps 13-14. The duplexed values are set equal. CUTIE is stepped by one and control is returned to the user subprogram.

d. Expressions.

$$\begin{aligned} GCLAT\ (2) = {} & \Big| 1.3216901 \times 10^{-17} \left[GCLAT\ (2)\right]^{8} \\ & - 4.1794914 \times 10^{-15} \left[GCLAT\ (2)\right]^{7} \\ & + 7.2770987 \times 10^{-14} \left[GCLAT\ (2)\right]^{6} \\ & + 8.5559479 \times 10^{-11} \left[GCLAT\ (2)\right]^{5} \\ & - 1.1979582 \times 10^{-11} \left[GCLAT\ (2)\right]^{4} \\ & - 1.3805593 \times 10^{-16} \left[GCLAT\ (2)\right]^{3} \\ & - 1.1969406 \times 10^{-8} \left[GCLAT\ (2)\right]^{2} \\ & + 1.0067381 \left[GCLAT\ (2)\right] \\ & - 1.9073486 \times 10^{-6} \Big| \end{aligned} \qquad (1)$$

$$GCLAT\ (1) = \tan^{-1} \left[1.0067384 \frac{\sin a}{\cos a} \right] \qquad (2)$$

where a = geocentric latitude in radians

2-97. SUBPROGRAM D01 (LGTØLC). LGTØLC computes geocentric latitude as a function of geographic latitude. The FORTRAN II reference statement is CALL LGTØLC.

a. Inputs. The duplexed input of geographic latitude is expressed in degrees and decimal parts of a degree. Geographic latitude is defined GGLAT, a variable of dimension two. A conversion factor from radians to degrees in FRTØD is also an input to LGTØLC.

b. Outputs. The duplexed output of geocentric latitude is expressed in degrees and decimal parts of a degree. Geocentric latitude is defined GCLAT, a variable of dimension two.

c. Program Logic. FD D01

(1) Steps 1-3. The input geographic latitude is examined. If zero, the duplexed output geocentric latitudes (1) and (2) are set to 0.0 and control is returned to the user subprogram; if not zero, the subprogram continues at step 4.

(2) Steps 4-7. The GRASE area 1-10 is set up with the coefficients used in expression (1). Expression (1) evaluates geocentric latitude (2) as a function of geographic latitude. If the input geographic latitude (2) is negative the sign of the computed geocentric latitude is reversed. The subprogram continues at step 8.

(3) Steps 8-12. Geographic latitude (1) is examined. If equal to 90.0, geocentric latitude (1) is set to 90.0 and the subprogram continues at step 13. If not equal to 90.0, its value is converted to radians. Expression (2) evaluates geocentric latitude (1) which is then converted to degrees. The subprogram continues at step 13.

(4) Steps 13-14. The duplexed values are set equal. CUTIE is stepped by one and control is returned to the user subprogram.

d. Expressions.

$$\begin{aligned} \text{GCLAT (2)} = {} & 1.2795114 \times 10^{-19} \left[\text{GGLAT (2)}\right]^9 \\ & - 5.8498736 \times 10^{-17} \left[\text{GGLAT (2)}\right]^8 \\ & + 1.1727320 \times 10^{-14} \left[\text{GGLAT (2)}\right]^7 \\ & - 8.8674860 \times 10^{-13} \left[\text{GGLAT (2)}\right]^6 \\ & - 3.1092951 \times 10^{-11} \left[\text{GGLAT (2)}\right]^5 \\ & - 1.5392019 \times 10^{-9} \left[\text{GGLAT (2)}\right]^4 \\ & + 1.3737086 \times 10^{-6} \left[\text{GGLAT (2)}\right]^3 \\ & - 2.3275893 \times 10^{-7} \left[\text{GGLAT (2)}\right]^2 \\ & + 9.9330703 \times 10^{-1} \left[\text{GGLAT (2)}\right] \\ & - 4.7683716 \times 10^{-7} \end{aligned} \qquad (1)$$

$$\text{GGLAT (1)} = \tan^{-1} \left[.99330666 \frac{\sin a}{\cos a}\right] \qquad (2)$$

where a = geographic latitude in radians

2-98. SUBPROGRAM D32 (LØCALT). LØCALT computes the missile geometric altitude above the earth ellipsoid and the current missile altitude above geoid. The FORTRAN II reference statement is CALL LØCALT.

a. Inputs. The inputs are as follows:

COMMON TAG	DIMENSION	ITEM	UNITS
FSPPS	2,3	Missile position vector	feet
PLWGS	2	Geoidal separation at launch pad	feet
PRWGS	2,1,1	Geoidal separation at radar	feet
GCRAD	2	Radius of earth ellipsoid at this geocentric	feet
FMALT	2	Geometric altitude of point above earth ellipsoid	feet
FRDSQ	2	Square of current missile distance from center of earth	ft^2
FRTØD	2	Conversion factor	deg/rad
SW(24)		If ØN, guided flight simulation is in progress	
FXYPT	2	X, Y projection of current missile position vector	feet

b. Outputs. Each output is in single-precision floating point form except the square root of the missile radius vector which is in double-precision form. The outputs are as follows:

COMMON TAG	DIMENSION	ITEM	SYMBOL	UNITS
FSGCL	2	Sine of the missile geocentric latitude	$\sin L_{CM}$	

COMMON TAG	DIMENSION	ITEM	SYMBOL	UNITS
GCLAT	2	Geocentric latitude	L_C	degrees
FGALT	2	Current missile altitude above geoid		feet
FRDUS	2	Radius vector magnitude from point to center of earth	r_m	feet
FMALT	2	Geometric altitude of point above earth ellipsoid	h'_m	feet

c. Program Logic. FD D32

(1) Steps 1-5. VECMAG computes the radius vector of the missile from the center of the earth. Expression (1) evaluates the sine of the missile geocentric latitude. ARCSIN computes the arcsine which is converted from radians to degrees to produce the geocentric latitude. ELLRAD computes the radius of the earth ellipsoid.

(2) Steps 6-11. The missile altitude above the earth ellipsoid is obtained by subtracting the earth ellipsoid radius from the missile radius vector. INTRØG interrogates SW(24) to determine if the guided flight simulator (powered phase) is in progress. If ØN, the geoidal separation at radar is subtracted from the geometric altitude for computation of the current missile altitude. If ØFF, the geoidal separation at the launch pad is subtracted from the geometric altitude for computation of the current missile altitude. CUTIE is stepped by one and control is returned to the user subprogram.

d. Expressions.

$$\sin L_{CM} \quad \frac{Lc}{r_m} \qquad (1)$$

2-99. SUBPROGRAM D49 (NEGSQR). NEGSQR prints an indication that a negative square root was attempted in SQRØØT. The FORTRAN II reference statement is CALL NEGSQR (X).

a. Inputs. The input is the argument X which is the argument given to SQRØØT for square root extraction and which SQRØØT has found to be negative. X is a duplexed quantity expressed in single-precision floating point form.

b. Outputs. The output is the following statement printed and written: NEGATIVE SQUARE ROOT ATTEMPTED. ARGUMENTS WERE ____ AND ____ .

c. Program Logic. BAREA re-establishes the B subprograms containing the library print routines in core. NEGSQR prints and writes the output statement containing the argument X. Control is transferred to RLLBCK with the IFLAG setting of the subprogram which supplied the input argument X to SQRØØT.

2-100. SUBPROGRAM D30 (RANGLE). RANGLE computes re-entry angle and speed of the missile relative to the surface of the rotating earth. The FORTRAN II reference statement is CALL RANGLE.

a. Inputs. The inputs are as follows:

COMMON TAG	DIMENSION	ITEM	SYMBOL	UNITS
FSPPS	2,3	Current missile position vector-single precision	$\vec{r}_m$	feet
FSPVL	2,3	Current missile velocity vector-single precision	$\vec{V}_m$	ft/sec
GØMGA	2	Rate of earth's rotation	Ω	rad/sec
FRTØD	2	Conversion constant - radians to degrees		deg/rad

b. Outputs. The outputs are as follows:

COMMON TAG	DIMENSION	ITEM	SYMBOL	UNITS
GCLAT	2	Geocentric latitude of current missile position	L_C	degrees
GGLAT	2	Geographic latitude of current missile position		degrees
GRSPD	2	Re-entry speed of missile	S_{RE}	ft/sec
GRNGL	2	Re-entry angle of missile (with respect to local horizontal)		degrees

c. Program Logic. FD D30

(1) Steps 1-2. Dual computations are performed to obtain duplexed outputs. Expression (1) evaluates the

geocentric latitude of the missile. VECMAG computes the magnitude of the missile position vector. This vector is divided into the Z component of the missile position vector to give the sine of the geocentric latitude. ARCSIN then computes the geocentric latitude.

(2) Steps 3-6. The coordinates of the vertical vector $\vec{U}$ are obtained. The X and Y coordinates of $\vec{U}$ are set to the X and Y coordinates of the input missile position. Expression (2) evaluates the Z component. VECMAG computes the XY projection and LCTØLG computes the geographic latitude. TANGNT computes the tangent.

(3) Steps 7-12. Expressions (3), (4), and (5) evaluate coordinates of the missile velocity vector $\vec{W}$ relative to the surface of the rotating earth. VCDØTP computes the magnitude of vectors $\vec{U}$ and $\vec{W}$ and the included angle. The magnitude of $\vec{W}$ is the missile re-entry speed. The complement of the angle between the two vectors is the re-entry angle. CUTIE is stepped by one and control is returned to the user subprogram.

d. <u>Expressions</u>.

$$L_C = \arcsin \frac{Z}{|\vec{r}_m|} \qquad (1)$$

$$Z' = \sqrt{X^2 + Y^2} \tan (L_g) \qquad (2)$$

$$W_x = \dot{X} + \Omega Y \qquad (3)$$

$$W_y = \dot{Y} - \Omega X \qquad (4)$$

$$W_Z = \dot{Z} \quad (5)$$

where

L_C = geocentric latitude

L_g = geographic latitude

Z' = Z component of vertical vector $\vec{U}$

All other terms are defined in the Inputs and Outputs paragraphs.

2-101. SUBPROGRAM D39 (RLLBCK). RLLBCK determines the subprogram in which an error occurred and establishes a re-entry point if possible. The FORTRAN II reference statement is CALL RLLBCK.

a. Inputs. The inputs are as follows:

COMMON TAG	DIMENSION	ITEM
ØFTEN	1	Number of RLLBCK errors since last check point
RNMAX	1	Maximum number of tolerable inconsistent machine errors between check points
IFLAG	1	Identifier of subprogram in which an error is detected
CUTIE	1	Successful computation unit tally
RBKT1	10	Table of IFLAG values since last check point
RBKT2	10	Table of CUTIE values since last check point
DMPCT	1	Common table tape dump counter

b. Outputs. The outputs are the updated registers OFTEN, RBKT1, RBKT2, IFILE, and IRECR. Depending upon the input parameters, one or more of the following statements is printed and written.

a. EXCESSIVE INCONSISTENT MALFUNCTION DETECTED

b. CONSISTENT MALFUNCTION IS DETECTED

c. MALFUNCTION DETECTED IN SUBPROGRAM _______
COMPUTATION UNIT TALLY IS _______

d. ROLLBACK IMPOSSIBLE

e. DIFFICULTIES ENCOUNTERED IN ATTEMPTING COMMON TABLE RESTORATION FROM TAPE 5 DURING ROLLBACK

c. Program Logic. FD D39

(1) Steps 1-8. BAREA re-establishes the B subprograms and B Common in core. If there were no illegal guidance commands, that is RLLBCK was not entered from ILLCMD, the subprogram continues at step 10. If there has been an illegal guidance command, INTRØG interrogates SW(75) to determine if OTC was requested (SW(75) = ØN). If ØFF, control is transferred to step 9. If ØN, INTRØG interrogates SW(158) to determine if this was the first pass through OTC (SW(158) = ØN). If ØFF, control is transferred to step 9. If ØN, SW(153) is set ØN to indicate that B2 (ØTCNT) subprograms are in core and SW(151) is set ØFF to indicate that a constraint has not been exceeded in the last target simulation. Control is returned to the major control subprogram DØCNT.

(2) Step 9. The subprogram exits to HALT for manual intervention.

(3) Steps 10-13. If there have been no RLLBCK errors since the last check point, control is transferred to step 19. If the number of RLLBCK errors is not equal to or greater than the maximum number of tolerable inconsistent machine errors between check points, control is transferred to step 14. Otherwise statements a and c are printed and written and control is transferred to step 9.

(4) Steps 14-18. If IFLAG is equal to any value in table RKBT1, and CUTIE is equal to any value in table RBKT2, the same error has occurred twice. If both of these conditions exist, statement b is printed and written and control is transferred to step 3. to allow continuation of an OTC run if a consistent malfunction is encountered for a target. Otherwise, the subprogram continues at the next step.

(5) Steps 19-22. ØFTEN is stepped by one. The contents of ØFTEN, IFLAG, and CUTIE are stored in a work area. Statement c is printed and written. If an error occurred before the first check point, statement d is printed and written, and control is transferred to step 3. Otherwise the subprogram continues at the next step.

(6) Steps 23-29. The contents of SW(153)-SW(157) are saved. Tape 5 is backspaced by BSREC. IFILE and IRECR are set to zero and tape 5 is read by U04. INTRØG interrogates SW(70) to determine if an error occurred in U04. If ØN, the subprogram continues at the next step; if ØFF, control is transferred to step 31.

(7) Step 30. Statement e is printed and written and control is transferred to step 3.

(8) Steps 31-36. The settings of SW(153)-SW(157) are restored. INTRØG interrogates SW(123) to determine if a tape redundancy has occurred in M04 (SW(123) = ØN). If ØN, control is transferred to step 30; if ØFF, the contents of ØFTEN are restored. The previously stored values of IFLAG and CUTIE are placed in the arrays RBKT1 and RBKT2. BAREA re-establishes the B subprograms and B Common in core. The sub-program entry point of the previous check point in AGAIN is obtained by LAGAIN and control is transferred to that point.

2-102. SUBPROGRAM C32 (RØUND). RØUND rounds a maximum of two duplexed double-precision numbers to the single-precision equivalents. The FORTRAN II reference statement is CALL RØUND (JJ, B, D).

a. Inputs. The inputs are the arguments JJ and B. JJ defines the number of duplexed double-precision numbers to be rounded and B defines the first of the consecutive registers containing the duplexed double-precision numbers to be rounded.

b. Outputs. The output is the argument D, which defines the first register containing the duplexed single-precision equivalent of the input argument B.

c. Program Logic. Rounding is performed by adding twice the low order part to the high order part of a double-precision number as illustrated in expressions (1) and (2).

d. Expressions.

$$D_1 = B_2 + (B_4 + B_4) \quad (1)$$

$$D_2 = B_6 + (B_8 + B_8) \quad (2)$$

D_1 and D_2 are the duplexed single-precision outputs. B_2 and B_6 are the duplexed high order part of the numbers to be rounded, and B_4 and B_8 are the respective low order parts.

2-103. SUBPROGRAM D95 AND D96 (SAVE4 AND RTRN4). SAVE4 saves the return path of the user subprogram and RTRN4 returns along this path. The FORTRAN II reference statement is CALL SAVE4 (N) or CALL RTRN4 (N).

a. Inputs. The input to the argument is N, which is a number from one to four depending upon the hierarchy of the user subprogram. The contents of IFØUR are also any input to SAVE4.

b. Outputs. The output from SAVE4 is the return path saved in IFØUR. No outputs are defined for RTRN4.

c. Program Logic.

(1) SAVE4. The return path of the user subprogram is established in IFØUR in accordance with the argument which is an increment determined by the hierarchy of the subprogram calling SAVE4. The subprogram exits to the user subprogram.

(2) RTRN4. Transfer is made along the path saved in IFØUR, which is stepped by the argument in accordance with the hierarchy of the user subprogram.

2-104. SUBPROGRAM D09 (SINE). SINE computes the sine of any angle expressed in radians. The FORTRAN II reference statement is CALL SINE (T, SD).

a. Inputs. The input is the duplexed argument T which defines a positive or negative angle in radians expressed in double-precision floating point form. If T is actually single precision, the least significant registers of T contain zeros. T is of dimension four.

b. Outputs. The output is the duplexed argument SD which defines the computed sine value in double-precision, floating point form. If SD is actually single precision (for input angle larger than 0.1 radian) the least significant registers of SD contain zeros. SD is of dimension four.

c. Program Logic. FD D09

(1) Steps 1-2. The absolute magnitude of the input angle determines the procedure for computing its sine. If the angle is larger than 0.0001 radian the subprogram continues at step 3. If the angle is less than or equal to 0.0001 radian the output is set equal to the input and the subprogram continues at step 9.

(2) Steps 3-5. The sine of an angle larger than 0.0001 radian is evaluated. For angles larger than 0.1 radian the subprogram computes the sine function in single precision beginning at step 6. For angles between 0.0001 and 0.1 radian

the subprogram computes the sine value in double precision as follows. A matrix of values is established to derive a polynomial from the Taylor sine expansion series. The subprogram then computes the sine of the input angle using the polynomial derived. The subprogram continues at step 9.

(3) Steps 6-8. The sine of an angle larger than 0.1 radian is evaluated. The library sine routine computes the sine for the (a) value of the input argument. The library cosine routine computes the cosine of its compliment angle for the (b) value of the duplexed input. The last two registers of the output are set to zero. The subprogram continues at step 9.

(4) Step 9. CUTIE is stepped by one and control is returned to the user subprogram.

2-105. SUBPROGRAM SOO (SLIBRY). SLIBRY consists of library functions and library subroutines. The functions require the use of floating point numbers. When used with FAP they must be defined exactly as given below. When used with FORTRAN they must be followed by an F, except for EXP(1), EXP(2), and EXP(3), which are used as arithmetic statements. The library subroutines are used to execute tape operations and to simulate FORTRAN input/output routines.

Library Functions	Library Subroutines
SIN	(BST)
CØS	(RWT)
ATAN	(EFT)
ATAND	(IØS)
SQRT	(RDS)
EXP	(WRS)
EXP(1	(BSR)
EXP(2	(WEF)
EXP(3	(REW)
LØG	(ETT)
(FPT)	(RCH)
	(TEF)
	(TCØ)
	(TRC)
	(TØU)
	(TES)

a. Sine and Cosine Functions. These functions compute the sine and cosine of a floating point argument expressed in radians.

(1) Use. The subroutine is used as any library tape function.

with FORTRAN:

Y1 = SINF(X), X in radians

Y2 = CØSF(X), X in radians

with FAP:

CLA X   X in radians

CALL SIN

Normal return with answer in accumulator.

(2) Error Comments. If $900 < |X| \leq 2^{20}\pi$, an inherent loss of two or more places of accuracy results and initiates the printout L = XXXXX, X = ±X.XXXXXXXXE±XX, SINF(X)/COSF (PI/2 - X) LOSING ACCURACY.
If $|X| \geq 2^{20}\pi$, all accuracy is lost and a dump occurs, preceded by the printout L = XXXXX, X = ±X.XXXXXXXXE±XX, ARG OF SINF(X)/COSF(PI/2 - X) TOO BIG, STOP.

(3) Method. The cosine is computed as sin $(\pi/2 - X)$.

$$|X| \leq 10^{-8} \qquad \sin(X) = X$$

$$10^{-8} < |X| \leq 10^{-2} \qquad \sin(X) = X - X^3/3! + X^5/5!$$

$$10^{-2} < |X| \leq 2^{20}\pi \qquad \sin(X) = a \sin\frac{\pi}{2} y,\ a = \pm 1,$$

$$-1 \leq Y \leq 1$$

$$\sin\frac{\pi}{2} y = C_1 Y + C_3 X^3 + C_5 X^5 + C_9 X^9$$

$C_1 = 1.570796318$ $C_3 = -.64596371$ $C_5 = .07968967928$

$C_7 = -.00467376554$ $C_9 = .00015148419$

b. Arctangent Function. This function computes the arctangent of a floating point number, giving the principal value in radians or degrees.

(1) Use. The program is used as any library tape function.

with FORTRAN:

$Y1 = ATANF(X), \; -\pi/2 < Y1 < \pi/2$

$Y2 = ATANDF(X), \; -90° < Y2 < 90°$

with FAP:

```
CLA X
CALL ATAN
Normal return with answer in accumulator
```

(2) Method.

$0 \leq \lvert X \rvert \leq 10^{-8}$	$\arctan X = X$
$10^{-8} \leq \lvert X \rvert \leq 10^{-1}$	$\arctan X = X - X^3/3 + X^5/5 - X^7/7$
$10^{-1} < \lvert X \rvert \leq 2^{128}$	Rand approximation

$$\arctan X = \pi/4 + \sum_{i=0}^{7} C_{2i+1} \left[(X-1)/(X+1)\right]^{2i+1}$$

$C_1 = .99999\ 93329$	$C_9 = .09642\ 00441$
$C_3 = .33329\ 85605$	$C_{11} = -.05590\ 98861$
$C_5 = .19946\ 53599$	$C_{13} = .02186\ 12288$
$C_7 = .13908\ 53351$	$C_{15} = .00405\ 40580$

c. Square Root Function. This function computes the square root of a floating point number, with an error comment for negative arguments.

(1) Use. The subprogram is used as a library tape function.

with FORTRAN:

```
Y = SQRTF(X)
```

with FAP:

```
CLA X
CALL SQRT
Normal return with answer in accumulator
```

(2) Error Comments. If the argument is negative a dump occurs, preceded by the printout L = XXXXX, X = -X.XXXXXXXXE±XX, SQRT UNDEFINED, STOP.

(3) Method. The Newton-Raphson method is used to compute the square root of the absolute value of X in the range $0 \leq |X| \leq 2^{128}$. The initial guess is computed by a method due to Los Alamos. This guess is so accurate that only three iterations are required for eight significant figure accuracy.

d. Logarithms, Base e. This function computes logarithm of a floating point argument to the base e.

(1) Use. The subroutine is used as any library tape function.

with FORTRAN:

```
Y = LØGF(X)
```

with FAP:

CLA X

CALL LØG

Normal return with $Y = \log_e (X)$ in accumulator

(2) Error Comments. If $X \leq 0$, a dump occurs preceded by the printout L = XXXXX, X = -X.XXXXXXXXE±XX, LOGF UNDEFINED, STOP.

(3) Methods.

$$\ell n X = (I + \log_2 f) \log_e 2, \text{ where } X = 2^I \cdot f$$

$\log_2 f$ is computed by Hasting's approximation

$$\log_2 f = -\frac{1}{2} + Z \left[(.5989786492Z^2 + .9614706323)Z^2 + 2.8853912903\right]$$

where $Z = (f - \sqrt{2}/2)\ (f + \sqrt{2}/2)$

e. Exponential Function (EXP). This function computes the exponential function e^x, for $|X| < 88.028$

(1) Use. The subroutine is used as a library tape function.

with FORTRAN:

Y1 = EXPF(X)

with FAP:

CLA X

CALL EXP

Normal return with answer in accumulator

(2) Error Comments. If $X \geq 88.028$, e^x is out of the

range and a dump occurs preceded by the printout L = XXXXX, X = X.XXXXXXXXE±XX EXPF OUT OF RANGE, STOP. If $X \leq -88.028$, e^x is out of range because of underflow. However, a zero is substituted for e^x and the program continues without a printout.

(3) Method. For $|X| < 1/2$ no reduction is necessary. For $|X| \geq 1/2$, $e^x = 2^I e^{f \ln 2}$

where $X/\ln 2 = I + f$, $f < 1$ and I is an integer.

The approximation used is due to E.G. Kogbitliantz.

For $0 \leq Z \leq \ln 2$

$$e^Z \sim 1 + \cfrac{2Z}{2 - Z + \cfrac{Z^2}{20} + \cfrac{98Z^2}{20(42 + Z^2)}}$$

f. Exponential Function (EXP(1). Computes the exponential function M^N for $M \neq 0$.

(1) Use. The subroutine is used as a library tape function.

with FORTRAN:

Y = M**N fixed base, fixed exponent

with FAP:

```
CLA M
LDQ N
TSX EXP(1, 4
```

Normal return with $Y = M^N$ in accumulator.

(2) Method. Successive multiplications.

g. Exponential Function (EXP(2). This function computes the exponential function A^N for $A \neq 0$.

(1) Use. The subroutine is used as a library tape function.

with FORTRAN:

Y = A**N floating base, fixed exponent

with FAP:

```
CAL A
LDQ N
TSX EXP(2, 4)
```

Normal return with $Y = A^N$ in accumulator

(2) Method. Successive multiplications.

h. Exponential Function (EXP(3). This function computes the exponential function A^B for $A \neq 0$.

(1) Use. The subroutine is used as a library tape function.

with FORTRAN:

Y = A**B floating base, floating exponent

with FAP:

```
CLA A
LDQ B
TSX EXP(3, 4)
```

Normal return with $Y = A^B$ in accumulator.

(2) Method.

$Y = e^{B \ln A}$

i. Floating Point Tape Check (FPT). This subroutine is executed whenever underflow or overflow is detected in either the AC or the MQ registers.

(1) Use. The subroutine is used with the following calling sequence.

CALL (FPT)

Return is made by use of a relatively addressed transfer instruction.

(2) Method. (FPT) examines the decrement part of location 00000 bits 14-17. These bits contain the code for the type of spill detected.

Operation	AC	MQ	Decrement 14	15	16	17	Octal Code
Add, Subtract		underflow	0	0	0	1	01
Multiply	underflow	underflow	0	0	1	1	03
Round	overflow		0	1	1	0	06
	overflow	overflow	0	1	1	1	07
Divide		underflow	1	0	0	1	11
	underflow		1	0	1	0	12
	underflow	underflow	1	0	1	1	13
		overflow	1	1	0	1	15

j. Backspace Tape. This subroutine backspaces the designated tape one physical record if BCD, or one logical record if binary.

(1) Use. This subroutine is used as a library tape subroutine.

with FORTRAN:

BACKSPACE i   i designates the tape unit

with FAP:

CAL I   Place unit designation in AC

TSX (BST), 4   Backspace tape

k. <u>End File Tape</u>. An end-of-file mark is written on the designated tape.

(1) Use. This subroutine is used as a library tape subroutine.

with FORTRAN:

END FILE i   i designates the tape unit

with FAP:

CAL I   Place unit designation in AC

TSX (EFT), 4   End file tape

l. <u>Rewind Tape</u>. The designated tape is rewound.

(1) Use. This subroutine is used as a library tape subroutine.

with FORTRAN:

REWIND i   i designates the tape unit

with FAP:

CAL I   Place unit designation in AC

TSX (RWT), 4   Rewind tape

m. <u>Tape Read/Write Simulation</u>. This subroutine initializes the tape unit and performs tape operation.

(1) Use. This subroutine is used as a library tape

subroutine to simulate the following:

(IØS)		Initializes select instruction addresses
(RDS)	RTDA	Reads tape decimal
(WRS)	WTDA	Writes tape decimal
(BSR)	BSRA	Backspaces tape one record
(WEF)	WEFA	Writes an end-of-file mark
(REW)	REWA	Rewinds tape
(ETT)	ETTA	End of tape test
(RCH)	RCHA	Reset and load
(TEF)	TEFA	Transfer on DSCA End-of-file
(TCØ)	TCØA	Transfer on DSCA in Operation
(TRC)	TRCA	Transfer on DSCA Redundancy Check

n. Input/Output Units (IØU). This subroutine saves total number of tapes for comparison check, and allocates storage locations for the tape units by use of the Boolean pseudo operation BØØL.

where A1 BØØL 1201 is the octal location of tape unit A1.

o. Write Tape Control (TES). This subroutine controls writing of a tape error for (STH) and (STB).

(1) Use. This subroutine is used as library tape subroutine. To initialize a tape write operation when used with (STH) and (STB), the instruction at (TES) is

TSX (WER), 4

(TES) is set to NØP by (WER) after checking last write.

(TES) NØP