LEONARD HEE
NASA AMES RESEARCH CENTER
MS 213-2
MOFFETT FIELD, CA 94035

USER'S GUIDE

TO SUBROUTINES

DATED: 8 DECEMBER 1981

TABLE OF CONTENTS

SECTION I - MATHEMATICAL SUBROUTINES/FUNCTIONS

NOTES: 1 - REAL-TIME OR NON-REAL-TIME
2 - REAL-TIME ONLY
3 - NON-REAL-TIME ONLY
4 - FORTRAN CALLABLE

SECTION II - CONVERSION SUBROUTINES/FUNCTIONS:

NOTES: 1 - Real-Time or Non-Real-Time
2 - Real-Time only
3 - Non-Real-Time only
4 - Fortran Callable

SECTION III - SPECIAL SUBROUTINES

NOTE: 1 - Real-Time or Non-Real-Time
2 - Real-Time only
3 - Non-Real-Time only
4 - Fortran calable

NAME: AEPC (End Point Check)

TYPE: Function

DESCRIPTION: End Point Checks Angle such that Angle is Restricted to ± 180 Degrees (Floating Point)

CALLING CONVENTIONS

FORTRAN: THETA = AEPC (THETA)

ASSEMBLY:
```
LEA  1,THETA
BL   AEPC
```

Result returned in register 7

TIME: 4.2 μsec/excess revolution

ACCURACY: N/A

REGISTERS DESTROYED: Ø, 7

RESTRICTIONS:

NAME: ALIM (Upper/Lower Limit)

TYPE: FUNCTION

DESCRIPTION: Limits Single Precision Floating Point to +/- Caller Specified Value

CALLING CONVENTIONS

FORTRAN: $Y = ALIM\ (X, X_{limit})$

ASSEMBLY:

```
BL      ALIM
DATAW   8
DATAW   W(X)
DATAW   -= E'X_lim' [or W(X_lim)]
```

Result returned in register 7

TIME: 11.4 μsec (avg)

ACCURACY: N/A

REGISTERS DESTROYED: Ø, 1, 5, 7

RESTRICTIONS: X_{limit} must be > 0

NAME: ALIML (Lower Limit)

TYPE: Function

DESCRIPTION: Lower Limits Single Precision FLoating Point Value to Caller Specified Value

CALLING CONVENTIONS

FORTRAN: $Y = ALIML\ (X, X_{min})$

ASSEMBLY:

```
BL      ALIML
DATAW   8
DATAW   W(X)
DATAW   = E'Xmin'      [or W(Xmin)]
```

Result returned to register 7

TIME: 5.7 μsec (avg)

ACCURACY: N/A

REGISTERS DESTROYED: Ø, 1, 6, 7

RESTRICTIONS:

NAME: ALIMU (Upper Limits)

TYPE: Function

DESCRIPTION: Upper Limits Single Precision Floating Point to Caller Specified Value

CALLING CONVENTIONS

FORTRAN: $Y = ALIMU\ (X, X_{max})$

ASSEMBLY:

```
BL      ALIMU
DATAW   8
DATAW   W(X)
DATAW   = E 'Xmax'      [or W(Xmax)]
```

Result returned in register 7

TIME: 5.7μsec (avg)

ACCURACY: N/A

REGISTERS DESTROYED: Ø, 1, 6, 7

RESTRICTIONS:

NAME: ARCTAN

TYPE: Function

DESCRIPTION: Computes ARCTANGENT from two orthogonal inputs (S.P. floating point)

CALLING CONVENTIONS

FORTRAN: THETA = ARCTAN (y,x)

ASSEMBLY:

```
BL      ARCTAN
DATAW   8
DATAW   W(y)
DATAW   W(x)
```

Result returned to register 7

where: x = K*SINE(Theta)

y = K*COSINE(Theta)

TIME: 65 μsec

ACCURACY: 1.7 ARCSEC

REGISTERS DESTROYED: 0,1,2,4,5,6 and 7

RESTRICTIONS:

NAME: ASQRT (Square Root)

TYPE: Function

DESCRIPTION: Computes the Square Root of a Single Precision Floating Point Value

CALLING CONVENTIONS

FORTRAN: Y = ASQRT (X)

ASSEMBLY:
```
LEA   1,X
BL    ASQRT
```

Result returned in register 7

TIME: 34.8 µsec

ACCURACY: 0.0003 %

REGISTERS DESTROYED: 0, 1, 3, 4, 5, 6 and 7

RESTRICTIONS: Negative Value Gives Zero result

NAME: ATAN5 (ARCTANGENT)

TYPE: Function

DESCRIPTION: Computes ARCTANGENT from Two Orthogonal Inputs (S.P. Floating Point) - Note: Higher Accuracy than ARCTAN

CALLING CONVENTIONS

FORTRAN: THETA = ATAN5 (Y,X)

ASSEMBLY:

```
BL      ATAN5
DATAW   8
DATAW   W(Y)
DATAW   W(X)
```

Result returned in register 7

where: X = K*SINE(THETA)
Y = K*COSINE(THETA)

TIME: 98.89 μsec

ACCURACY: 2.3 ARCSEC

REGISTERS DESTROYED: 0, 1, 2, 4, 5, 6 and 7

RESTRICTIONS:

NAME: FGAUSS (Random Number - GAUSSIAN)

TYPE: Function

DESCRIPTION: Generates a S.P. Floating Point Random Number in Range of -3.0 to +3.0

CALLING CONVENTIONS

FORTRAN: X = FGAUSS (Y)

ASSEMBLY: BL FGAUSS

On Return: R7 = GAUSSIAN Random Number

where: Y = Dummy Argument (Not used but required)

X = GAUSSIAN Random Number

TIME: 55.1 μsec

ACCURACY:

REGISTERS DESTROYED: Ø, 6, 7

RESTRICTIONS: Seed is internal to subroutine, cannot be passed.

NAME: RADNO (Random Number - Uniform)

TYPE: Function

DESCRIPTION: Provides a uniform Distribution Random Number Based on Previous Random Number (Seed). Range 0 → .99999

CALLING CONVENTIONS

FORTRAN: CALL RADNO

ASSEMBLY: BL RADNO

On Return: ZARDMB = Fixed Point Random Number (Range 0 to 1)

ZBRDMB = Floating Point Random Number (Range 0 to 1)

TIME: 47 μsec

ACCURACY:

REGISTERS DESTROYED: Ø, 4, 5

RESTRICTIONS: Unless specific requirements, users should not call random number, but use ZARDMB and ZBRDMB

NAME: SINCOD (SINE/COSINE)

TYPE: Subroutine

DESCRIPTION: Computes Sine and Cosine of Floating Point ANgle Given in Degrees (+/-180)

CALLING CONVENTIONS

FORTRAN: CALL SINCOD (THETA, $SINE_{(1)}$)

ASSEMBLY:

```
BL      SINCOD
DATAW   8
DATAW   W(THETA)
DATAW   W(SINE)
```

TIME: 99.3 μsec

ACCURACY: 0.1 ARC SEC

REGISTERS DESTROYED: 0, 1, 2, 4, 5, 6 and 7

RESTRICTIONS: Sine must be two word array

NAME: SINCOR (Sine/Cosine)

TYPE: Subroutine

DESCRIPTION: Computes Sine and Cosine of Floating Point Angle Given in Radians

CALLING CONVENTIONS

FORTRAN: CALL SINCOR (THETA, SINE $_{(1)}$)

ASSEMBLY:

```
BL      SINCOR
DATAW   8
DATAW   W(THETA)
DATAW   W(SINE)
```

TIME: 107 μsec

ACCURACY: 0.1 ARC SEC

REGISTERS DESTROYED: 0, 1, 2, 4, 5, 6 and 7

RESTRICTIONS: SINE must be two word array

NAME: TRIMAT (Matrix Multiply)

TYPE: Subroutine

DESCRIPTION: (3X3) Matrix by (3X3) Matrix Multiply (S.P. Floating Point)

CALLING CONVENTIONS

FORTRAN: CALL TRIMAT (PRI, SEC, PROD)

ASSEMBLY:

```
BL      TRIMAT
DATAW   12
DATAW   W(PRI)
DATAW   W(SEC)
DATAW   W(PROD)
```

where: PRI = Primary Matrix
SEC = Secondary Matrix
PROD = Product Matrix

TIME: 548.8 μsec

ACCURACY:

REGISTERS DESTROYED: 0

RESTRICTIONS:

NAME: AFIXFL (Fix-To-Float)

TYPE: Function

DESCRIPTION: Converts Fixed Pointed Number to S.P. Floating Point

CALLING CONVENTIONS

FORTRAN: A = AFIXFL (INPUT, SCALE)

ASSEMBLY:

```
BL      AFIXFL
DATAW   8
DATAW   W(INPUT)
DATAW   W(SCALE)
```

Result returned in register 7

TIME: 36 μsec

ACCURACY: 31 Bits

REGISTERS DESTROYED: Ø, 3, 4, 5, 6, 7

RESTRICTIONS: $-31 \leq \text{scale} \leq 31$

NAME: ASCDEC (ASCII-TO-DECIMAL)

TYPE: Subroutine

DESCRIPTION: Converts an ASCII String Representing a Decimal Value to a Fixed Point Decimal Word

CALLING CONVENTION

FORTRAN: CALL ASCDEC (INBUF, N, SCALE, OUT, ERFLAG)

ASSEMBLY:

```
BL      ASCDEC
DATAW   20
DATAW   B (INBUF)
DATAW   W(N)
DATAW   W(SCALE)
DATAW   W(OUT)
DATAW   B(ERFLAG)
```

where:
- INBUF = ASCII Decimal Input Buffer
- N = Number of Characters
- SCALE = Scale Factor of Fixed Point Decimal Output
- OUT = Fixed Point Decimal Value Result
- ERFLAG= Input Error Flag

TIME: 134 (3 Char); 182 (5 Char); 336 (9 Char)

ACCURACY:

REGISTERS DESTROYED: 0, 1, 2, 3, 4, 5, 6 and 7

RESTRICTIONS: $10^{-9} < \text{Dec Value} < 10^{8}$

NAME: ASCFLO (ASCII-TO-FLOAT)

TYPE: SUBROUTINE

DESCRIPTION: Converts a decimal number expressed in 8 bit ASCII to S.P. floating point.

CALLING CONVENTIONS

FORTRAN: CALL ASCFLO (INPUT, OUTPUT)

ASSEMBLY:

```
BL      ASCFLO
DATAW   8
DATAW   B(INPUT)
DATAW   B(INPUT)
DATAW   W(OUTPUT)
```

where: INPUT = Buffer containing ASCII string of the form (±) XX---XX.XX---XX Sign is optional.

TIME: 75 μsec + 12.9 μsec/Digit

ACCURACY:

REGISTERS DESTROYED: 0

RESTRICTIONS: Number <4,294,967,295; Max # of Fractional Digits = 8

NAME: ASCHEX (ASCII-to-HEX)

TYPE: Subroutine

DESCRIPTION: Converts an ASCII String Representing a Hex Value to a Hex Word Value

CALLING CONVENTIONS

FORTRAN: CALL ASCHEX (IBUF, N, IOUT)

ASSEMBLY:

```
BL      ASCHEX
DATAW   B(IBUF_
DATAW   W(N)
DATAW   W(IOUT)
```

where: IBUF = Hex Input Buffer

N = Number of characters

IOUT = Hex Output Word

TIME: 129.6 μsec (eight characters)

ACCURACY:

REGISTERS DESTROYED: 0, 1, 2, 3, 4 and 5

RESTRICTIONS: $1 \leq N \leq 8$ (No error checking)

NAME: ASCHMS (ASCII-to-Hrs/Min/Sec)

TYPE: Subroutine

DESCRIPTION: Converts Hrs/Mins/Sec (ASCII Format HH:MM:SS) to Seconds (S.P. Floating Point)

CALLING CONVENTIONS

FORTRAN: CALL ASCHMS (FMT, IA, LE, N)

ASSEMBLY:

```
BL      ASCHMS
DATAW   16
DATAW   W(FMT)
DATAW   B(IA)
DATAW   B(LE)
DATAW   W(N)
```

where: FMT = Results in Seconds

IA = HRS/MINS/SEC Input Buffer

LE = Input Error Flag

N = Number of characters

TIME: 192 μsec

ACCURACY:

REGISTERS DESTROYED: 0, 1, 2, 3, 4, 5, 6 and 7

RESTRICTIONS: $2 \leq N \leq 8$; Max time = 23 hrs, 59 min, 59 sec

NAME: ASCINT (ASCII-TO-INTEGER)

TYPE: Subroutine

DESCRIPTION: Converts and integer, expressed in 8-bit ASCII to a S.P. integer.

CALLING CONVENTIONS

FORTRAN: CALL ASCINT (INPUT, OUTPUT)

ASSEMBLY:

```
BL      ASCINT
DATAW   8
DATAW   B(INPUT)
DATAW   W(OUTPUT)
```

where: INPUT = Buffer containing ASCII string of the Form (± XX --- XX. Sign is optional.

TIME: 35.8 μsec + 12.3μsec/Digit

ACCURACY:

REGISTERS DESTROYED: 0

RESTRICTIONS: Number <4,294,967,295

NAME: ASCLAT (ASCII-to-Latitude)

TYPE: Subroutine

DESCRIPTION: Converts Latitude (in ASCII Format $^{N}_{S}$XX.XX.XX.X) to S.P. Floating Point Degrees

CALLING CONVENTIONS

FORTRAN: CALL ASCLAT (FL, IA, LE, N)

ASSEMBLY:

```
BL      ASCLAT
DATAW   16
DATAW   W(FL)
DATAW   B(IA)
DATAW   B(:LE)
DATAW   W(N)
```

where: FL = Results in Degrees Latitude (S = Neg)

IA = ASCII latitude input buffer address

LE = Input error flag

N = Number of input characters

TIME: 326 μsec (12 characters)

ACCURACY:

REGISTERS DESTROYED: 0, 1, 2, 3, 4, 5, 6 and 7

RESTRICTIONS: $1 \leq N \leq 12$; Latitude $\leq 180°$ (N or S)

NAME: ASCLON (ASCII-to-Longitude)

TYPE: Subroutine

DESCRIPTION: Converts Longitude (in ASCII Format $^{W}_{E}$XXX.XX.XX.X) to S.P. Floating Point Degrees

CALLING CONVENTIONS

FORTRAN: CALL ASCLON (FL, IA, LE, N)

ASSEMBLY:

```
BL      ASCLON
DATAW   16
DATAW   W(FL)
DATAW   B(IA)
DATAW   B(LE)
DATAW   W(N)
```

where: FL = Result in Degrees Longitude (W = neg)

IA = ASCII Longitude input buffer address

LE = Input error flag

N = Number of Input characters

TIME: 336 μsec (12 characters)

ACCURACY:

REGISTERS DESTROYED: 0, 1, 2, 3, 4, 5, 6 and 7

RESTRICTIONS: $1 \leq N \leq 12$; Longitude $\leq 180°$ (E or W)

NAME: BCDFLO (BCD-to-Float)

TYPE: Subroutine

DESCRIPTION: Converts BCD to S.P. Floating Point

CALLING CONVENTIONS

FORTRAN: CALL BCDFLO (INPUT, SAVER, OUTPUT)

```
ASSEMBLY:   BL      BCDFLO
            DATAW   12
            DATAW   B(INPUT)
            DATAW   W(SAVER)
            DATAW   W(OUTPUT)
```

where: Saver = Input Parameters

Byte 1 = Total Field Length

Byte 2 = Fractional Field Length

Byte 3 = Starting Nibble in First Byte of Input Buffer (0 or 1). If a 1, nibble 0 - Bit 4 = sign bit

TIME: 45 μsec + 18 μsec/digit

ACCURACY:

REGISTERS DESTROYED: 0

RESTRICTIONS: Max field length = 8;

NAME: BCDIN (BCD-to-Integer)

TYPE: Subroutine

DESCRIPTION: Converts BCD to S.P. Integer Value

CALLING CONVENTIONS

FORTRAN: CALL BCDIN (IBCD, IBIT, N, INT)

ASSEMBLY:

```
BL      BCDIN
DATAW   16
DATAW   H(IBCD)
DATAW   W(IBIT)
DATAW   W(N)
DATAW   W(INT)
```

where: IBCD = Packed BCD Input

IBIT = Starting BIT Position of First BCD Digit

N = Number of BCD Digits (4 max)

INT = Integer Results

TIME: 16.8 μsec + 9.8 μsec/digit (Bit 0 Start)

ACCURACY:

REGISTERS DESTROYED: 0, 1, 3, 4, 5, 6 and 7

RESTRICTIONS: $0 \leq N \leq 4$

NAME: DECASC (Decimal-to-ASCII)

TYPE: Subroutine

DESCRIPTION: Converts a Fixed Point Decimal Word to a Character String Representing the Decimal Value

CALLING CONVENTIONS

FORTRAN: CALL DECASC (INPUT, SCALE, N, OUTBUF, INUM)

ASSEMBLY:

```
BL      DECASC
DATAW   20
DATAW   INPUT
DATAW   SCALE
DATAW   N
DATAW   OUTBUF
DATAW   INUM
```

where:
- INPUT = Decimal Word to be Converted
- SCALE = Scale Factor of Fixed Point Decimal Input
- N = Total Number of Characters
- OUTBUF = ASCII Decimal Output Buffer
- INUM = Number of Characters Left of Decimal Point

TIME: 288 μsec (15 characters)

ACCURACY:

REGISTERS DESTROYED: 0, 1, 2, 3, 4, 5, 6 and 7

RESTRICTIONS: $10^{-9} <$ Dec Value $< 10^{8}$

NAME: FLOASC (Float-to-ASCII)

TYPE: Subroutine

DESCRIPTION: Converts a Positive S.P. Floating Point Value to an ASCII String Representing the Decimal Value

CALLING CONVENTIONS

FORTRAN: CALL FLOASC (INPUT, SAVER, OUTPUT)

ASSEMBLY:

```
BL      FLOASC
DATAW   12
DATAW   W(INPUT)
DATAW   W(SAVER)
DATAW   B(OUTPUT)
```

where: SAVER = Output Parameters:

Byte 1 = Total Field Length

Byte 2 = Fractional Field Length

TIME: 66 μsec + 12 μsec/digit

ACCURACY:

REGISTERS DESTROYED: 0

RESTRICTIONS: Max Field Length = 8; $0 \leq$ Input $\leq$ 99,999,999.

NAME: FLOBCD (Float-to-BCD)

TYPE: Subroutine

DESCRIPTION: Converts S.P. Floating Point Value to Binary Coded Decimal

CALLING CONVENTIONS

FORTRAN: CALL FLOBCD (INPUT, SAVER, OUTPUT)

ASSEMBLY:

```
BL      FLOBCD
DATAW   12
DATAW   W(INPUT)
DATAW   W(SAVER)
DATAW   B(OUTPUT)
```

where: SAVER = Output Parameter:

Byte 1 = Total Field Length

Byte 2 = Fractional Field Length

Byte 3 = Starting Nibble in First byte of output buffer (0 or 1)

TIME: 58 μsec + 12.4 μsec/digit

ACCURACY:

REGISTERS DESTROYED: 0

RESTRICTIONS: Max Field Length = 8; $0 \leq$ Input $\leq$ 99,999,999

NAME: HEXASC (Hex-to-ASCII)

TYPE: Subroutine

DESCRIPTION: Converts a Hex Integer to an ASCII String Representing the Hex Value

CALLING CONVENTIONS

FORTRAN: CALL HEXASC (IBUF, N, INP)

ASSEMBLY:

```
BL      HEXASC
DATAW   12
DATAW   B(IBUF)
DATAW   W(N)
DATAW   W(INP)
```

where: IBUF = Hex Output Buffer

N = Number of Characters

INP = Hex Input Word

TIME: 96 μsec (8 characters)

ACCURACY:

REGISTERS DESTROYED: 0, 1, 2, 3, 4, 5 and 7

RESTRICTIONS: $1 \leq N \leq 8$

NAME: HMSASC (Hrs/Min/Sec-to-ASCII)

TYPE: Subroutine

DESCRIPTION: Converts seconds (S.P. Floating Point) to ASCII (Format HH:MM:SS)

CALLING CONVENTIONS

FORTRAN: CALL HMSASC (FMT, IA, N)

ASSEMBLY:

```
BL      HMSASC
DATAW   12
DATAW   W(FMT)
DATAW   B(IA)
DATAW   W(N)
```

where: FMT = Input Seconds

IA = HRS/MINS/SEC Output Buffer

N = Number of Characters

TIME: 134.4 μsec

ACCURACY:

REGISTERS DESTROYED: 0, 1, 2, 3, 4, 5, 6 and 7

RESTRICTIONS: $2 \leq N \leq 8$; Max time = 86399 seconds

NAME: IFLOFIX (Float-to-Fix)

TYPE: Function

DESCRIPTION: Convert a S.P. Floating Point Value to Fixed Point Value

CALLING CONVENTIONS

FORTRAN: A = IFLOFIX (INPUT, SCALE)

ASSEMBLY:

```
BL      IFLOFIX
DATAW   8
DATAW   W(INPUT)
DATAW   W(SCALE)
```

Result returned to register 7

TIME: 50 μsec

ACCURACY: 31 Bits

REGISTERS DESTROYED: Ø, 3, 5, 6, 7

RESTRICTIONS: $-31 \leq \text{scale} \leq 31$

NAME: INTASC (Integer-to-ASCII)

TYPE: Subroutine

DESCRIPTION: Convert a S.P. Integer Value to an ASCII String Representing the Integer Balue

CALLING CONVENTIONS

FORTRAN: CALL INTASC (INPUT, SAVER, OUTPUT)

ASSEMBLY:

```
BL      INTASC
DATAW   12
DATAW   W(INPUT)
DATAW   W(SAVER)
DATAW   B(OUTPUT)
```

where: Saver = Output Parameter:

Byte 1 = Total Field Length

TIME: 42.6 μsec + 13.8 μsec/Digit

ACCURACY:

REGISTERS DESTROYED: 0

RESTRICTIONS: Max Field Length = 8; $0 \leq$ Input $\leq$ 99,999,999

NAME: INTBCD (Integer-to-BCD)

TYPE: Subroutine

DESCRIPTION: Convert a Positive Integer Value to Binary Coded Decimal

CALLING CONVENTIONS

FORTRAN: CALL INTBCD (INPUT, SAVER, OUTPUT)

ASSEMBLY:

```
BL      INTBCD
DATAW   12
DATAW   W(INPUT)
DATAW   W(SAVER)
DATAW   B(OUTPUT)
```

where: SAVER = Output Parameters:

Byte 1 = Total Field Length

Byte 2 = Starting Nibble in First Byte of Output Buffer (0 or 1)

TIME: 44 μsec + 13.7 sec/Digit

ACCURACY:

REGISTERS DESTROYED: 0

RESTRICTIONS: Max Field Length = 8; $0 \leq \text{Input} \leq (10^9 - 1)$

NAME: LATASC (Latitude-to-ASCII)

TYPE: Subroutine

DESCRIPTION: Converts to a S.P. Floating Point Angle (Degrees) to an ASCII String of Format $\begin{matrix} N \\ S \end{matrix}$ XXX°XX'XX.X"

CALLING CONVENTIONS

FORTRAN: CALL LATASC (FL, IA, N)

ASSEMBLY:

```
BL      LATASC
DATAW   12
DATAW   W(FL)
DATAW   B(IA)
DATAW   W(N)
```

where: FL = Input Angle in Degrees

IA = ASCII Latitude Output Buffer

N = Number of Output Characters

TIME: 182 sec (13 characters)

ACCURACY:

REGISTERS DESTROYED: 0, 1, 2, 3, 4, 5, 6 and 7

RESTRICTIONS: $1 \leq N \leq 13$; FL = 90°

NAME: LONASC (Longitude-to-ASCII)

TYPE: Subroutine

DESCRIPTION Converts a S.P. Floating Point Angle (Degrees) to an ASCII String of Format $\begin{matrix}E\\W\end{matrix}$ XXX°XX'XX.X"

CALLING CONVENTIONS

FORTRAN: CALL LONASC (FL, IA, N)

ASSEMBLY:

```
BL      LONASC
DATAW   12
DATAW   W(FL)
DATAW   B(IA)
DATAW   W(N)
```

where: FL = Input Angle in Degrees

IA = ASCII Longitude Output Buffer

N = Number of Output Characters

TIME: 182 μsec (13 characters)

ACCURACY:

REGISTERS DESTROYED: 0, 1, 2, 3, 4, 5, 6 and 7

RESTRICTIONS: $1 \leq N \leq 13$; FL $\leq 180°$

NAME: BITFR (Bit Function Read)

TYPE: Subroutine

DESCRIPTION: Allows the Caller to Mask (Reset) Bits in a Variable or Shift a Variable or a Combination of the Above

CALLING CONVENTIONS

FORTRAN: CALL BITFR (VAR1, MASK, SC, VAR2)

ASSEMBLY: None; Use SEL Assembler Instructions

where: VAR1 = Input Bit Configuration

MASK = Mask Configuration

SC = Shift Count (+ = Left, - = Right)

VAR2 = Mask/Shifted Result

Execution Sequence: VAR1 masked with MASK, Shifted by SC and stored in VAR2

TIME: 34 μsec

ACCURACY:

REGISTERS DESTROYED: 0

RESTRICTIONS: $-31 \leq SC \leq 31$

NAME: BITFS (Bit Function Set)

TYPE: Subroutine

DESCRIPTION: Allows the Caller to Shift a Variable, Mask and Set Multiple Bits in a Variable, or a Combination of the Above

CALLING CONVENTIONS

FORTRAN: CALL BITFS (VAR1, MASK, SC, VAR2)

ASSEMBLY: None; Use SEL Assembler Instructions

where:
- VAR1 = Input Bit Configuration
- MASK = Mask Configuration (For VAR1 and VAR2)
- SC = Shift Count (= Left, - = Right)
- VAR2 = Contains Bit Set Configuration on Entry; on Exit Contains Result

Execution Sequence:
1) VAR1 shifted by SC then masked with MASK
2) VAR2 masked with one's complement of MASK
3) Step 1 and 2 results 'ored' and store in VAR2

TIME: 38 μsec

ACCURACY:

REGISTERS DESTROYED: 0

RESTRICTIONS: $-31 \leq SC \leq 31$

NAME: BITS (Bit Set)

TYPE: Subroutine

DESCRIPTION: Allows Setting/Resetting of a Specified Bit in a Byte/Halfword/Word as a Function of a Logical Variable (Set if not equal to 0)

CALLING CONVENTIONS

FORTRAN: CALL BITS (VAR1, K, VAR2)

ASSEMBLY: None; Use SEL Assembler Bit Instructions

where: VAR1 = BYTE, HALFWORD, Word in which BIT is to be Set/Reset

K = Bit Number to be Set/Reset

VAR = Defines Setting (VAR2 if not equal to 0) of Bit in VAR1

TIME: 35 μsec

ACCURACY:

REGISTERS DESTROYED 0

RESTRICTIONS: K(Bit #) Must not exceed precision of VAR1

NAME: BITT (Bit Test)

TYPE: Subroutine

DESCRIPTION: Returns a Logical Byte Whose State Reflects the State of a Bit within a User SPecified BYTE/HALFWORD/WORD

CALLING CONVENTIONS

FORTRAN: CALL BITT (VAR1, K, VAR2)

ASSEMBLY: None; use SEL Bit Instructions

where: VAR1 = Byte, Halfword, Word in which Bit is to be Tested

K = Bit Number to be Tested

VAR2 = Logical Byte to be set True (Bit = 1) or False (Bit = 0)

TIME: 35 μsec

ACCURACY:

REGISTERS DESTROYED: 0

RESTRICTIONS: K(Bit #) Must Not Exceed Precision of VAR1

NAME: ESRD (SYNCHRO/Resolver Drive)

TYPE: Subroutine

DESCRIPTION: Computes XZ and YZ Drives for Either SYNCHROS or Resolvers

CALLING CONVENTIONS

FORTRAN: CALL ESRD (ANGLE, XZ, YZ, ITYPE)

ASSEMBLY:

```
BL      ESRD
DATAW   16
DATAW   W(ANGLE)
DATAW   W(XZ)
DATAW   W(YZ)
DATAW   W(ITYPE)
```

where: XZ = SIN (ANGLE + 60) FOR SYNCHRO
= SIN (ANGLE) FOR RESOLVER
YZ = SIN (ANGLE + 120) FOR SYNCHRO
= COS (ANGLE) FOR RESOLVER
ITYPE = 1 FOR SYNCHRO; 2 FOR RESOLVER

TIME: 186 μsec for SYNCHRO; 161 for Resolver

ACCURACY:

REGISTERS DESTROYED: Ø, 1, 6, 7

RESTRICTIONS: $-180° \leq \text{Angle} \leq 180°$

NAME: GENEVA (Geneva Drive)

TYPE: Subroutine

DESCRIPTION: Converts S.P. Floating Point Value to Sine/Cosine of 36*(100 Digit), 36*(10 Digit) and 36*(Unit Digit)

CALLING CONVENTIONS

FORTRAN: CALL GENEVA (ARG1, ARG2, ARG3, ARG4, ARG5, ARG6, ARG7)

ASSEMBLY:

```
BL      GENEVA
DATAW   32
DATAW   W(ARG1)
DATAW   W(ARG2)
DATAW   W(ARG3)
DATAW   W(ARG4)
DATAW   W(ARG5)
DATAW   W(ARG6)
DATAW   W(ARG7)
```

where: ARG1 = Input Value

ARG2,3 = Sin/Cos of Hundreds

ARG4,5 = Sin/Cos of Tens

ARG6,7 = Sin/Cos of Units

TIME: 484.69 μsec

ACCURACY:

REGISTERS DESTROYED: 0, 1, 2, 3, 4, 5, 6 and 7

RESTRICTIONS: $0 \leq ARG1 \leq 1000$

NAME: SDATE (System Date)

TYPE: Subroutine

DESCRIPTION: Assembly Language Callable Routine which returns RTM System Month-Day-Year

CALLING CONVENTIONS

FORTRAN: None

ASSEMBLY: BL SDATE

On Exit: R6/R7 = MM/DD/YY in ASCII

TIME: 7 μsec

ACCURACY:

REGISTERS DESTROYED: Ø, 6, 7

RESTRICTIONS: CPU 1 Only

NAME: SPOOLFIL (Spool File Create)

TYPE: Subroutine

DESCRIPTION: Creates a file of the form JCL.SØxx where xx = 0-40 for use in Dynamic Batch Job files. Part of the created JCL should be to delete the file

CALLING CONVENTIONS

FORTRAN: CALL SPOOLFIL (FILENAME, SIZE, ERRTN)

ASSEMBLY:

```
BL      SPOOLFIL
DATAW   12
DATAW   D(FILENAME)
DATAW   W(SIZE)
DATAW   W(ERRTN)
```

where: FILENAME = DOUBLEWORD VARIABLE WHERE ACTUAL FILENAME WILL BE STORED.

SIZE = SIZE OF FILE TO BE CREATED IN BLOCKS

ERRTN = ADDRESS (OR LABEL) TO RETURN IF FILE COULD NOT BE CREATED

TIME: N/A

ACCURACY: N/A

REGISTERS DESTROYED: ALL

RESTRICTIONS: NON-REAL TIME ONLY

NAME: SYMLK (Symbol Look-Up NOWAIT I/O)

TYPE: Subroutine

DESCRIPTION: Assembly Language Callable "No Wait I/O" Symbol Look-Up Routine

CALLING CONVENTIONS

FORTRAN: None

ASSEMBLY: On Entry: R6/R7 = ASCII Symbol Name

R3 = 24 Word User Buffer

BL SYMLK

On Exit: If Cond Code 1 = 1 Request was denied. If not, user must monitor word 24 of buffer:
-1 = Search in Progress,
0 = Not Found, Other = Symbol Absolute Address. Words 1-23 of buffer as described in RTM Ref. Manual, Figure 16-1.

→ word numb... should be ... for:
-1 =
0 =
OTHER =
WORDS 1-23 —

TIME: Variable

ACCURACY:

REGISTERS DESTROYED: 0, 1, 2, 3, 4, 5, 6 and 7

RESTRICTIONS: To be used in Real-Time Only. Clear with Computer Systems Lead Personnel before using.

NAME: SYMLKW (Symbol Look-Up Wait I/O)

TYPE: Subroutine

DESCRIPTION: Assembly Language Callable "WAIT I/O" Symbol Look-Up Routine

CALLING CONVENTIONS

FORTRAN: None

ASSEMBLY: On Entry: R6/R7 = ASCII Symbol Name

R3 = 24 Word User Buffer

```
BL      SYMLKW
```

On Exit: User Buffer Filled with Symbol Data as described in RTM Reference Manual, Figure 16-1, with the exception that Word 24 = Absolute Address of Symbol (If Found or 0 If Not Found)

TIME: N/A

ACCURACY:

REGISTERS DESTROYED: 0, 1, 3, 4, 5, 6, and 7

RESTRICTIONS: Non-Real Time Usage Only (CPU 1 Only)

NAME: ZTIME (System Time)

TYPE: Subroutine

DESCRIPTION: Assembly Language Callable Routine to Fetch RTM System Time

CALLING CONVENTIONS

FORTRAN: BL ZTIME

On Exit: R7 (BYTE 1) = Hours (0-23)

(BYTE 2) = Minutes (0-59)

(BYTE 3) = Seconds (0-59)

(BYTE 4) = Interrupts (0-60)

TIME: 16.7 μsec

ACCURACY:

REGISTERS DESTROYED: Ø, 7

RESTRICTIONS: CPU 1 Only