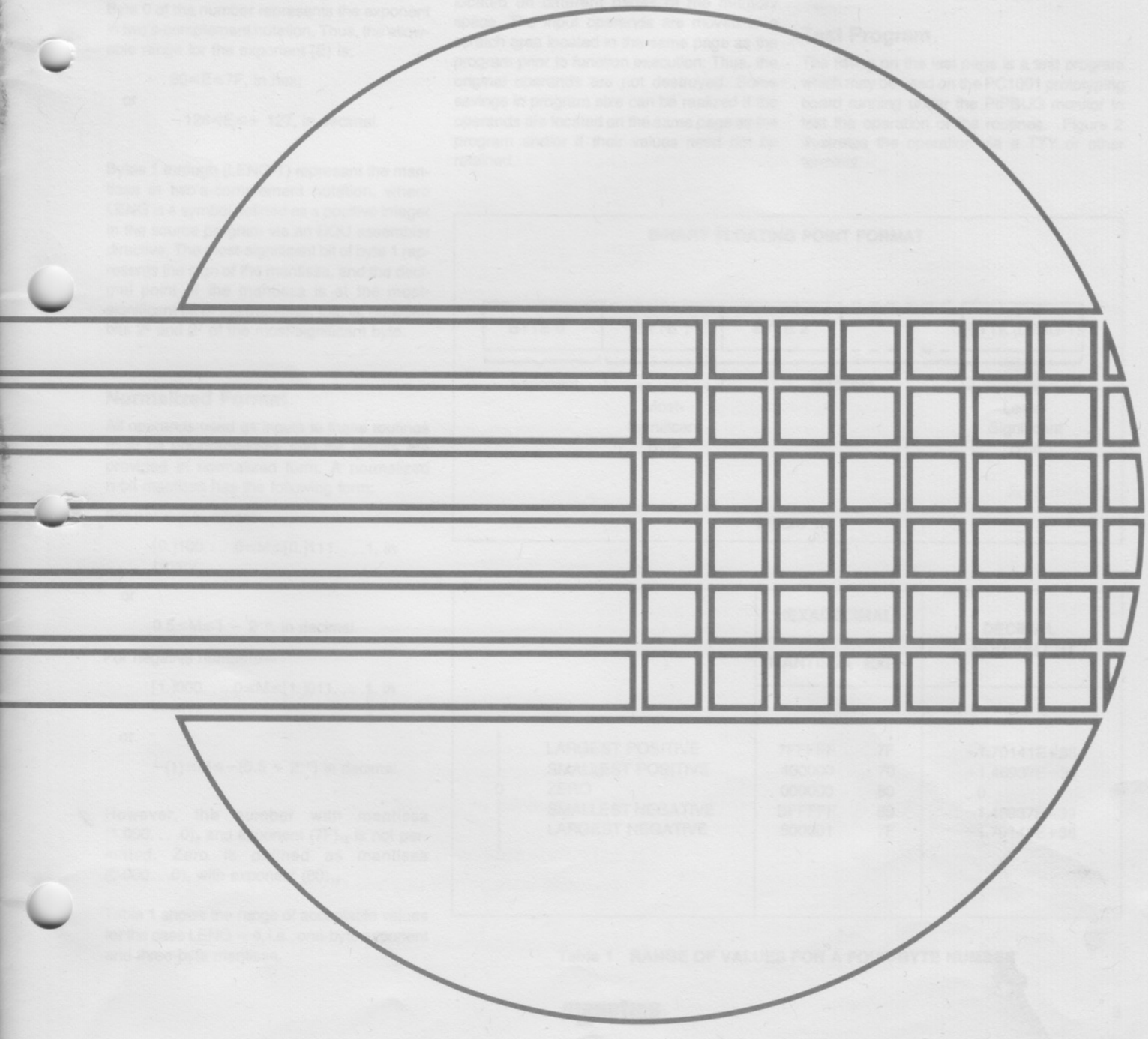
A 2650 APPLICATIONS MEMO

SIGNETICS 2650 BINARY FLOATING POINT ROUTINES AS57



INTRODUCTION

The capability of operating on binary floating point numbers is provided by the four routines presented in this applications memo. These routines perform the addition, subtraction, multiplication, and division of binary floating point operands in normalized two's complement notation. Multiple precision is facilitated through the use of a variable-length mantissa.

Data Format

The format for the operands used in these routines is shown in Figure 1.

Byte 0 of the number represents the exponent in two's-complement notation. Thus, the allowable range for the exponent (E) is:

$80 \leq E \leq 7F$, in hex;

or

$-128 \leq E \leq +127$, in decimal.

Bytes 1 through (LENG-1) represent the mantissa in two's-complement notation, where LENG is a symbol defined as a positive integer in the source program via an EQU assembler directive. The most-significant bit of byte 1 represents the sign of the mantissa, and the decimal point of the mantissa is at the most-significant bit of the mantissa, that is, between bits 2^6 and 2^7 of the most-significant byte.

Normalized Format

All operands used as inputs to these routines must be pre-normalized, and the results are provided in normalized form. A normalized n-bit mantissa has the following form:

For positive numbers—

$[0.]100\ldots0 \leq M \leq [0.]111\ldots1$, in binary;

or

$0.5 \leq M \leq 1 - 2^{-n}$, in decimal.

For negative numbers—

$[1.]000\ldots0 \leq M \leq [1.]011\ldots1$, in binary;

or

$-(1) \leq M \leq -(0.5 + 2^{-n})$ in decimal.

However, the number with mantissa $(1.000\ldots0)_2$ and exponent $(7F)_{16}$ is not permitted. Zero is defined as mantissa $(0.000\ldots0)_2$ with exponent $(80)_{16}$.

Table 1 shows the range of acceptable values for the case LENG = 4, i.e., one-byte exponent and three-byte mantissa.

OPERATIONAL DETAILS

The routines operate as follows:

(OPERAND1) # (OPERAND2) ⟶ RESULT,

where # is one of the four operators +, −, ×, or :.

Operands 1 and 2 are stored in memory starting at addresses OP1 and OP2 and, as mentioned previously, must be pre-normalized. The normalized result is situated in memory starting at location RSLT. This area need not be cleared prior to execution of the routine. The program area and the operands may be located on different pages of the memory space. The input operands are moved to a scratch area located in the same page as the program prior to function execution. Thus, the original operands are not destroyed. Some savings in program size can be realized if the operands are located on the same page as the program and/or if their values need not be retained.

Rounding of the result is controlled by the contents of the location ROUN:

(ROUN) = H'00' specifies no rounding,
(ROUN) = H'80' specifies roundup.

Various error conditions (overflow, underflow, etc.) result in jumps to different error locations to facilitate test and/or corrective actions.

The main program calls the routines by performing the following subroutine branches:

BSTA, UN	BADD for addition
BSTA, UN	BSUB for subtraction
BSTA, UN	BMUL for multiplication
BSTA, UN	BDIV for division

Test Program

The listing on the last page is a test program which may be used on the PC1001 prototyping board running under the PIPBUG monitor to test the operation of the routines. Figure 2 illustrates the operation via a TTY or other terminal.

BINARY FLOATING POINT FORMAT

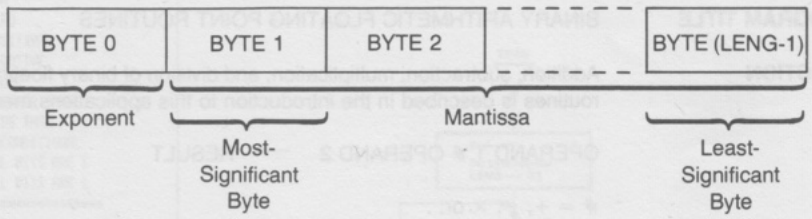


Figure 1

		HEXADECIMAL		DECIMAL EQUIVALENT
		MANTISSA	EXP.	
+	LARGEST POSITIVE	7FFFFF	7F	+1.70141E+38
	SMALLEST POSITIVE	400000	70	+1.46937E−39
0	ZERO	000000	80	0
	SMALLEST NEGATIVE	BFFFFF	80	−1.46937E−39
−	LARGEST NEGATIVE	800001	7F	−1.70141E+38

Table 1 RANGE OF VALUES FOR A FOUR-BYTE NUMBER

OPERATION OF TEST PROGRAM

```
*G 6AA

01456789 + FF456789 = 0156C16B  (NO ROUNDING)
45987654 + 46456789 = 44468ACC
54678934 + 56B87654 = 55A4B142

01456789 - FF456789 = 00681B4D
01546576 - 02765434 = 02B3DE87
80000000 - 65876543 = 65789ABD

*G 6A6

AA456789 * 23567894 = CC5DC5E3  (ROUNDING)
45768765 * 23876545 = 689051C8
FF87BFCD * 457BCDFE = 448BB0C9

AAAAAAAA : BBBBBBBB = F0500000
BBBBBBBB : AAAAAAAA = 11666667
F0500000 * 11666667 = 01400000
```

Figure 2

PROGRAM TITLE	BINARY ARITHMETIC FLOATING POINT ROUTINES
FUNCTION	Addition, subtraction, multiplication, and division of binary floating point numbers. The specification of numbers and routines is described in the introduction to this applications memo. OPERAND 1 # OPERAND 2 → RESULT # = +, −, × or: .
PARAMETERS **INPUT:**	ROUN must contain the rounding constant. Length of operands and result are defined by LENG. OPERAND 1 is in memory starting at address OP1. OPERAND 2 is in memory starting at address OP2.
OUTPUT:	RESULT is in memory starting at address RSLT.

Refer to Figures 3 through 9 for flowcharts and program listings.

HARDWARE AFFECTED

REGISTERS	**R0** X	**R1** X	**R2** X	**R3** X	**R1'**	**R2'**	**R3'**
PSU	**F**	**II**	**SP** X				
PSL	**CC** X	**IDC** X	**RS**	**WC** X	**OVF** X	**COM** X	**C** X

RAM REQUIRED (BYTES): 6 X LENG +5

ROM REQUIRED BYTES): 598

EXECUTION TIME: Variable

MAXIMUM SUBROUTINE NESTING LEVELS: 1

ASSEMBLER/COMPILER USED: TWIN VER 1.0

```
LINE ADDR  OBJECT  E SOURCE

0001                * P07601Ø0
0002                ************************************************
0003                *
0004                * BINARY FLOATING POINT ARITHMETIC PACKAGE.
0005                *
0006                * THIS PACKAGE CONSISTS OF: AN ADDITION ROUTINE,
0007                *                           A SUBTRACTION ROUTINE,
0008                *                           A MULTIPLICATION ROUTINE,
0009                *                           AND A DIVISION ROUTINE
0010                *                           FOR TWO BINARY FLOATING
0011                *                           POINT NUMBERS.
0012                *
0013                * THE FORMAT OF THE BINARY FLOATING POINT NUMBERS IS
0014                * AS FOLLOWS: BYTE Ø = EXPONENT IN TWO'S COMPLEMENT
0015                *             BYTE 1-->(LENG-1) = MANTISSA IN TWO'S
0016                *             COMPLEMENT OF WHICH BYTE 1 IS THE MOST
0017                *             SIGNIFICANT BYTE.
0018                *             THE POINT POSITION IS IN FRONT OF THE
0019                *             MANTISSA
0020                *
0021                ************************************************
0022                *
0023                * DEFINITIONS OF SYMBOLS:
0024                *
0025 0000          R0   EQU   0        PROCESSOR-REGISTERS
0026 0001          R1   EQU   1
0027 0002          R2   EQU   2
0028 0003          R3   EQU   3
0029 0080          S    EQU   H'80'    PSU:  SENSE
0030 0040          F    EQU   H'40'           FLAG
0031 0020          II   EQU   H'20'           INTERRUPT INHIBIT
0032 0007          SP   EQU   H'07'           STACKPOINTER
0033 00C0          CC   EQU   H'C0'    PSL:  CONDITION CODE
0034 0020          IDC  EQU   H'20'           INTERDIGIT CARRY
0035 0010          RS   EQU   H'10'           REGISTER BANK SELECT
0036 0008          WC   EQU   H'08'           1=WITH, Ø=WITHOUT CARRY
0037 0004          OVF  EQU   H'04'           OVERFLOW
0038 0002          COM  EQU   H'02'           1=LOGIC, Ø=ARITH COMPARE
0039 0001          C    EQU   H'01'           CARRY/BORROW
0040 0000          Z    EQU   0        BRANCH COND.:  ZERO
0041 0001          P    EQU   1                       POSITIVE
0042 0002          N    EQU   2                       NEGATIVE
0043 0000          EQ   EQU   0                       EQUAL
0044 0001          GT   EQU   1                       GREATER THAN
0045 0002          LT   EQU   2                       LESS THAN
0046 0003          UN   EQU   3                       UNCONDITIONAL
0047 0000          AL   EQU   0                       ALL BITS ARE 1
0048 0002          NO   EQU   2                     NOT ALL BITS ARE 1
0049                ************************************************
0050                *
0051                * DEFINITIONS OF PROGRAM DEFINED SYMBOLS
0052                *
0053 0004          LENG EQU   4        LENGTH OF OPERAND
0054 0004          LEN  EQU   LENG
0055 0008          LEN2 EQU   LEN+LEN
0056 0010          LEN4 EQU   LEN2+LEN2
0057 0020          LEN8 EQU   LEN4+LEN4
0058 0019          MLEN EQU   LEN8-7
0059 001F          DLEN EQU   LEN8-1
0060                *
0061                ************************************************
0062                *
0063                * SCRATCH-PAD AREA *
0064                *
0065 0000               ORG   H'780'
0066 0780          ROUN RES   1        ROUNDING CONSTANT
0067 0781          ADR  RES   2        INDIRECT ADDRESS
0068 0783          FLAG RES   1        FLAG
0069 0784          OPA  RES   LEN      OPERAND 1 SCRATCH-PAD AREA
0070 0788          SIGN RES   1        SIGN FLAG
0071 0789          OPB  RES   LEN2     OPERAND 2 SCRATCH-PAD AREA
0072                *
0073                ************************************************
0074                *
0075                * OPERANDS AREA *
0076                *
0077 0791               ORG   H'7C8'
0078 07C8          OP1  RES   LEN      OPERAND 1
0079 07C4          OP2  RES   LEN      OPERAND 2
0080 07C8          RSLT RES   LEN      RESULT
0081                *
0082                ************************************************
0083                *
0084                * PROGRAM AREA *
0085                *
0086 07CC               ORG   H'440'
0087 0440 07       PNT1 DATA  <OP1     INDIRECT ADDRESS OF OPERAND 1
0088 0441 C8            DATA  >OP1
0089 0442 07       PNT2 DATA  <OP2     INDIRECT ADDRESS OF OPERAND 2
0090 0443 C4            DATA  >OP2
0091 0444 07       PNTR DATA  <RSLT    INDIRECT ADDRESS OF RESULT
0092 0445 C8            DATA  >RSLT
0093                *
0094                ************************************************
0095                *
0096                * TRANSFER OPERANDS ROUTINE *
0097                *
0098                * THIS ROUTINE TRANSFERS THE OPERANDS TO THE
0099                * SCRATCH-PAD.
0100                *
0101 0446 7708     TRAN PPSL  WC       WITH CARRY
0102 0448 0704          LODI,R3 LEN    SET BYTE COUNTER
0103 044A BFC440   LPC  LODA,R0 *PNT1,R3,- DECR BYTE COUNTER AND TRANSFER
0104 044D CF6784        STRA,R0 OPA,R3  BYTE OF OPERAND1 TO SCRATCH-PAD
0105 0450 BFE442        LODA,R0 *PNT2,R3  TRANSFER BYTE OF OPERAND2
0106 0453 CF6789        STRA,R0 OPB,R3  TO SCRATCH-PAD
0107 0456 5B72          BRNR,R3 LPC     CONTINUE IF BYTE COUNTER IS
0108 0458 17            RETC,UN         NOT Ø, ELSE RETURN
```

TRANSFER OPERANDS ROUTINE

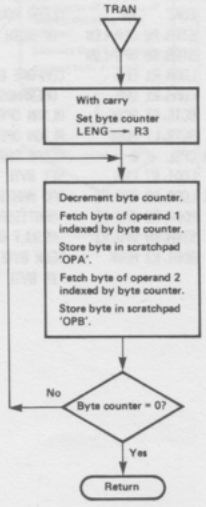


Figure 3

```
LINE ADDR  OBJECT  E SOURCE

0110               ************************************************
0111               *
0112               * SUBTRACTION ROUTINE *
0113               *
0114               * THIS ROUTINE SUBTRACTS TWO BINARY FLOATING POINT
0115               * NUMBERS
0116               *
0117 0459 3B6B     BSUB BSTR,UN TRAN     TRANSFER OPERANDS
0118 045B 3F0533       BSTA,UN TCOM      PERFORM TWO'S COMPLEMENT
0119 045E 1B2E         BCTR,UN ADDJ
0120               *
0121 0460 0607     ADDC LODI,R2 <OPB     ADDRESS OF OPERAND 2 IN
0122 0462 0789         LODI,R3 >OPB      REGISTERS
0123 0464 1B07         BCTR,UN ADDE
0124               *
0125 0466 0607     ADDD LODI,R2 <OPA     ADDRESS OF OPERAND 1 IN
0126 0468 0784         LODI,R3 >OPA      REGISTERS
0127 046A 0D0789       LODA,R1 OPB       FETCH EXP OF OPERAND 1 IN R1
0128 046D CE0781   ADDE STRA,R2 ADR      SET INDIRECT ADDRESS WITH
0129 0470 CF0782       STRA,R3 ADR+1     ADDRESS WHICH IS IN REGISTERS
0130 0473 0701     ADDF LODI,R3 1        SET BYTE COUNTER
0131 0475 7701         PPSL   C          SET CARRY
0132 0477 0FE781       LODA,R0 *ADR,R3   FETCH M.S.BYTE OF MANTISSA
0133 047A 1A02         BCTR,N  ADDG      IF BYTE IS NEGATIVE, BRANCH
0134 047C 7501         CPSL   C          ELSE CLEAR CARRY
0135 047E 0604     ADDG LODI,R2 LEN      SET END OF BYTE COUNTER
0136 0480 3F051B       BSTA,UN RRIN      ROTATE RIGHT MANTISSA AND
0137               *                      INCREMENT EXPONENT
0138 0483 0C8781       LODA,R0 *ADR      TEST TWO EXPONENTS
0139 0486 E1           COMZ   R1
0140 0487 986A         BCFR,EQ ADDF      IF NOT EQUAL, CONTINUE
0141 0489 1B16         BCTR,UN ADDH      IF EQUAL, ALIGN READY, GO BACK
0142               *
0143               ************************************************
0144               *
0145               * ADDITION ROUTINE *
0146               *
0147               * THIS ROUTINE ADDS TWO BINARY FLOATING POINT NUMBERS
0148               *
0149 0488 3F0446   BADD BSTA,UN TRAN      TRANSFER OPERANDS
0150 048E 7502     ADDJ CPSL   COM        ARITHMETIC COMPARE
0151 0490 20           EORZ   R0         CLEAR ROUNDING BYTE
0152 0491 CC0788       STRA,R0 OPA+LEN    OF BOTH OPERANDS
0153 0494 CC078D       STRA,R0 OPB+LEN
0154 0497 0D0784       LODA,R1 OPA       COMPARE EXPONENTS OF THE TWO
0155 049A ED0789       COMA,R1 OPB        OPERANDS
0156 049D 1941         BCTR,GT ADDC      ALIGN OPERAND 2
0157 049F 1A45         BCTR,LT ADDD      ALIGN OPERAND 1
0158 04A1 7501     ADDH CPSL   C         CLEAR CARRY
0159 04A3 0704         LODI,R3 LEN       SET BYTE COUNTER
0160 04A5 0F6789   ADDK LODA,R0 OPB,R3    ADD MANTISSA OPERAND 1 AND
0161 04A8 8F6784       ADDA,R0 OPA,R3     MANTISSA OPERAND 2 AND STORE
0162 04AB CF6789       STRA,R0 OPB,R3     RESULT ON PLACE OF OPERAND 2
0163 04AE FB75         BDRR,R3 ADDK      DECR BYTE COUNTER AND BRANCH
0164               *                      IF BYTE COUNTER IS NOT 0
```

```
LINE ADDR  OBJECT  E SOURCE

0166 04B0 0407     ROOF LODI,R0 <OPB     ADDRESS OF OPERAND 2 (RESULT)
0167 04B2 0589          LODI,R1 >OPB      IN REGISTERS
0168 04B4 3F0510        BSTA,UN OVFL      INCR EXPONENT AND ROTATE RIGHT
0169               *                      MANTISSA OF RESULT IF OVF = 1
0170 04B7 3F0675        BSTA,UN NORM      NORMALIZE RESULT
0171 04BA 0704          LODI,R3 LEN       SET BYTE COUNTER
0172 04BC 0F6789        LODA,R0 OPB,R3    FETCH ROUNDING BYTE OF RESULT
0173 04BF 00078A        LODA,R1 OPB+1     FETCH M.S. BYTE OF RESULT
0174 04C2 1913          BCTR,P  ROFG      BRANCH IF RESULT IS POSITIVE
0175 04C4 7701     ROFE PPSL    C         CLEAR BORROW
0176 04C6 AC0780        SUBA,R0 ROUN      SUBTRACT ROUNDING CONSTANT
0177 04C9 0F4789   ROFF LODA,R0 OPB,R3,-  SUBTRACT BORROW FROM MANTISSA
0178 04CC A400          SUBI,R0 0          RESULT
0179 04CE CF6789        STRA,R0 OPB,R3    STORE BYTE IN RESULT
0180 04D1 E701          COMI,R3 1         TEST AND BRANCH IF SUBTRACTION
0181 04D3 9874          BCFR,EQ ROFF       OF BORROW IS NOT READY
0182 04D5 1B11          BCTR,UN ROFH      CONTINUE
0183               *
0184 04D7 7501     ROFG CPSL    C         CLEAR CARRY
0185 04D9 8C0780        ADDA,R0 ROUN      ADD ROUNDING CONSTANT
0186 04DC 0F4789   ROFA LODA,R0 OPB,R3,-  ADD CARRY AND MANTISSA RESULT
0187 04DF 8400          ADDI,R0 0
0188 04E1 CF6789        STRA,R0 OPB,R3    STORE RESULT
0189 04E4 E701          COMI,R3 1         TEST AND BRANCH IF ADDITION
0190 04E6 9874          BCFR,EQ ROFA       OF CARRY IS NOT READY
0191 04E8 3B2C     ROFH BSTR,UN OVF5      INCR EXPONENT AND ROTATE RIGHT
0192               *                      MANTISSA RESULT IF OVF = 1
0193 04EA 0603          LODI,R2 LEN-1     SET BYTE COUNTER
0194 04EC 3F0677        BSTA,UN NORA      NORMALIZE RESULT
0195 04EF 0704          LODI,R3 LEN       SET BYTE COUNTER
0196 04F1 05FF          LODI,R1 H'FF'     SET FLAG
0197 04F3 0603          LODI,R2 3         TABLE INDEX
0198 04F5 E702     ROFB COMI,R3 2         CHANGE TABLE INDEX IF
0199 04F7 1902          BCTR,GT ROFD       BYTE COUNTER IS LESS THAN 2
0200 04F9 03            LODZ   R3
0201 04FA C2            STRZ   R2
0202 04FB 0F4789   ROFD LODA,R0 OPB,R3,-  TRANSFER RESULT FROM SCRATCH-
0203 04FE CFE444        STRA,R0 *PNTR,R3   PAD TO DEFINED RESULT AREA
0204 0501 EE650C        COMA,R0 TBL-1,R2  COMPARE RESULT WITH ILLEGAL
0205               *                      VALUE
0206 0504 1802          BCTR,EQ ROFC      CLEAR FLAG IF NOT EQUAL
0207 0506 0500          LODI,R1 0
0208 0508 5868     ROFC BRNR,R3 ROFB      TEST AND BRANCH IF TRANSFER
0209               *                      AND COMPARE IS NOT READY
0210 050A 01            LODZ   R1         TEST AND BRANCH TO ERROR HALT
0211 050B 14            RETC,Z             IF RESULT HAS THE ILLEGAL
0212 050C 40       ERRI HALT              VALUE,ELSE RETURN
0213               *
0214 050D 7F8000   TBL  DATA   H'7F,80,00' ILLEGAL VALUE OF RESULT
0215               *
```

ADDITION/SUBTRACTION AND ROUNDOFF ROUTINES

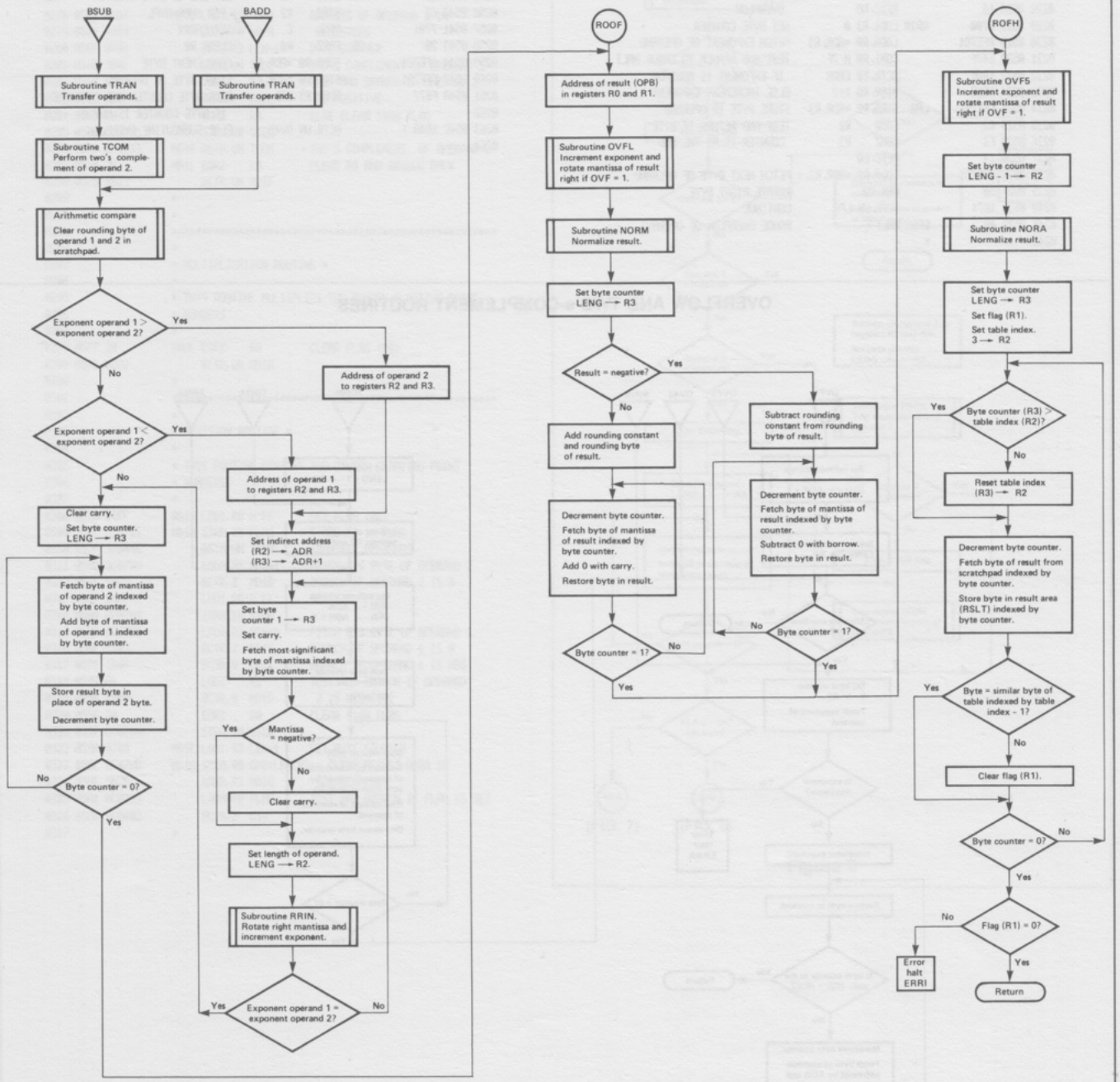


Figure 4

```
LINE ADDR  OBJECT  E SOURCE

0217              ************************************************
0218              *
0219              * OVERFLOW ROUTINE *
0220              *
0221              * THIS ROUTINE INCREMENTS THE EXPONENT AND ROTATES
0222              * THE MANTISSA RIGHT IF THE OVF FLAG = 1
0223              *
0224 0510 CC0781  OVFL STRA,R0 ADR      SET INDIRECT ADDRESS WITH
0225 0513 CD0782      STRA,R1 ADR+1     ADDRESS OF OPERAND
0226 0516 0604    OVF5 LODI,R2 LEN     SET END OF BYTE COUNTER
0227 0518 B504    OV45 TPSL    OVF     TEST AND RETURN IF NO MANTISSA
0228 051A 16           RETC,NO        OVERFLOW
0229 051B 0700    RRIN LODI,R3 0       SET BYTE COUNTER
0230 051D 0FE781       LODA,R0 *ADR,R3 FETCH EXPONENT OF OPERAND
0231 0520 E47F         COMI,R0 H'7F'   TEST AND BRANCH TO ERROR HALT
0232 0522 188E         BCTR,EQ ERRA     IF EXPONENT IS MAXIMUM
0233 0524 D800         BIRR,R0 $+2     ELSE INCREMENT EXPONENT
0234 0526 CFE781  LPA  STRA,R0 *ADR,R3 STORE BYTE IN OPERAND
0235 0529 02           LODZ    R2      TEST AND RETURN IF BYTE
0236 052A E3           COMZ    R3       COUNTER IS AT THE END
0237 052B 14           RETC,EQ
0238 052C 0FA781       LODA,R0 *ADR,R3,+ FETCH NEXT BYTE OF OPERAND
0239 052F 50           RRR,R0          ROTATE RIGHT BYTE
0240 0530 1B74         BCTR,UN LPA     CONTINUE
0241 0532 40      ERRA HALT            RANGE OVERFLOW OF OPERAND
0242              *

0243              ************************************************
0244              *
0245              * TWO'S COMPLEMENT ROUTINE
0246              *
0247              * THIS ROUTINE PREFORMS A TWO'S COMPLEMENT OF THE
0248              * BINARY FLOATING POINT NUMBER ADDRESSED BY ADR.
0249              *
0250 0533 0703    TCOM LODI,R3 LEN-1   BYTE COUNTER
0251 0535 0607    TWOA LODI,R2 <OPB    ADDRESS OF OPERAND 2
0252 0537 0489         LODI,R0 >OPB     IN REGISTERS
0253 0539 CE0781  TWOB STRA,R2 ADR     SET INDIRECT ADDRESS
0254 053C CC0782       STRA,R0 ADR+1
0255 053F 03      TWOC LODZ    R3      SET END OF BYTE COUNTER
0256 0540 C2           STRZ    R2       FOR SUBR OVFL
0257 0541 7701         PPSL    C       SET CARRY
0258 0543 20      LPB  EORZ    R0      CLEAR R0
0259 0544 AFE781       SUBA,R0 *ADR,R3 COMPLEMENT BYTE
0260 0547 CFE781       STRA,R0 *ADR,R3 STORE BYTE IN OPERAND
0261 054A FB77         BDRR,R3 LPB     DECR BYTE COUNTER AND CONTINUE
0262              *                      IF BYTE COUNTER IS NOT 0
0263 054C 1B4A         BCTR,UN OV45    ELSE SUBROUTINE OVERFLOW
0264              *
```

OVERFLOW AND TWO's-COMPLEMENT ROUTINES

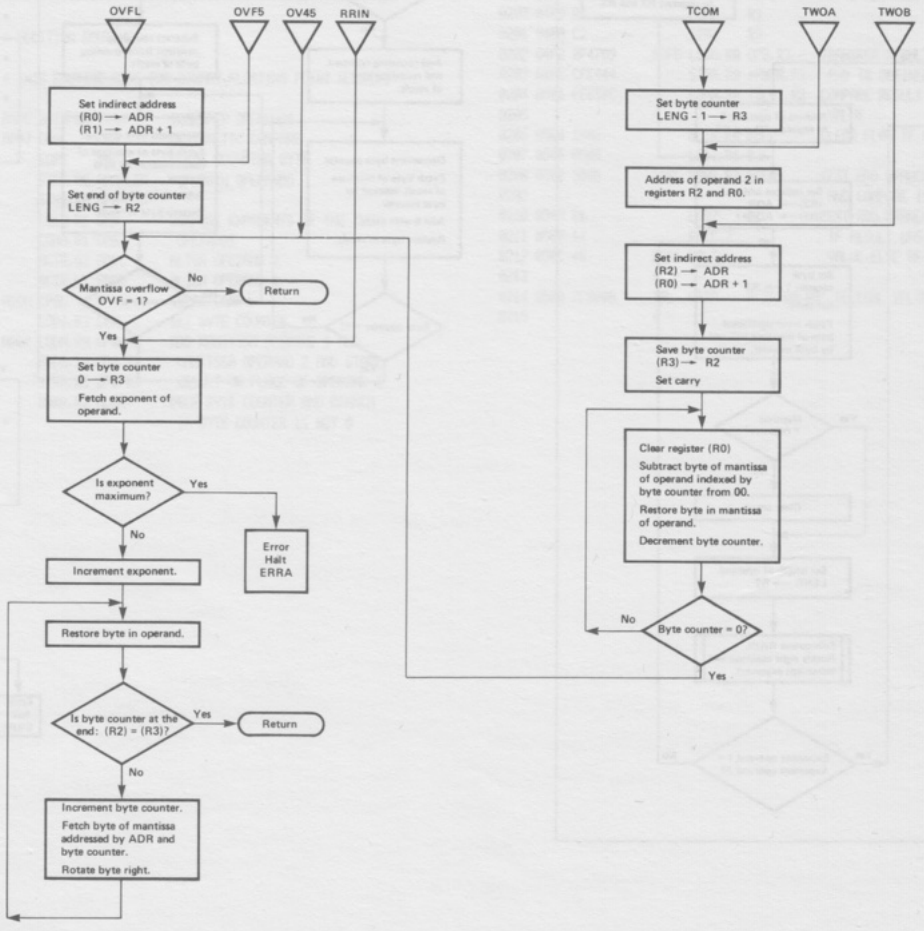


Figure 5

```
0265                ************************************************
0266                *
0267 054E 40        ERRC HALT              DUMMY ERROR
0268 054F 0C0783    MDIB LODA,R0 FLAG      TEST AND BRANCH TO ERROR
0269 0552 987A          BCFR,Z  ERRC        HALT IF FLAG IS SET
0270 0554 20        MDIC EORZ   R0         CLEAR R0
0271 0555 CFA444    MDIN STRA,R0 *PNTR,R3,+ RESULT OF OPERATION IS ZERO
0272 0558 E703          COMI,R3 LEN-1      STORE 0 IN MANTISSA RESULT
0273 055A 9879          BCFR,EQ MDIN        AND H'80' IN EXPONENT
0274 055C 0480          LODI,R0 H'80'
0275 055E CC8444        STRA,R0 *PNTR
0276 0561 17            RETC,UN
0277                *
0278 0562 0607      MDIE LODI,R2 <OPA      ADDRESS OF OPERAND 1 IN
0279 0564 0484          LODI,R0 >OPA        REGISTERS
0280 0566 0703          LODI,R3 LEN-1      BYTE COUNTER
0281 0568 3B4F          BSTR,UN TWOB       TWO'S COMPLEMENT OF OPERAND 1
0282 056A 01            LODZ    R1         TEST AND BRANCH IF OPERAND 2
0283 056B 1907          BCTR,P  MDIL        IS POSITIVE
0284 056D 20            EORZ    R0         ELSE CLEAR SIGN FLAG
0285 056E CC0788        STRA,R0 SIGN
0286 0571 3F0533    MDID BSTA,UN TCOM      TWO'S COMPLEMENT  OF OPERAND 2
0287 0574 20        MDIL EORZ   R0         CLEAR R0 AND BRANCH BACK
0288 0575 1B23          BCTR,UN MDIF
0289                *
0290                *
0291                ************************************************
0292                *
0293                * MULTIPLICATION ROUTINE *
0294                *
0295                * THIS ROUTINE MULTIPLIES TWO BINARY FLOATING POINT
0296                * NUMBERS
0297                *
0298 0577 20        BMUL EORZ   R0         CLEAR FLAG (R0)
0299 0578 1B02          BCTR,UN MDIA
0300                *
0301                ************************************************
0302                *
0303                * DIVISION ROUTINE *
0304                *
0305                * THIS ROUTINE DIVIDES TWO BINARY FLOATING POINT
0306                * NUMBERS
0307                *
0308 057A 04FF      BDIV LODI,R0 H'FF'     SET FLAG (R0)
0309 057C CC0783    MDIA STRA,R0 FLAG      STORE R0 IN FLAG
0310 057F 3F0446        BSTA,UN TRAN       TRANSFER OPERANDS
0311 0582 040078        LODA,R1 OPB+1      FETCH M.S. BYTE OF OPERAND 2
0312 0585 1848          BCTR,Z  MDIB       BRANCH IF OPERAND 2 IS 0
0313 0587 04FF          LODI,R0 H'FF'      SET SIGN FLAG
0314 0589 CC0788        STRA,R0 SIGN
0315 058C 0E0785        LODA,R2 OPA+1      FETCH M.S. BYTE OF OPERAND 1
0316 058F 1843          BCTR,Z  MDIC       BRANCH IF OPERAND 1 IS 0
0317 0591 1A4F          BCTR,N  MDIE       BRANCH IF OPERAND 1 IS NEG
0318 0593 01            LODZ    R1         TEST AND BRANCH IF OPERAND
0319 0594 1A5B          BCTR,N  MDID        2 IS NEGATIVE
0320 0596 20            EORZ    R0         CLEAR SIGN FLAG
0321 0597 CC0788        STRA,R0 SIGN
0322 059A 0705      MDIF LODI,R3 LEN+1     SET BYTE COUNTER
0323 059C CF478D    MDIG STRA,R0 OPB+LEN,R3,- CLEAR RESULT AREA IN
0324 059F 5B7B          BRNR,R3 MDIG        SCRATCH-PAD
0325 05A1 0C0783        LODA,R0 FLAG       TEST AND BRANCH IF FLAG IS SET
0326 05A4 9C0682        BCFA,Z  DIV
0327                *
```

MULTIPLICATION/DIVISION ROUTINE

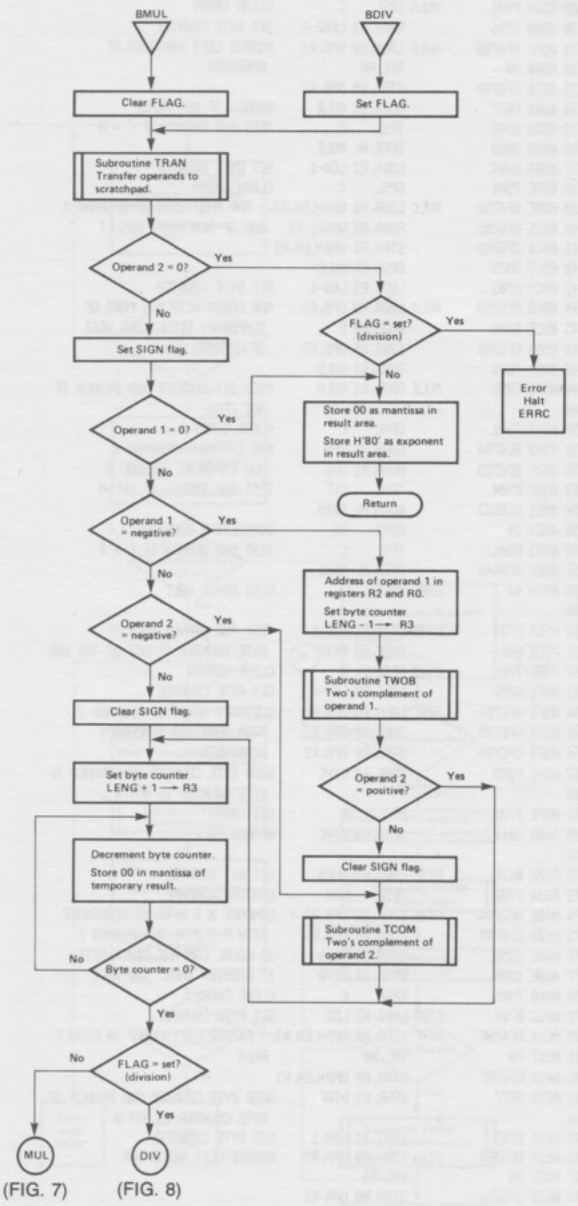


Figure 6

```
LINE ADDR  OBJECT  E SOURCE

0328 05A7 0619      MUL  LODI,R2 MLEN     SET BIT COUNTER
0329 05A9 7501      MULA CPSL    C        CLEAR CARRY
0330 05AB 0706           LODI,R3 LEN2-2   SET BYTE COUNTER
0331 05AD 0F6789    MULB LODA,R0 OPB,R3   ROTATE LEFT MANTISSA OF
0332 05B0 D0             RRL,R0           OPERAND2
0333 05B1 CF6789         STRA,R0 OPB,R3
0334 05B4 FB77           BDRR,R3 MULB     BRANCH IF NOT READY
0335 05B6 B501           TPSL    C        TEST AND BRANCH IF C = 0
0336 05B8 981B           BCFR,AL MULE
0337 05BA 0703           LODI,R3 LEN-1    SET BYTE COUNTER
0338 05BC 7501           CPSL    C        CLEAR CARRY
0339 05BE 0F478D    MULC LODA,R0 OPB+LEN,R3,- ADD MANTISSAS OF OPERAND 1
0340 05C1 8F6785         ADDA,R0 OPA+1,R3   AND OF TEMPORARY RESULT
0341 05C4 CF678D         STRA,R0 OPB+LEN,R3
0342 05C7 5B75           BRNR,R3 MULC
0343 05C9 0703           LODI,R3 LEN-1    SET BYTE COUNTER
0344 05CB 0F6789    MULD LODA,R0 OPB,R3   ADD CARRY WITH M.S. PART OF
0345 05CE 8400           ADDI,R0 0          TEMPORARY RESULT AND REST
0346 05D0 CF6789         STRA,R0 OPB,R3     OF OPERAND 2
0347 05D3 FB76           BDRR,R3 MULD
0348 05D5 FA52      MULE BDRR,R2 MULA     DECR BIT COUNTER AND BRANCH IF
0349                *                      NOT ZERO
0350 05D7 7501           CPSL    C        CLEAR CARRY
0351 05D9 0D0784         LODA,R1 OPA      ADD EXPONENT OPERAND 1
0352 05DC 8D0789         ADDA,R1 OPB       AND EXPONENT OPERAND 2
0353 05DF B504           TPSL    OVF      TEST AND BRANCH IF OVF=0
0354 05E1 9C0663         BCFA,AL MDIH
0355 05E4 20             EORZ    R0       CORRECTING CONSTANT = 0
0356 05E5 B501           TPSL    C        TEST AND BRANCH IF C = 0
0357 05E7 9C0646         BCFA,AL MDII
0358 05EA 40        ERRG HALT             ELSE ERROR HALT
0359                *
0360 05EB E703      DIVA COMI,R3 LEN-1    TEST AND BRANCH BACK IF
0361 05ED 9817           BCFR,EQ DIVD     BYTE COUNTER IS NOT AT THE END
0362 05EF 7701      DIVB PPSL    C        CLEAR BORROW
0363 05F1 0703           LODI,R3 LEN-1    SET BYTE COUNTER
0364 05F3 0F6784    DIVC LODA,R0 OPA,R3   SUBTRACT MANTISSA OPERAND 2
0365 05F6 AF6789         SUBA,R0 OPB,R3    FROM MANTISSA TEMPORARY
0366 05F9 CF6784         STRA,R0 OPA,R3    REMAINDER
0367 05FC FB75           BDRR,R3 DIVC     DECR BYTE COUNTER AND BRANCH IF
0368                *                      BYTE COUNTER IS NOT 0
0369 05FE 7701           PPSL    C        SET CARRY
0370 0600 1B10           BCTR,UN DIVE     BRANCH BACK
0371                *
0372 0602 061F      DIV  LODI,R2 DLEN     SET BIT COUNTER
0373 0604 7702           PPSL    COM      LOGICAL COMPARE
0374 0606 0F2784    DIVD LODA,R0 OPA,R3,+ COMPARE M.S. BYTE OF REMAINDER
0375 0609 EF6789         COMA,R0 OPB,R3    WITH M.S. BYTE OF OPERAND 2
0376 060C 185D           BCTR,EQ DIVA     IF EQUAL COMPARE OTHER BYTES
0377 060E 195F           BCTR,GT DIVB     IF GREATER THAN, SUBTRACT
0378 0610 7501           CPSL    C        CLEAR CARRY
0379 0612 0704      DIVE LODI,R3 LEN      SET BYTE COUNTER
0380 0614 0F478D    DIVF LODA,R0 OPB+LEN,R3,- ROTATE LEFT CARRY IN RESULT
0381 0617 D0             RRL,R0           AREA
0382 0618 CF678D         STRA,R0 OPB+LEN,R3
0383 061B 5B77           BRNR,R3 DIVF     DECR BYTE COUNTER AND BRANCH IF
0384                *                      BYTE COUNTER IS NOT 0
0385 061D 0703           LODI,R3 LEN-1    SET BYTE COUNTER
0386 061F 0F6784    DIVG LODA,R0 OPA,R3   ROTATE LEFT REMAINDER
0387 0622 D0             RRL,R0
0388 0623 CF6784         STRA,R0 OPA,R3
0389 0626 FB77           BDRR,R3 DIVG     DECR BYTE COUNTER AND BRANCH IF
0390                *                      BYTE COUNTER IS NOT 0
0391 0628 FA5C           BDRR,R2 DIVD     DECR BIT COUNTER AND BRANCH IF
0392                *                      BIT COUNTER IS NOT 0
0393 062A 0F278C    DIVH LODA,R0 OPB+LEN-1,R3,+ TRANSFER MANTISSA RESULT
0394 062D CF6789         STRA,R0 OPB,R3    TO PLACE OF MANTISSA OF
0395 0630 E704           COMI,R3 LEN       OPERAND 2 ON SCRATCH-PAD
0396 0632 9876           BCFR,EQ DIVH
0397 0634 7701           PPSL    C        CLEAR BORROW
0398 0636 0D0784         LODA,R1 OPA      SUBTRACT EXPONENT OPERAND 2
0399 0639 AD0789         SUBA,R1 OPB       FROM EXPONENT OPERAND 1
0400 063C 0401           LODI,R0 1        CORRECTING CONSTANT = 1
0401 063E B504           TPSL    OVF      TEST AND BRANCH IF OVF = 0
0402 0640 9818           BCFR,AL MDIK
0403 0642 B501           TPSL    C        TEST AND BRANCH TO ERROR HALT
0404 0644 9813           BCFR,AL ERRF      IF C = 0
0405 0646 CC0789    MDII STRA,R0 OPB      STORE TEMPORARY EXP. RESULT
0406 0649 3B2A           BSTR,UN NORM     NORMALIZE RESULT
0407 064B 7501           CPSL    C        CLEAR CARRY
0408 064D 0C0789         LODA,R0 OPB      ADD TEMPORARY EXPONENT AND
0409 0650 81             ADDZ    R1        CORRECTING CONSTANT
0410 0651 CC0789         STRA,R0 OPB      STORE EXPONENT RESULT
0411 0654 B504           TPSL    OVF      TEST AND BRANCH IF OVF = 1
0412 0656 1810           BCTR,AL MDIJ
0413 0658 40        ERRH HALT             ELSE ERROR HALT
0414 0659 40        ERRF HALT             ERROR HALT
0415                *
0416 065A 7501      MDIK CPSL    C        CLEAR CARRY
0417 065C 81             ADDZ    R1       ADD TEMPORARY EXPONENT AND
0418 065D C1             STRZ    R1        CORRECTING CONSTANT
0419 065E 20             EORZ    R0       NEW CORRECTING CONSTANT = 0
0420 065F B504           TPSL    OVF      TEST AND BRANCH IF OVF = 1
0421 0661 1863           BCTR,AL MDII
0422 0663 CD0789    MDIH STRA,R1 OPB      STORE EXPONENT
0423 0666 3B0D           BSTR,UN NORM     NORMALIZE RESULT
0424 0668 0704      MDIJ LODI,R3 LEN      BYTE COUNTER
0425 066A 0C0788         LODA,R0 SIGN     TEST AND PREFROM TWO'S
0426 066D BC0535         BSFA,Z  TWOA      COMPLEMENT IF SIGN = SET
0427 0670 7504           CPSL    OVF      CLEAR OVF FLAG
0428 0672 1F0480         BCTR,UN ROOF     ROUND RESULT
0429                *
```

MULTIPLICATION ROUTING (Cont.)

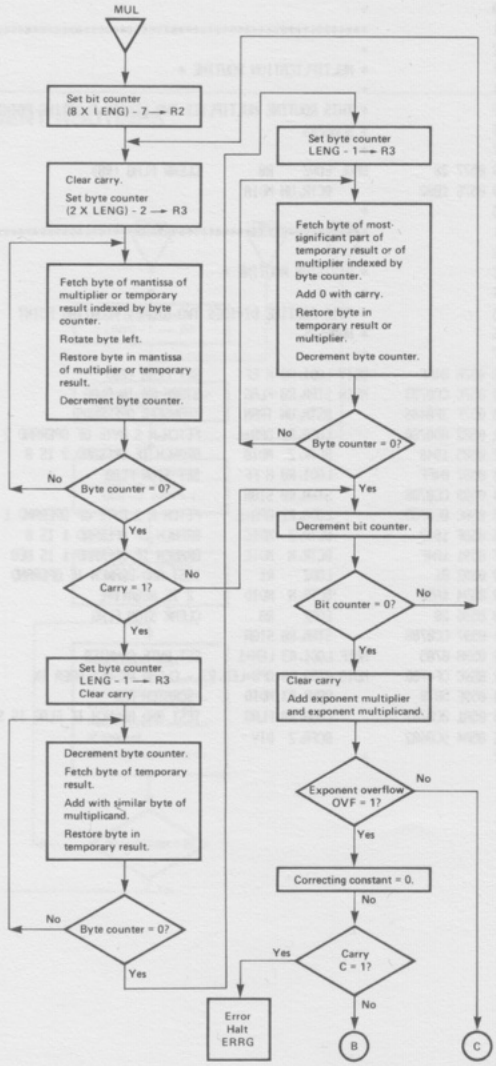
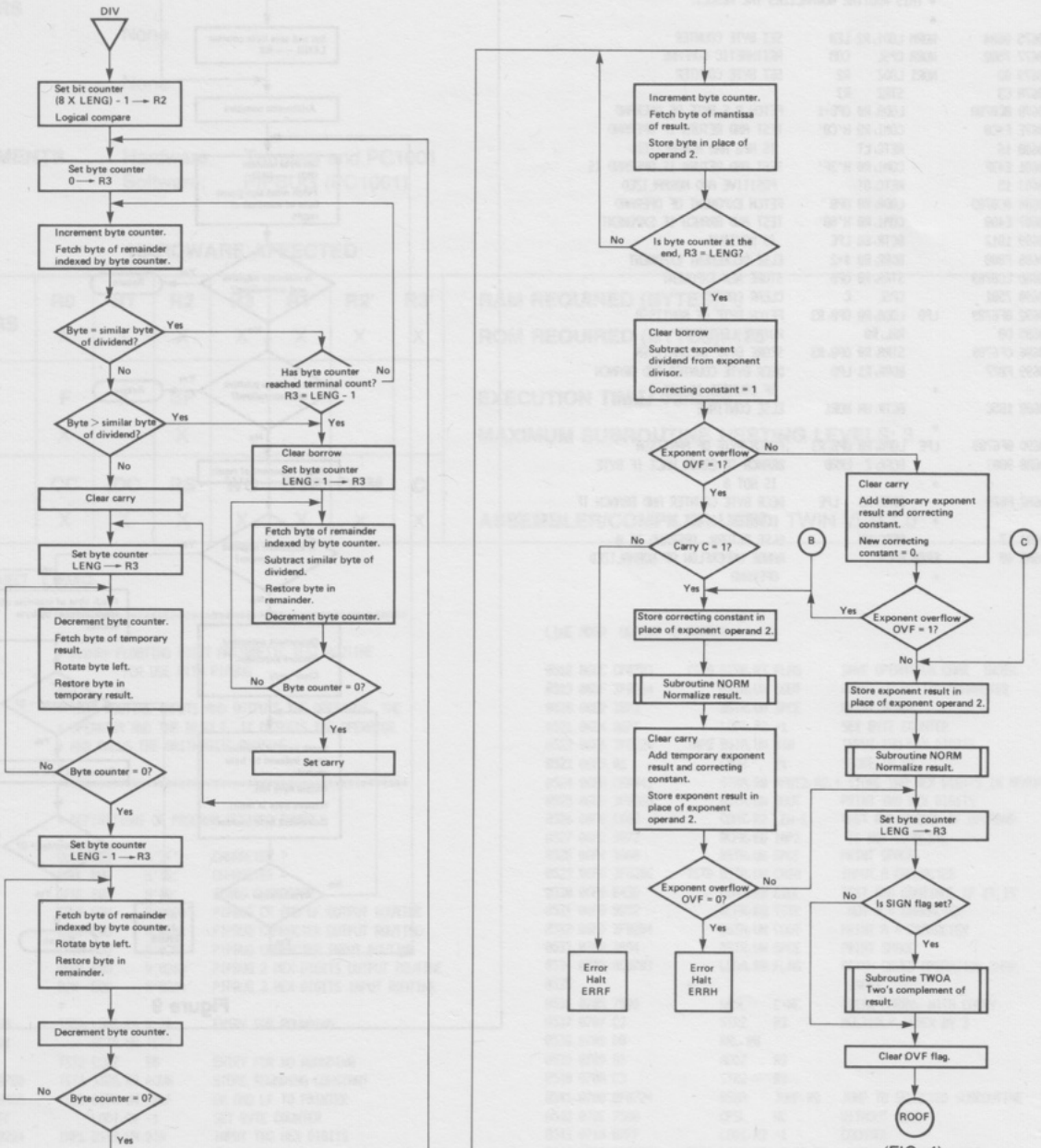


Figure 7

DIVISION ROUTINE (Cont.)



(FIG. 4)

Figure 8

```
LINE ADDR  OBJECT  E SOURCE

0431                ************************************************
0432                *
0433                * NORMALIZE ROUTINE *
0434                *
0435                * THIS ROUTINE NORMALIZES THE RESULT.
0436                *
0437 0675 0604     NORM LODI,R2 LEN      SET BYTE COUNTER
0438 0677 7502     NORA CPSL    COM      ARITHMETIC COMPARE
0439 0679 02       NOR1 LODZ    R2       SET BYTE COUNTER
0440 067A C3            STRZ    R3
0441 067B 0C078A        LODA,R0 OPB+1    FETCH M.S.BYTE OF OPERAND
0442 067E E4C0          COMI,R0 H'C0'    TEST AND RETURN IF OPERAND
0443 0680 16            RETC,LT          IS NEG AND NORMALIZED
0444 0681 E43F          COMI,R0 H'3F'    TEST AND RETURN IF OPERAND IS
0445 0683 15            RETC,GT          POSITIVE AND NORMALIZED
0446 0684 0C0789        LODA,R0 OPB      FETCH EXPONENT OF OPERAND
0447 0687 E480          COMI,R0 H'80'    TEST AND BRANCH IF EXPONENT
0448 0689 1812          BCTR,EQ LPE      IS MAXIMUM
0449 068B F800          BDRR,R0 $+2      ELSE DECREMENT EXPONENT
0450 068D CC0789        STRA,R0 OPB      STORE NEW EXPONENT
0451 0690 7501          CPSL    C        CLEAR CARRY
0452 0692 0F6789   LPD  LODA,R0 OPB,R3   FETCH BYTE OF MANTISSA
0453 0695 D0            RRL,R0           ROTATE LEFT BYTE
0454 0696 CF6789        STRA,R0 OPB,R3   STORE BYTE IN MANTISSA
0455 0699 FB77          BDRR,R3 LPD      DECR BYTE COUNTER AND BRANCH
0456                *                    IF COUNTER IS NOT 0
0457 069B 1B5C          BCTR,UN NOR1     ELSE CONTINUE
0458                *
0459 069D 0F6789   LPE  LODA,R0 OPB,R3   FETCH BYTE OF MANTISSA
0460 06A0 9803          BCFR,Z  ERRB     BRANCH TO ERROR HALT IF BYTE
0461                *                    IS NOT 0
0462 06A2 FB79          BDRR,R3   LPE    DECR BYTE COUNTER AND BRANCH IF
0463                *                    COUNTER IS NOT 0
0464 06A4 17            RETC,UN          ELSE RETURN, OPERAND IS 0
0465 06A5 40       ERRB HALT            RANGE UNDERFLOW OF NORMALIZED
0466                *                    OPERAND
```

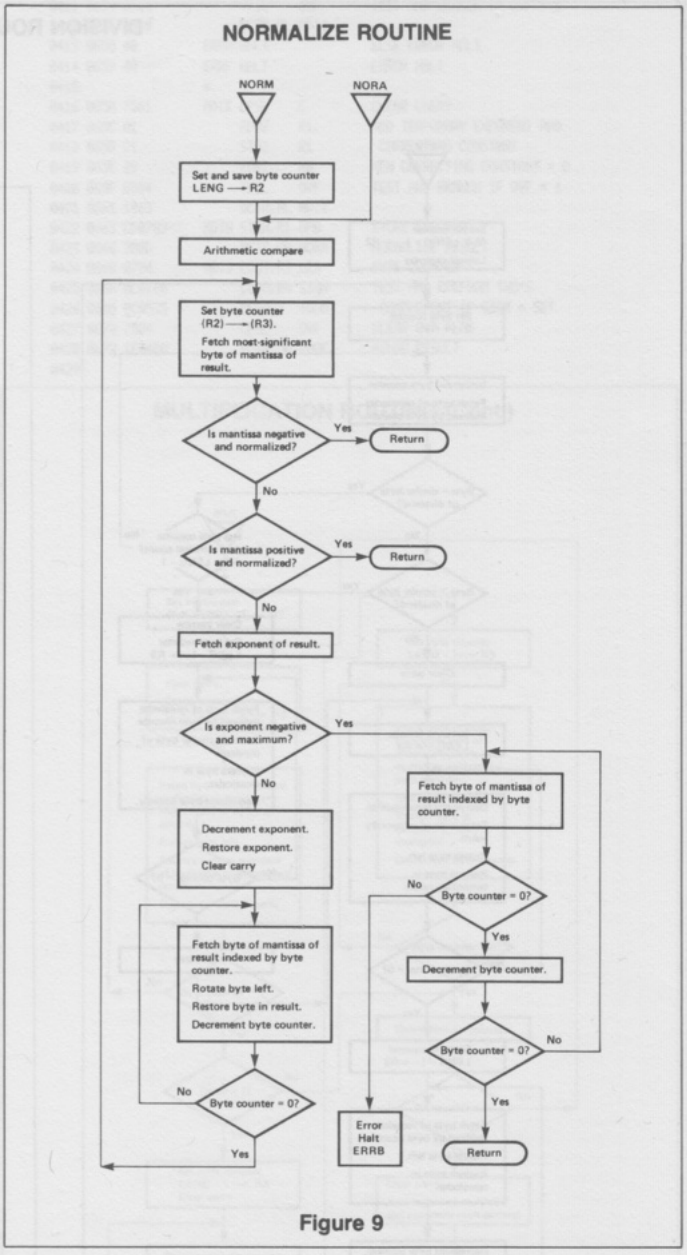


Figure 9

PROGRAM TITLE	TEST ROUTINE FOR BINARY ARITHMETIC FLOATING POINT ROUTINES
FUNCTION	Inputs and echoes operands and operator and outputs the result of the operation via a teletype.
PARAMETERS	
INPUT:	None
OUTPUT:	None
SPECIAL REQUIREMENTS	Hardware: Terminal and PC1001 Software: PIPBUG (PC1001)

HARDWARE AFFECTED

REGISTERS	**R0**	**R1**	**R2**	**R3**	**R1'**	**R2'**	**R3'**
	X	X	X	X	X	X	X
PSU	**F**	**II**	**SP**				
	X		X				
PSL	**CC**	**IDC**	**RS**	**WC**	**OVF**	**COM**	**C**
	X	X	X	X	X	X	X

RAM REQUIRED (BYTES): 0

ROM REQUIRED (BYTES): 138

EXECUTION TIME: Variable

MAXIMUM SUBROUTINE NESTING LEVELS: 3

ASSEMBLER/COMPILER USED: TWIN VER 1.0

```
LINE ADDR  OBJECT  E SOURCE

0468                ************************************************
0469                *
0470                * BINARY FLOATING POINT ARITHMETIC TEST ROUTINE
0471                *       FOR USE WITH PIPBUG
0472                *
0473                * THIS ROUTINE INPUTS AND OUTPUTS THE OPERANDS, THE
0474                * OPERATOR AND THE RESULT.  IT DETECTS THE OPERATOR
0475                * AND CALLS THE ARITHMETIC ROUTINE.
0476                *
0477                ************************************************
0478                *
0479                * DEFINITIONS OF PROGRAM DEFINED SYMBOLS
0480                *
0481 003F          QUES EQU    H'3F'     CHARACTER ?
0482 003D          EQUL EQU    H'3D'     CHARACTER =
0483 0020          SPAC EQU    H'20'     SPACE CHARACTER
0484 008A          CRLF EQU    H'008A'   PIPBUG CR AND LF OUTPUT ROUTINE
0485 02B4          COUT EQU    H'02B4'   PIPBUG CHARACTER OUTPUT ROUTINE
0486 0286          CHIN EQU    H'0286'   PIPBUG CHARACTER INPUT ROUTINE
0487 0269          BOUT EQU    H'0269'   PIPBUG 2 HEX DIGITS OUTPUT ROUTINE
0488 0224          BIN  EQU    H'0224'   PIPBUG 2 HEX DIGITS INPUT ROUTINE
0489                *
0490 06A6 0480     TST1 LODI,R0 H'80'     ENTRY FOR ROUNDING
0491 06A8 1B01          BCTR,UN TST4
0492 06AA 20       TST2 EORZ   R0        ENTRY FOR NO ROUNDING
0493 06AB CC0780   TST4 STRA,R0 ROUN      STORE ROUNDING CONSTANT
0494 06AE 3F008A   TST3 BSTA,UN CRLF      CR AND LF TO PRINTER
0495 06B1 06FF          LODI,R2 -1        SET BYTE COUNTER
0496 06B3 3F0224   INP1 BSTA,UN BIN       INPUT TWO HEX DIGITS
0497 06B6 01            LODZ   R1         STORE IN R0
0498 06B7 CEA440        STRA,R0 *PNT1,R2,+ STORE TWO HEX DIGITS IN MEMORY
0499 06BA 3F0269        BSTA,UN BOUT      PRINT TWO HEX DIGITS
0500 06BD E603          COMI,R2 LEN-1     TEST AND BRANCH IF OPERAND
0501 06BF 9872          BCFR,EQ INP1       IS NOT  COMPLETE
0502 06C1 3B13          BSTR,UN SPCE      PRINT SPACE
0503 06C3 3F0286   TSTC BSTA,UN CHIN      INPUT OPERATION CHARACTER
0504 06C6 0704          LODI,R3 4         OPERATION CHARACTER COUNTER
0505 06C8 EF4720   TSTA COMA,R0 OPSI,R3,- TEST AND BRANCH IF IT
0506 06CB 180F          BCTR,EQ TSTB       IS AN OPERATION CHARACTER
0507 06CD 5B79          BRNR,R3 TSTA      TEST AND BRANCH IF COUNTER IS
0508                *                      NOT ZERO
0509 06CF 043F     TSTK LODI,R0 QUES      PRINT ?
0510 06D1 3F02B4        BSTA,UN COUT
0511 06D4 1B58          BCTR,UN TST3      START A CALCULATION AGAIN
0512                *
0513 06D6 0420     SPCE LODI,R0 SPAC      SPACE CHARACTER IN R0

BSTA,UN COUT
RETC,UN

LINE ADDR  OBJECT  E SOURCE

0518 06DC CF0783   TSTB STRA,R3 FLAG      SAVE OPERATION CHAR. INDEX
0519 06DF 3F02B4        BSTA,UN COUT      PRINT OPERATION CHARACTER
0520 06E2 3B72          BSTR,UN SPCE      PRINT SPACE
0521 06E4 06FF          LODI,R2 -1        SET BYTE COUNTER
0522 06E6 3F0224   INP2 BSTA,UN BIN       INPUT TWO HEX DIGITS
0523 06E9 01            LODZ   R1         STORE IN R0
0524 06EA CEA442        STRA,R0 *PNT2,R2,+ STORE TWO HEX DIGITS IN MEMORY
0525 06ED 3F0269        BSTA,UN BOUT      PRINT TWO HEX DIGITS
0526 06F0 E603          COMI,R2 LEN-1     TEST AND BRANCH IF OPERAND
0527 06F2 9872          BCFR,EQ INP2       IS NOT COMPLETE
0528 06F4 3B60          BSTR,UN SPCE      PRINT SPACE
0529 06F6 3F0286   TSTD BSTA,UN CHIN      INPUT A CHARACTER
0530 06F9 E43D          COMI,R0 EQUL      TEST AND CONTINUE IF IT IS
0531 06FB 9852          BCFR,EQ TSTK       NOT A = CHARACTER
0532 06FD 3F02B4        BSTA,UN COUT      PRINT A = CHARACTER
0533 0700 3B54          BSTR,UN SPCE      PRINT SPACE
0534 0702 0C0783        LODA,R0 FLAG      FETCH SAVED OPERATION CHAR.
0535                *                      INDEX
0536 0705 7509          CPSL   C+WC       CLEAR CARRY, WITH CARRY
0537 0707 C3            STRZ   R3         MULTIPLY INDEX BY 3
0538 0708 D0            RRL,R0
0539 0709 83            ADDZ   R3
0540 070A C3            STRZ   R3
0541 070B BF0724        BSXA   JUMP,R3    JUMP TO SELECTED SUBROUTINE
0542 070E 7508          CPSL   WC         WITHOUT CARRY
0543 0710 07FF          LODI,R3 -1        COUNTER
0544 0712 0FA444   TSTG LODA,R0 *PNTR,R3,+ FETCH BYTE OF RESULT
0545 0715 C1            STRZ   R1
0546 0716 3F0269        BSTA,UN BOUT      PRINT TWO HEX DIGITS
0547 0719 E703          COMI,R3 LEN-1     TEST AND BRANCH IF OUTPUT
0548 071B 9875          BCFR,EQ TSTG       RESULT IS NOT READY
0549 071D 1F06AE        BCTA,UN TST3      START NEW CALCULATION AGAIN
0550                *
0551 0720 2B2D2A3A OPSI DATA   H'2B,2D,2A,3A' OPERATION CHAR: +,-,*,
0552                *
0553 0724 1F0488   JUMP BCTA,UN BADD      ADDITION ROUTINE
0554 0727 1F0459        BCTA,UN BSUB      SUBTRACTION ROUTINE
0555 072A 1F0577        BCTA,UN BMUL      MULTIPLICATION ROUTINE
0556 072D 1F057A        BCTA,UN BDIV      DIVISION ROUTINE
0557                *
0558 0446               END   TRAN

 TOTAL ASSEMBLY ERRORS = 0000
```

Signetics 2650 Microprocessor Application Memos currently available:

AS50	Serial Input/Output
AS51	Bit & Byte Testing Procedures
AS52	General Delay Routines
AS53	Binary Arithmetic Routines
AS54	Conversion Routines
AS55	Fixed Point Decimal Arithmetic Routines
AS56	2650 Sorting Routines
AS57	Binary Arithmetic Floating Point Routines
AS58	BCD Arithmetic Floating Point Routines
SP50	2650 Evaluation Printed Circuit Board (PC1001)
SP51	2650 Demo System
SP52	Support Software for use with the NCSS Timesharing System
SP53	Simulator, Version 1.2
SP54	Support Software for use with the General Electric Mark III Timesharing System
SP55	The ABC1500 Adaptable Board Computer
SS50	PIPBUG
SS51	Absolute Object Format
MP51	Initialization
MP52	Low-Cost Clock Generator Circuits
MP53	Address and Data Bus Interfacing Techniques
MP54	2650 Input/Output Structures and Interfaces

signetics

a subsidiary of **U.S. Philips Corporation**

Signetics Corporation
811 East Arques Avenue
Sunnyvale, California 94086
Telephone 408/739-7700

Feb. 1977