PHILIPS

PHILIPS Electronic components and materials

Technical note 094

2650 BCD floating point routines

signetics

The capability of operating on Binary Coded Decimal (BCD) floating point numbers is provided by the four routines presented in this technical note. These routines perform the addition, subtraction, multiplication, and division of BCD floating point operands in normalized signed-magnitude notation. Multiple precision os facilitated through a variable-length mantissa.

Data Format

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

Note that the magnitude bytes contain two BCD digits per byte. Bytes 0 and 1 are the exponent of the number in signed-magnitude form. Byte 0 is the sign, and is positive if equal to H'00' and negative otherwise. Thus, the range for the exponent (E) is:

$$-99 \leqslant E \leqslant +99$$

Bytes 2 through (LENG−1) are the mantissa in signed-magnitude form, where LENG is a symbol defined as a positive integer in the source program via an EQU assembler directive. Byte 2 is the mantissa sign and follows the same convention as the exponent sign. The remaining bytes are the magnitude, with the decimal point placed at the left of the Most-Significant Digit (MSD).

Normalized Format

All operands used as inputs to these routines must be normalized, and results are provided in normalized form. A normalized mantissa is one for which the MSD is not equal to zero. Thus, the allowable range for the magnitude of the mantissa (M) is:

$$[0.]\ 1000. \ldots \leqslant M \leqslant [0.]\ 9999. \ldots$$

Zero is defined as mantissa of zero with exponent −99.

Table I shows the range of acceptable values for the case LENG = 5, that is, a four-digit 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 normlized. 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'66' specifies no rounding

(ROUN) = H.b6' specifies round-off (round if most-significant truncated digit is M5).

(ROUN) = H'F6' specifies round-up (round if most-significant truncated digit M1).

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	DADD for addition
BSTA,UN	DSUB for subtraction
BSTA,UN	DMUL for multiplication
BSTA,UN	DDIV for division

TEST PROGRAM

The listing at the end of this technical note is a test program that 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.

Figures 3 through 14 provide flowcharts and listings for the four routines.

BCD FLOATING POINT FORMAT

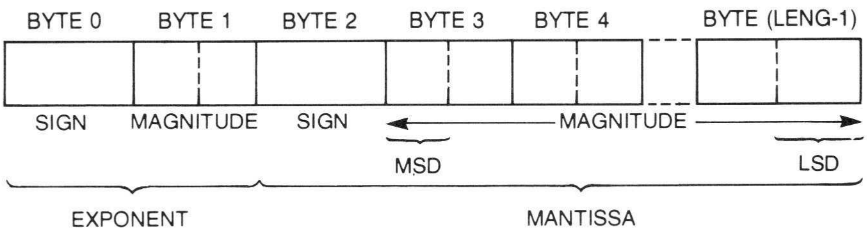


Figure 1

		EXP.	MANTISSA	DECIMAL EQUIVALENT
+	LARGEST POSITIVE	0099	009999	$+0.9999 \times 10^{99}$
↑	SMALLEST POSITIVE	FF99	001000	$+0.1000 \times 10^{-99}$
0	ZERO	FF99	000000	0.0
↓	SMALLEST NEGATIVE	FF99	FF1000	-0.1000×10^{-99}
−	LARGEST NEGATIVE	0099	FF9999	-0.9999×10^{99}

Table 1 – RANGE OF VALUES FOR A FIVE-BYTE NUMBER

OPERATION OF TEST PROGRAM

```
*G6C4
+10+12345678 + +10+98765432 = +11+11111111
+10+12345678 + +05+98765432 = +10+12346666
+05+12345678 - +10+98765432 = +10-98765309
+10+12345678 - +10-98765432 = +11+11111111
+10-12345678 + +10-98765432 = +11-11111111

-95+88776655 - -95+88665544 = -97+11111100
-96+88776655 - -96+88775544 =                 (Underflow)
*G6C4

+90+23452345 + +98+99999999 = +98+99999999
+92+23452345 + +98+99999999 = +99+10000002
+99+87654321 + +99+12345679 =                 (Overflow)
*G6C4
+55+99999999 * +44+99999999 = +99+99999998
+33+12345679 * +49+18000000 = +81+22222222
+33+12345679 * -49+81000000 = -16+10000000

+99+99999999 : +99-55667788 = +01-17963710
+99-55667788 * +01-17963709 = +99+99999994
+99-55667788 * +01-17963710 =                 (Overflow)
```

Figure 2

PROGRAM TITLE

BCD ARITHMETIC FLOATING POINT PACKAGE

FUNCTION

Performs addition, subtraction, multiplication, and division of BCD floating point numbers.

OPERAND 1 & OPERAND 2 → RESULT
& = +, −, ×, :

PARAMETERS
INPUT:

OPERAND 1 is in memory starting at address OP1.
OPERAND 2 is in memory starting at address OP2.
Length of operands and result are defined by LENG.
ROUN contains rounding constant including offset of H'66'.

OUTPUT:

RESULT of the operation is stored in memory starting at address RSLT.

SPECIAL REQUIREMENTS

None

	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 × LENG + 4

ROM REQUIRED (BYTES): 644

EXECUTION TIME: Variable

MAXIMUM SUBROUTINE NESTING LEVELS: 2

ASSEMBLER/COMPILER USED: TWIN VER 2.0

```
LINE ADDR  OBJECT  E SOURCE

0001                 * PD760110
0002                 ****************************************************
0003                 *
0004                 * BCD FLOATING POINT ARITHMETIC PACKAGE.
0005                 *
0006                 * THIS PACKAGE CONSISTS OF THE ADDITION,
0007                 *                               SUBTRACTION,
0008                 *                               MULTIPLICATION,
0009                 *                               DIVISION
0010                 *  ROUTINES FOR TWO FLOATING POINT BCD NUMBERS
0011                 *
0012                 * THE BCD FLOATING POINT NUMBERS ARE REPRESENTED
0013                 *  AS FOLLOWS: BYTE 0   = SIGN OF THE EXPONENT
0014                 *                          H'00'=POS, H'F0'=NEG
0015                 *              BYTE 1   = ABSOLUTE EXPONENT.
0016                 *              BYTE 2   = SIGN OF THE MANTISSA
0017                 *                          H'00'=POS, H'F0'=NEG
0018                 *              BYTE 3->LENG-1 = ABSOLUTE MANTISSA.
0019                 *
0020                 *              BYTE 3   = MS BYTE  OF THE MANTISSA.
0021                 *              THE POINT POSITION IS IN FRONT OF THE
0022                 *                MANTISSA.
0023                 ****************************************************
0024                 *
0025                 * DEFINITIONS OF SYMBOLS:
0026 0000            R0   EQU     0         PROCESSOR-REGISTERS
0027 0001            R1   EQU     1
0028 0002            R2   EQU     2
0029 0003            R3   EQU     3
0030 0080            S    EQU     H'80'     PSU:  SENSE
0031 0040            F    EQU     H'40'           FLAG
0032 0010            RS   EQU     H'10'     PSL.  REGISTER BANK SELECT
0033 0001            C    EQU     H'01'           CARRY/NO BORROW
0034 0002            COM  EQU     H'02'           LOGICAL COMPARE
0035 0008            WC   EQU     H'08'           1=WITH, 0=WITHOUT CARRY
0036 00FF            ALL  EQU     H'FF'           ALL BITS
0037 0000            Z    EQU     0         BRANCH COND.  ZERO
0038 0001            P    EQU     1                       POSITIVE
0039 0002            N    EQU     2                       NEGATIVE
0040 0000            EQ   EQU     0                       EQUAL
0041 0001            GT   EQU     1                       GREATER THAN
0042 0002            LT   EQU     2                       LESS THAN
0043 0003            UN   EQU     3                       UNCONDITIONAL
0044 0000            AL1  EQU     0                       ALL BITS ARE 1
0045 0002            NO   EQU     2                       NOT ALL BITS ARE 1
0046                 *******************************************************
0047                 *
0048                 * DEFINITION OF PROGRAM DEFINED SYMBOLS
0049 0000            POS  EQU     0         POSITIVE
0050 00F0            NEG  EQU     H'F0'     NEGATIVE.
0051 0007            LENG EQU     7         LENGTH OF OPERAND.
0052 0004            LEN  EQU     LENG-3    LENGTH OF MANTISSA.
0053 0066            OFST EQU     H'66'     OFFSET FOR ADDITION.
0054                 *
0056                 ******************************************************
0057                 *
0058                 * SCRATCH PAD AREA IN SAME PAGE AS PROGRAM
0059 0000                 ORG     H'6E0'
0060                 *
0061 06E0            PTR  RES     2         ADDRESS POINTER.
0062 06E2            CNTR RES     1         INDEX COUNTER
0063                 *
0064 06E3            OPA  RES     1         OPERAND A WORKAREA.
0065 06E4            EXPA RES     1
0066 06E5            SMAA RES     1
0067 06E6            MANA RES     LEN+1
0068                 *
0069 06EB            OPB  RES     1         OPERAND B WORKAREA.
0070 06EC            EXPB RES     1
0071 06ED            SMAB RES     1
0072 06EE            MANB RES     LEN+1
0073                 *
0074 06F3            EXT  RES     LEN+1     EXTENSION REGISTER
0075                 *
0076 06F8            EEXP RES     1         EXTENDED EXPONENT B.
0077                 *
0078                 ****************************************************
0079                 *
0080                 * OPERAND AREA - CAN BE OTHER PAGE THAN PROGRAM
0081 06F9                 ORG     H'700'
0082                 *
0083 0700            OP1  RES     LENG      OPERAND 1 STORAGE LOCATION.
0084 0707            OP2  RES     LENG      OPERAND 2 STORAGE LOCATION.
0085 070E            RSLT RES     LENG      RESULT STORAGE LOCATION.
0087                 *******************************************************
0088                 *
```

```
LINE ADDR  OBJECT  E SOURCE

0089                 * START OF PROGRAM
0090 0715                 ORG     H'440'
0091                 *                      POINTERS AND TABLES OF THE
0092                 *                        MAIN PROGRAM.
0093 0440 07         PTR1 DATA    <OP1      OPERAND 1 POINTER.
0094 0441 00              DATA    >OP1
0095 0442 07         PTR2 DATA    <OP2      OPERAND 2 POINTER.
0096 0443 07              DATA    >OP2
0097 0444 07         SPT2 DATA    <OP2      SIGN MANTISSA 2
0098 0445 09              DATA    >OP2+2
0099 0446 07         PTRR DATA    <RSLT     RESULT POINTER.
0100 0447 0E              DATA    >RSLT
0101 0448 B6         ROUN DATA    OFST+H'50' ROUNDING CONSTANT H'50'
0102                 *
0103                 **************************************************
0104                 *
0105                 * TRANSFER OPERANDS SUBROUTINE
0106                 *
0107                 *    TRANSFER OP1--> OPB WORKAREA, CLEAR EXTENDED REG.
0108                 *               OP2--> OPA WORKAREA,CLEAR ROUND BYTE.
0109 0449 0507       TRAN LODI,R1 LENG      LOAD INDEX COUNTER
0110 044B 0DC440     TRA1 LODA,R0 *PTR1,R1,- LOAD BYTE OPERAND 1
0111 044E CD66EB          STRA,R0 OPB,R1    STORE BYTE IN WORKAREA OPB.
0112 0451 0DE442          LODA,R0 *PTR2,R1  LOAD BYTE OPERAND 2
0113 0454 CD66E3          STRA,R0 OPA,R1    STORE BYTE IN WORKAREA OPA.
0114 0457 5972            BRNR,R1 TRA1      IF NOT READY BRANCH.
0115 0459 CD06EA          STRA,R1 MANA+LEN  CLEAR ROUND BYTE.
0116 045C 0507            LODI,R1 LENG      LOAD INDEX COUNTER.
0117 045E 20              EORZ    R0        CLEAR R0.
0118 045F CD46F2     TRA2 STRA,R0 MANB+LEN,R1,- CLEAR EXTENDED REGISTER.
0119 0462 597B            BRNR,R1 TRA2
0120 0464 17              RETC,UN
0121                 *
```

TRANSFER OPERANDS SUBROUTINE

Transfers operands OP1, OP2 to work area; clears rounding bytes and extended register.

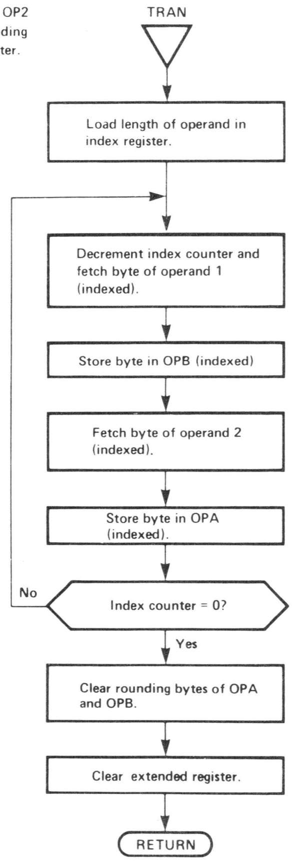

TRAN
Load length of operand in index register.
Decrement index counter and fetch byte of operand 1 (indexed).
Store byte in OPB (indexed)
Fetch byte of operand 2 (indexed).
Store byte in OPA (indexed).
No
Index counter = 0?
Yes
Clear rounding bytes of OPA and OPB.
Clear extended register.
RETURN

Figure 3

ADDITION AND SUBTRACTION ROUTINE

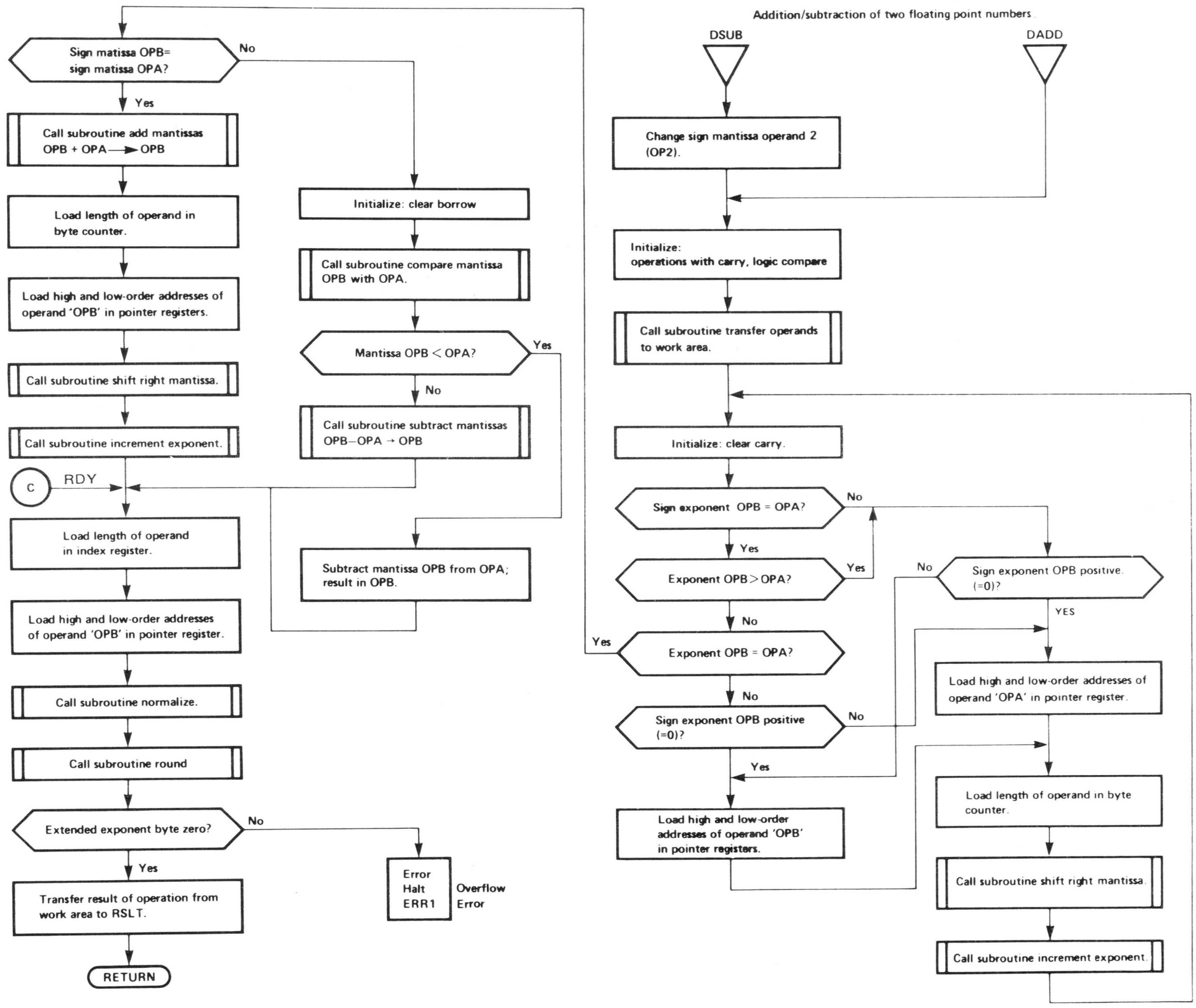

Addition/subtraction of two floating point numbers
DSUB
DADD
Change sign mantissa operand 2 (OP2).
Initialize: operations with carry, logic compare
Call subroutine transfer operands to work area.
Initialize: clear carry.
Sign exponent OPB = OPA?
No
Yes
Exponent OPB > OPA?
Yes
No
Exponent OPB = OPA?
Yes
No
Sign exponent OPB positive (=0)?
No
Yes
Load high and low-order addresses of operand 'OPB' in pointer registers.
Sign exponent OPB positive (=0)?
No
YES
Load high and low-order addresses of operand 'OPA' in pointer register.
Load length of operand in byte counter.
Call subroutine shift right mantissa.
Call subroutine increment exponent.
Sign matissa OPB= sign matissa OPA?
No
Yes
Call subroutine add mantissas OPB + OPA → OPB
Load length of operand in byte counter.
Load high and low-order addresses of operand 'OPB' in pointer registers.
Call subroutine shift right mantissa.
Call subroutine increment exponent.
C
RDY
Load length of operand in index register.
Load high and low-order addresses of operand 'OPB' in pointer register.
Call subroutine normalize.
Call subroutine round
Extended exponent byte zero?
No
Yes
Transfer result of operation from work area to RSLT.
RETURN
Initialize: clear borrow
Call subroutine compare mantissa OPB with OPA.
Mantissa OPB < OPA?
Yes
No
Call subroutine subtract mantissas OPB−OPA → OPB
Subtract mantissa OPB from OPA; result in OPB.
Error Halt ERR1
Overflow Error

Figure 4

```
LINE ADDR  OBJECT  E SOURCE

0123                 ************************************************************
0124                 *
0125                 * SUBTRACTION ROUTINE
0126                 *
0127                 * THIS ROUTINE SUBTRACTS TWO BCD FLOATING POINT NUMBERS
0128                 *
0129 0465 0F8444     DSUB LODA,R3 *SPT2      LOAD SIGN MANTISSA OP2.
0130 0468 27F0            EORI,R3 H'F0'      CHANGE SIGN.
0131 046A 1B03            BCTR,UN DA1
0132                 *
0133                 *************************************************************
0134                 *
0135                 * ADDITION ROUTINE
0136                 *
0137                 * THIS ROUTINE ADDS TWO BCD FLOATING POINT NUMBERS
0138                 *
0139 046C 0F8444     DADD LODA,R3 *SPT2      LOAD SIGN MANTISSA OP2.
0140 046F 770A       DA1  PPSL    WC+COM     WITH CARRY/LOGIC COMPARE.
0141 0471 3B56            BSTR,UN TRAN      *CALL SUBR. TRANSFER OPERANDS.
0142 0473 CF06E5          STRA,R3 SMAA       STORE SIGN IN OPERAND A.
0143 0476 7501       DA2  CPSL    C          CLEAR CARRY.
0144 0478 0D06EB          LODA,R1 OPB        LOAD SIGN EXP. B.
0145 047B ED06E3          COMA,R1 OPA        COMPARE WITH EXPA.
0146 047E 1816            BCTR,EQ DA6        IF EQUAL BRANCH ELSE TEST
0147 0480 590E       DA3  BRNR,R1 DA5          SIGN AND BRANCH IF NEG.
0148 0482 0406       DA7  LODI,R0 <OPA       LOAD HIGH AND LOW ORDER
0149 0484 05E3            LODI,R1 >OPA         ADDR. OPERAND A.
0150 0486 0607       DA4  LODI,R2 LENG       LOAD LENGTH INDEX.
0151 0488 3F0681          BSTA,UN SHFR      *CALL SUBR. SHIFT RIGHT.
0152 048B 3F069F          BSTA,UN INCR      *CALL SUBR. INCREMENT EXP.
0153 048E 1B66            BCTR,UN DA2
0154 0490 0406       DA5  LODI,R0 <OPB       LOAD HIGH AND LOW ORDER
0155 0492 05EB            LODI,R1 >OPB         ADDR. OPERAND B.
0156 0494 1B70            BCTR,UN DA4
0157 0496 0C06EC     DA6  LODA,R0 EXPB       LOAD EXP. B.
0158 0499 EC06E4          COMA,R0 EXPA       COMPARE WITH EXP. A.
0159 049C 1962            BCTR,GT DA3        BRANCH IF B>A.
0160 049E 1804            BCTR,EQ DAD0       BRANCH IF B=A.
0161 04A0 5960            BRNR,R1 DA7        IF NEG. EXP INCR OPERAND A.
0162 04A2 1B6C            BCTR,UN DA5          ELSE INCREMENT OPERAND B.
0163                 *
0164 04A4 0C06E5     DAD0 LODA,R0 SMAA       LOAD SIGN MANTISSA A.
0165 04A7 EF06ED          COMA,R3 SMAB       COMPARE WITH SIGN B.
0166 04AA 9810            BCFR,EQ DAD1       BRANCH IF NOT EQUAL.
0167 04AC 3B2C            BSTR,UN ADD       *CALL SUBR. ADDITION B=B+A.
0168 04AE 0608            LODI,R2 LENG+1     LOAD LENGTH OF SHIFT COUNTER.
0169 04B0 0406            LODI,R0 <OPB       LOAD HIGH AND LOW ORDER
0170 04B2 05EB            LODI,R1 >OPB         ADDR. OPERAND B.
0171 04B4 3F0681          BSTA,UN SHFR      *CALL SUBR SHIFT RIGHT.
0172 04B7 3F069F          BSTA,UN INCR      *CALL SUBR. INCREMENT EXP.
0173 04BA 1B1B            BCTR,UN RDY1       ADDITION READY: BRANCH.
0174 04BC 7701       DAD1 PPSL    C          CLEAR BORROW
```

```
0175 04BE 3B31           BSTR,UN COMP       *CALL SUBR. COMPARE A-B.
0176 04C0 1A04           BCTR,LT DAD2        IF B<A BRANCH ELSE
0177 04C2 3B3A           BSTR,UN SUB        *CALL SUBR. SUBTRACTION B=B-A.
0178 04C4 1B11           BCTR,UN RDY1        IF SUBTRACTION READY BRANCH.
0179 04C6 CF06ED    DAD2 STRA,R3 SMAB        STORE NEW SIGN EXP. IN B.
0180 04C9 0605           LODI,R2 LEN+1       LOAD SUBTRACTION COUNTER.
0181 04CB 0E46E6    DAD3 LODA,R0 MANA,R2,-   LOAD BYTE MANTISSA A.
0182 04CE AE66EE         SUBA,R0 MANB,R2     SUBTRACT BYTE MANTISSA B.
0183 04D1 94             DAR,R0              DECIMAL ADJUST RESULT
0184 04D2 CE66EE         STRA,R0 MANB,R2     STORE RESULT IN B.
0185 04D5 5A74           BRNR,R2 DAD3        IF NOT READY BRANCH
0186 04D7 1F0539    RDY1 BCTA,UN RDY         BRANCH TO NORMALIZE AND ROUN
0187                *

0189                *    ADDITION SUBROUTINE
0190                *    THIS SUBROUTINE PERFORMS THE ADDITION OF
0191                *      B+A -->B.
0192                *
0193 04DA 7501      ADD  CPSL    C           CLEAR CARRY.
0194 04DC 0500           LODI,R1 0           CLEAR INTERBYTE CARRY REG.
0195 04DE 0605           LODI,R2 LEN+1       LOAD INDEX COUNTER.
0196 04E0 51        ADD1 RRR,R1              SHIFT CARRY IN R1.
0197 04E1 0E46EE         LODA,R0 MANB,R2,-   LOAD MANTISSA BYTE B.
0198 04E4 8466           ADDI,R0 H'66'         AND ADD OFFSET.
0199 04E6 D1             RRL,R1              SHIFT INTERBYTE CARRY IN C.
0200 04E7 8E66E6         ADDA,R0 MANA,R2     ADD MANTISSA BYTE A TO B.
0201 04EA 94             DAR,R0              DEC. ADJ. ADDITION RESULT.
0202 04EB CE66EE         STRA,R0 MANB,R2     STORE RESULT IN MANTISSA B
```

```
0203 04EE 5A70           BRNR,R2 ADD1        IF NOT READY BRANCH.
0204 04F0 17             RETC,UN
0205                *
0206                *    COMPARE SUBROUTINE.
0207                *    THIS SUBROUTINE COMPARES B AND A.
0208                *    THE CC ARE SET TO THE PROPER VALUE.
0209                *
0210 04F1 06FB      COMP LODI,R2 -LEN-1      LOAD INDEX COUNTER.
0211 04F3 0E65F3    COM1 LODA,R0 MANB-256+LEN+1,R2 LOAD BYTE OF OPERAND B
0212 04F6 EE65EB         COMA,R0 MANA-256+LEN+1,R2 COMPARE WITH B
0213 04F9 9802           BCFR,EQ COM2        BRANCH IF NOT EQUAL.
0214 04FB DA76           BIRR,R2 COM1        IF NOT READY BRANCH.
0215 04FD 17        COM2 RETC,UN
0216                *
0217                *    SUBTRACT SUBROUTINE.
0218                *    THIS ROUTINE SUBTRACTS A FROM B AND THE RESULT
0219                *    IS PLACED IN B.  B-A=B
0220                *
0221 04FE 7701      SUB  PPSL    C           CLEAR BORROW
0222 0500 0605           LODI,R2 LEN+1       LOAD INDEX COUNTER.
0223 0502 0E46EE    SUB1 LODA,R0 MANB,R2,-   LOAD BYTE MANTISSA B
0224 0505 AE66E6         SUBA,R0 MANA,R2     SUBTRACT BYTE MANTISSA A.
0225 0508 94             DAR,R0              DECIMAL ADJUST RESULT.
0226 0509 CE66EE         STRA,R0 MANB,R2     STORE RESULT IN B
0227 050C 5A74           BRNR,R2 SUB1        IF NOT READY BRANCH.
0228 050E 17             RETC,UN
0229                *
```

ADD MANTISSA SUBROUTINE

Adds mantissas of OPB and OPA; result is placed in OPB.

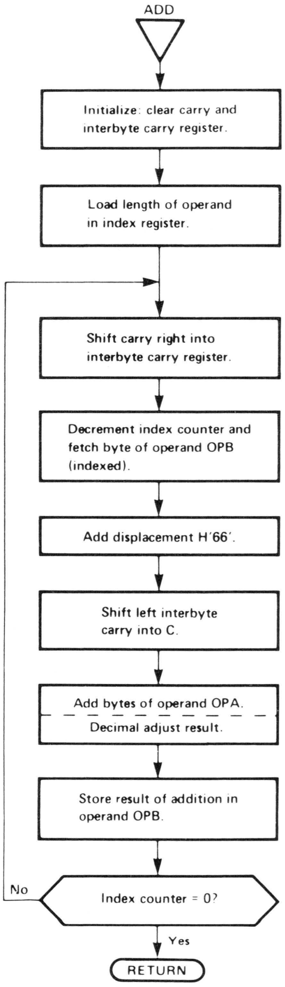

ADD
Initialize: clear carry and interbyte carry register.
Load length of operand in index register.
Shift carry right into interbyte carry register.
Decrement index counter and fetch byte of operand OPB (indexed).
Add displacement H'66'.
Shift left interbyte carry into C.
Add bytes of operand OPA.
Decimal adjust result.
Store result of addition in operand OPB.
No
Index counter = 0?
Yes
RETURN

Figure 5

COMPARE MANTISSA SUBROUTINE

Compares mantissa of OPB with OPA to set the condition codes CC.

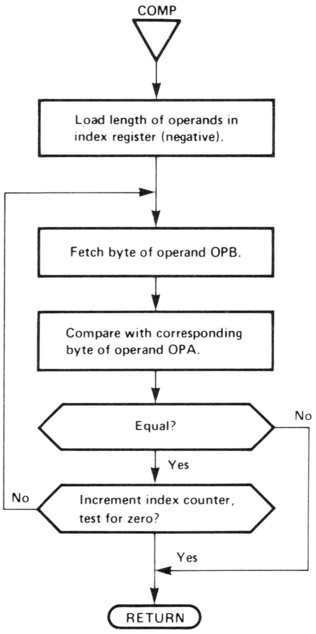

COMP
Load length of operands in index register (negative).
Fetch byte of operand OPB.
Compare with corresponding byte of operand OPA.
Equal?
No
Yes
No
Increment index counter, test for zero?
Yes
RETURN

Figure 6

SUBTRACT MANTISSA SUBROUTINE

Subracts mantissa OPA from OPB; result in OPB.

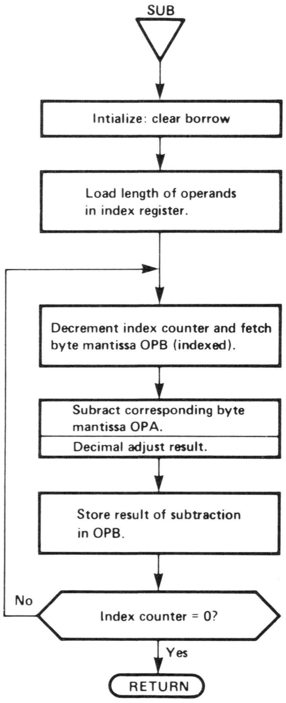

SUB
Intialize: clear borrow
Load length of operands in index register.
Decrement index counter and fetch byte mantissa OPB (indexed).
Subract corresponding byte mantissa OPA.
Decimal adjust result.
Store result of subtraction in OPB.
No
Index counter = 0?
Yes
RETURN

Figure 7

```
LINE ADDR  OBJECT  E SOURCE

0231                 ****************************************************
0232                 *
0233                 * THIS IS PART OF DIVISION ROUTINE - PLACED HERE FOR
0234                 *    ACCESS TO SUBROUTINES BY RELATIVE ADDRESSING
0235                 *
0236 050F 0709       DIV  LODI,R3 LEN+LEN+1  LOAD DIVISION COUNTER.
0237 0511 0500       DIV1 LODI,R1 0          CLEAR RESULT COUNTER.
0238 0513 3B5C       DIV2 BSTR,UN COMP      *CALL SUBR COMPARE A - B.
0239 0515 1A07            BCTR,LT DIV3       IF B<A BRANCH.
0240 0517 3B65            BSTR,UN SUB       *CALL SUBR SUBTRACTION B-A
0241 0519 8566            ADDI,R1 H'66'      INCREMENT RESULT COUNTER
0242 051B 95              DAR,R1             DECIMAL ADJUST RESULT
0243 051C 1B75            BCTR,UN DIV2
0244 051E 8D06F7     DIV3 ADDA,R1 MANB+LEN+LEN+1  ADD COUNTER TO LS BYTE.
0245 0521 CD06F7          STRA,R1 MANB+LEN+LEN+1  STORE RESULT IN EXT. REG
0246 0524 0406            LODI,R0 <OPB       LOAD HIGH AND LOW ORDER
0247 0526 05EB            LODI,R1 >OPB         ADDR OPERAND B.
0248 0528 060D            LODI,R2 LENG+LEN+2 LOAD LENGTH INDEX.
0249 052A 3F0591          BSTA,UN SHFL      *CALL SUBR.  SHIFT LEFT.
0250 052D FB62            BDRR,R3 DIV1       IF NOT READY BRANCH.
0251 052F 3F069F          BSTA,UN INCR      *CALL SUBR INCREMENT EXP.
0252 0532 0709            LODI,R3 LEN+LEN+1  LOAD SHIFT COUNTER.
0253 0534 3F059A     DIV4 BSTA,UN SFL       *CALL SUBR SHIFT LEFT.
0254 0537 FB7B            BDRR,R3 DIV4
0255                 *                         NORMALIZE AND TRANSFER.
0256 0539 0406       RDY  LODI,R0 <OPB       LOAD HIGH AND LOW ORDER
0257 053B 05EB            LODI,R1 >OPB         ADDR.  OPERAND B.
0258 053D 0609            LODI,R2 LENG+2     LOAD LENGTH INDEX REGISTER.
0259 053F 3B14            BSTR,UN NORM      *CALL NORMALIZE SUBR.
0260 0541 3F0669          BSTA,UN RND       *CALL ROUNDING SUBR.
0261 0544 0C06F8          LODA,R0 EEXP       LOAD EXTENDED EXPONENT.
0262 0547 980B            BCFR,Z  ERR1       IF EXT.  EXP.  NOT ZERO BRANCH.
0263                 *                       TRANSFER RESULT
0264 0549 0707            LODI,R3 LENG       LOAD INDEX LENGTH OF REGISTER
0265 054B 0F46EB     RDY2 LODA,R0 OPB,R3,-   LOAD OPERAND B BYTE
0266 054E CFE446          STRA,R0 *PTRR,R3   TRANSFER TO RESULT LOCATION
0267 0551 5B78            BRNR,R3 RDY2
0268 0553 17              RETC,UN
0269 0554 40         ERR1 HALT               OVERFLOW HALT.
0270                 *

0272                 *    NORMALIZE SUBROUTINE.
0273                 *
0274 0555 CE06E2     NORM STRA,R2 CNTR       STORE INDEX REGISTER.
0275 0558 CC06E0     NRM0 STRA,R0 PTR        STORE HIGH AND LOW ORDER
0276 055B CD06E1          STRA,R1 PTR+1        ADDR OPERAND.
0277 055E 0700       NRM  LODI,R3 0          CLEAR ZERO DIGIT COUNTER.
0278 0560 0502            LODI,R1 2          LOAD INDEX COUNTER.
0279 0562 7501            CPSL    C          CLEAR CARRY.
0280 0564 0DA6E0     NRM1 LODA,R0 *PTR,R1,+  LOAD MANTISSA BYTE.
0281 0567 9819            BCFR,Z  NRM2
0282 0569 8702            ADDI,R3 2          UPDATE ZERO DIGIT COUNTER.
0283 056B E507            COMI,R1 LENG       BRANCH IF END OF MANTISSA
0284 056D 9875            BCFR,EQ NRM1         NOT REACHED ELSE UPDATE
0285 056F C1              STRZ    R1           FORMAT TO REPRESENT ZERO.
0286 0570 04F0            LODI,R9 H'F0'      NEG SIGN -> SIGN EXP.
0287 0572 CC86E0          STRA,R0 *PTR
0288 0575 0499            LODI,R0 H'99'      MAX -> EXP
0289 0577 CDA6E0          STRA,R0 *PTR,R1,+
0290 057A 20              EORZ    R0         POS -> SIGN MANTISSA.
0291 057B CDA6E0          STRA,R0 *PTR,R1,+
0292 057E CC06F8          STRA,R0 EEXP       CLEAR EXTENDED REGISTER.
0293 0581 17              RETC,UN
0294                 *
0295 0582 44F0       NRM2 ANDI,R0 H'F0'      TEST MS DIGIT OF MANTISSA
0296 0584 9802            BCFR,Z  NRM3         BYTE.
0297 0586 8701            ADDI,R3 1          UPDATE ZERO COUNTER.
0298 0588 03         NRM3 LODZ    R3         UPDATE CONDITION CODES.
0299 0589 14              RETC,Z             BRANCH IF (R3) ZERO.
0300 058A 3B0E       NRM4 BSTR,UN SFL       *CALL SUBR.  SHIFT LEFT.
0301 058C 3B21            BSTR,UN DECR      *CALL SUBR.  DECREMENT EXP.
0302 058E FB7A            BDRR,R3 NRM4       DECR.  ZERO DIGIT COUNTER.
0303 0590 17              RETC,UN
0304                 *
0305                 *    SHIFT LEFT SUBROUTINE
0306                 *    THIS SUBR.  SHIFTS OPERAND ONE DIGIT LEFT.
0307                 *      EXTENDED REGISTER INCLUDED
0308                 *      CARRY MUST BE ZERO
0309                 *
```

NORMALIZE SUBROUTINE

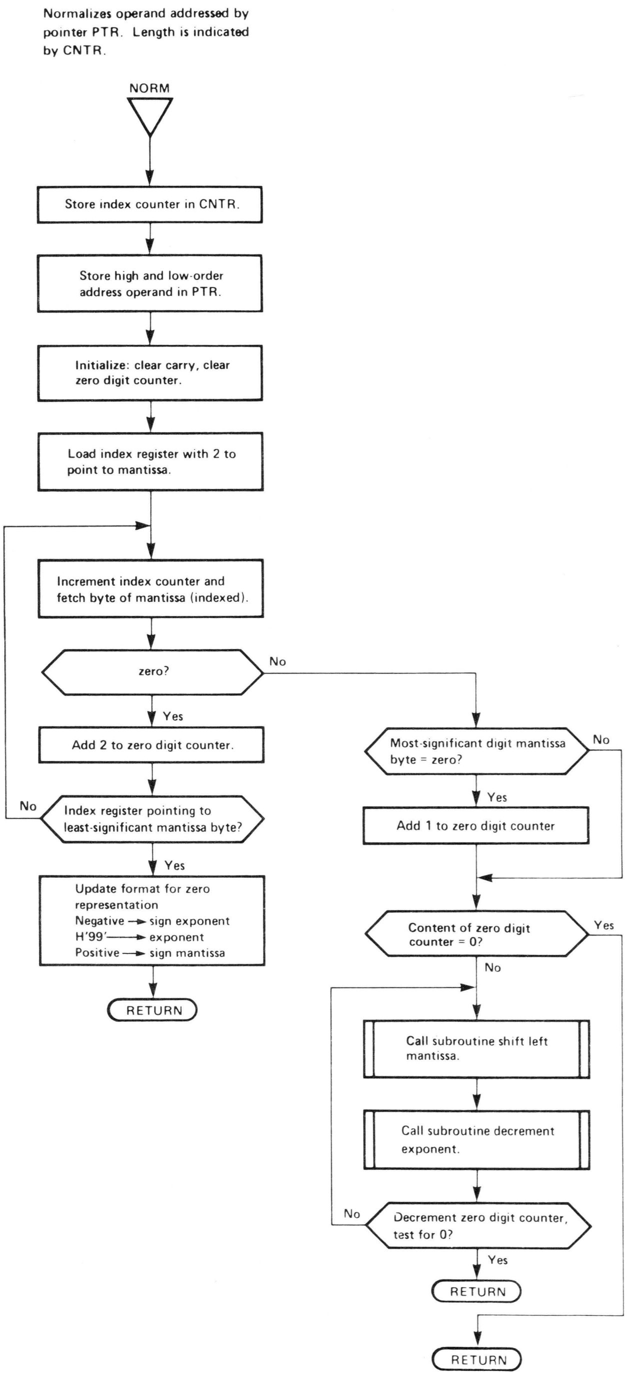

Normalizes operand addressed by pointer PTR. Length is indicated by CNTR.
NORM
Store index counter in CNTR.
Store high and low-order address operand in PTR.
Initialize: clear carry, clear zero digit counter.
Load index register with 2 to point to mantissa.
Increment index counter and fetch byte of mantissa (indexed).
zero?
No
Yes
Add 2 to zero digit counter.
No
Index register pointing to least-significant mantissa byte?
Yes
Update format for zero representation
Negative → sign exponent
H'99' → exponent
Positive → sign mantissa
RETURN
Most-significant digit mantissa byte = zero?
No
Yes
Add 1 to zero digit counter
Content of zero digit counter = 0?
Yes
No
Call subroutine shift left mantissa.
Call subroutine decrement exponent.
No
Decrement zero digit counter, test for 0?
Yes
RETURN
RETURN

Figure 8

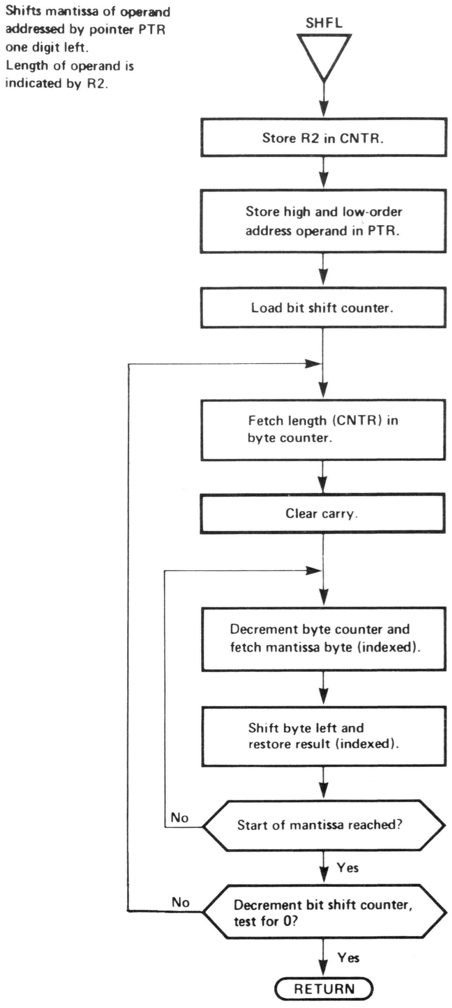
```
LINE ADDR  OBJECT  E SOURCE

0310 0591 CE06E2      SHFL STRA,R2 CNTR       STORE BYTE SHIFT COUNTER.
0311 0594 CC06E0      SFL0 STRA,R0 PTR        STORE HIGH AND LOW ORDER
0312 0597 CD06E1           STRA,R1 PTR+1        ADDR. OPERAND.
0313 059A 0504        SFL  LODI,R1 4          LOAD BIT SHIFT COUNTER.
0314 059C 0E06E2      SFL1 LODA,R2 CNTR       LOAD BYTE SHIFT COUNTER.
0315 059F 7501             CPSL    C
0316 05A1 0EC6E0      SFL2 LODA,R0 *PTR,R2,-  LOAD BYTE OF MANTISSA.
0317 05A4 D0               RRL,R0             ROTATE LEFT
0318 05A5 CEE6E0           STRA,R0 *PTR,R2    STORE RESULT.
0319 05A8 E603             COMI,R2 3          COMPARE INDEX WITH START OF
0320 05AA 9875             BCFR,EQ SFL2        REGISTER,BRANCH IF NOT READY
0321 05AC F96E             BDRR,R1 SFL1       IF NOT READY BRANCH.
0322 05AE 17               RETC,UN
0323                  *
0324                  *    DECREMENT EXPONENT OPERAND
0325                  *    THE ADDRESS OF THE OPERAND IS IN PTR
0326                  *    (R1) MUST BE ZERO.
0327                  *
0328 05AF 0DA6E0      DECR LODA,R0 *PTR,R1,+  LOAD EXPONENT OPERAND.
0329 05B2 0E86E0           LODA,R2 *PTR       LOAD SIGN EXPONENT.
```

```
LINE ADDR  OBJECT  E SOURCE

0330 05B5 1809             BCTR,Z  DEC3       IF POSITIVE BRANCH.
0331 05B7 8467        DEC1 ADDI,R0 H'67'      ADD 1 TO EXPONENT.
0332 05B9 94               DAR,R0             DECIMAL ADJUST RESULT.
0333 05BA 181A             BCTR,Z  ERR2       IF (R0)=0 BRANCH TO ERROR.
0334 05BC CDE6E0      DEC2 STRA,R0 *PTR,R1    STORE RESULT IN EXPONENT.
0335 05BF 17               RETC,UN
0336                  *
0337 05C0 580F        DEC3 BRNR,R0 DEC5       IF EXPONENT NOT ZERO BRANCH.
0338 05C2 EC06F8           COMA,R0 EEXP       COMPARE EXTENDED EXPONENT
0339 05C5 9807             BCFR,EQ DEC4       WITH 0, IF NOT EQUAL BRANCH.
0340 05C7 06F0             LODI,R2 H'F0'      NEGATIVE SIGN.
0341 05C9 CE86E0           STRA,R2 *PTR       STORE NEG. SIGN IN OPERAND.
0342 05CC 1B69             BCTR,UN DEC1
0343                  *
0344 05CE CC06F8      DEC4 STRA,R0 EEXP       CLEAR EXTENDED EXPONENT.
0345 05D1 A400        DEC5 SUBI,R0 0          SUBTRACT 1 FROM EXPONENT.
0346 05D3 94               DAR,R0             DECIMAL ADJUST.
0347 05D4 1B66             BCTR,UN DEC2
0348                  *
0349 05D6 40          ERR2 HALT               UNDERFLOW HALT.
```

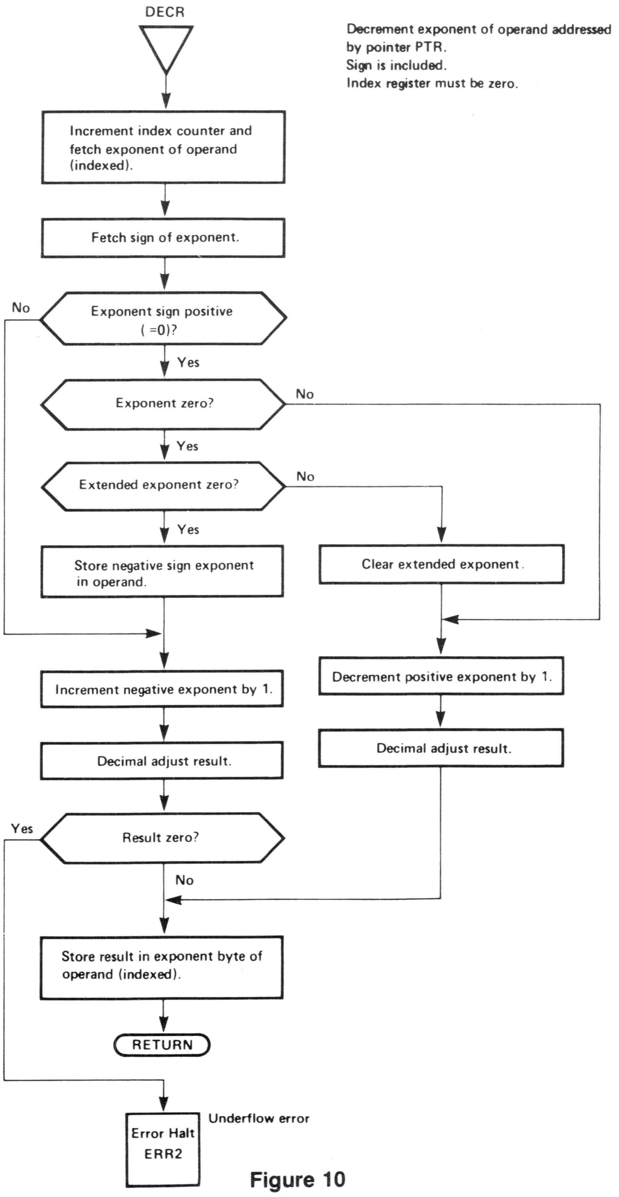
SHIFT MANTISSA LEFT SUBROUTINE


Shifts mantissa of operand addressed by pointer PTR one digit left.
Length of operand is indicated by R2.

SHFL
Store R2 in CNTR.
Store high and low-order address operand in PTR.
Load bit shift counter.
Fetch length (CNTR) in byte counter.
Clear carry.
Decrement byte counter and fetch mantissa byte (indexed).
Shift byte left and restore result (indexed).
No
Start of mantissa reached?
Yes
No
Decrement bit shift counter, test for 0?
Yes
RETURN

Figure 9

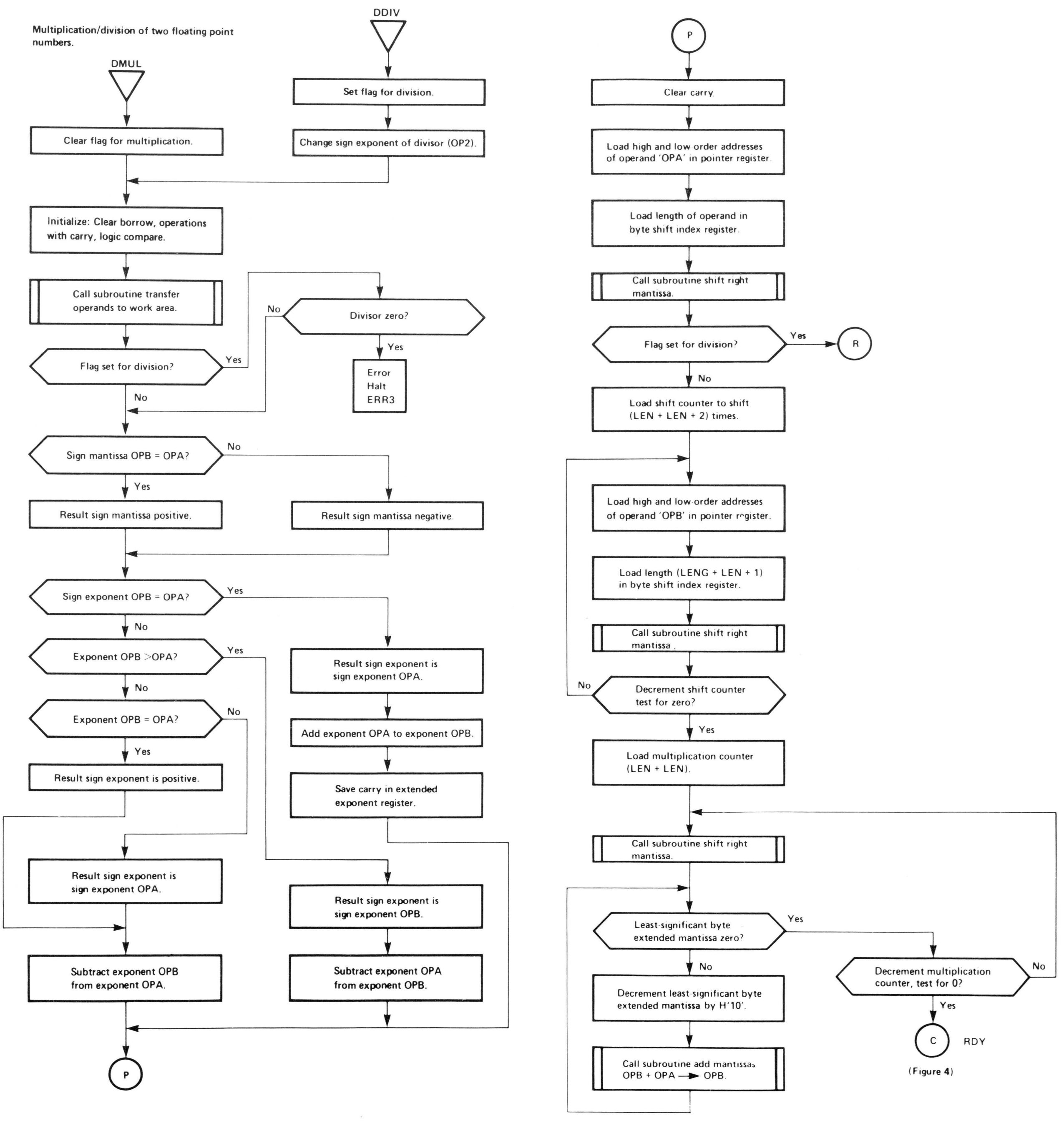
DECREMENT EXPONENT SUBROUTINE


Decrement exponent of operand addressed by pointer PTR.
Sign is included.
Index register must be zero.

DECR
Increment index counter and fetch exponent of operand (indexed).
Fetch sign of exponent.
No
Exponent sign positive (=0)?
Yes
Exponent zero?
No
Yes
Extended exponent zero?
No
Yes
Store negative sign exponent in operand.
Clear extended exponent.
Increment negative exponent by 1.
Decrement positive exponent by 1.
Decimal adjust result.
Decimal adjust result.
Yes
Result zero?
No
Store result in exponent byte of operand (indexed).
RETURN
Error Halt ERR2
Underflow error

Figure 10

MULTIPLICATION AND DIVISION ROUTINE



Multiplication/division of two floating point numbers.
DMUL
Clear flag for multiplication.
DDIV
Set flag for division.
Change sign exponent of divisor (OP2).
Initialize: Clear borrow, operations with carry, logic compare.
Call subroutine transfer operands to work area.
Flag set for division?
Yes
No
Divisor zero?
No
Yes
Error Halt ERR3
Sign mantissa OPB = OPA?
No
Yes
Result sign mantissa positive.
Result sign mantissa negative.
Sign exponent OPB = OPA?
Yes
No
Exponent OPB >OPA?
Yes
No
Exponent OPB = OPA?
No
Yes
Result sign exponent is positive.
Result sign exponent is sign exponent OPA.
Add exponent OPA to exponent OPB.
Save carry in extended exponent register.
Result sign exponent is sign exponent OPA.
Result sign exponent is sign exponent OPB.
Subtract exponent OPB from exponent OPA.
Subtract exponent OPA from exponent OPB.
P
P
Clear carry.
Load high and low-order addresses of operand 'OPA' in pointer register.
Load length of operand in byte shift index register.
Call subroutine shift right mantissa.
Flag set for division?
Yes
R
No
Load shift counter to shift (LEN + LEN + 2) times.
Load high and low-order addresses of operand 'OPB' in pointer register.
Load length (LENG + LEN + 1) in byte shift index register.
Call subroutine shift right mantissa.
No
Decrement shift counter test for zero?
Yes
Load multiplication counter (LEN + LEN).
Call subroutine shift right mantissa.
Least-significant byte extended mantissa zero?
Yes
No
Decrement least-significant byte extended mantissa by H'10'.
Call subroutine add mantissas OPB + OPA ⟶ OPB.
Decrement multiplication counter, test for 0?
No
Yes
C
RDY
(Figure 4)

Figure 11

MULTIPLICATION AND DIVISION ROUTINE Cont.

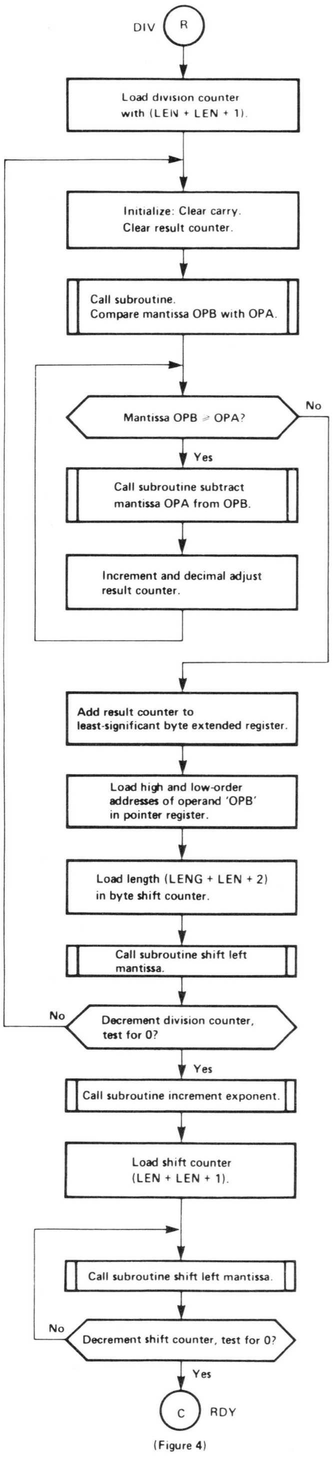

DIV
R
Load division counter with (LEN + LEN + 1).
Initialize: Clear carry. Clear result counter.
Call subroutine. Compare mantissa OPB with OPA.
Mantissa OPB ≥ OPA?
No
Yes
Call subroutine subtract mantissa OPA from OPB.
Increment and decimal adjust result counter.
Add result counter to least-significant byte extended register.
Load high and low-order addresses of operand 'OPB' in pointer register.
Load length (LENG + LEN + 2) in byte shift counter.
Call subroutine shift left mantissa.
No
Decrement division counter, test for 0?
Yes
Call subroutine increment exponent.
Load shift counter (LEN + LEN + 1).
Call subroutine shift left mantissa.
No
Decrement shift counter, test for 0?
Yes
C
RDY
(Figure 4)

Figure 11 A

```
ADDR  OBJECT  E SOURCE

                   *
05D7 40            ERR3 HALT
                   *
                   ******************************************************
                   *
                   * DIVISION ROUTINE
                   *
                   * THIS ROUTINE DIVIDES TWO BCD FLOATING POINT NUMBERS
                   *
05D8 0780          DDIV LODI,R3 H'80'      LOAD R3 (DIV FLAG).
05DA 0E8442             LODA,R2 *PTR2      LOAD SIGN EXP2.
05DD 26F0               EORI,R2 H'F0'        CHANGE SIGN
05DF 1B05               BCTR,UN DM1
                   *
                   ******************************************************
                   *
                   * MULTIPLICATION ROUTINE
                   *
                   * THIS ROUTINE MULTIPLIES TWO BCD FLOATING POINT NUMBERS
                   *
05E1 0700          DMUL LODI,R3 0          CLEAR R3 (MUL FLAG)
05E3 0E8442             LODA,R2 *PTR2      LOAD SIGN EXP2.
05E6 770B          DM1  PPSL    WC+COM+C   INITIALIZE: WITH CARRY
                   *                         LOGIC COMPARE, CLEAR BORROW
05E8 3F0449             BSTA,UN TRAN      *CALL SUBR. TRANSFER.
05EB 03                 LODZ    R3         UPDATE CONDITION CODES.
05EC 1805               BCTR,Z  DM2        IF NOT ZERO(DIV) TEST FIRST
05EE 0C06E6             LODA,R0 MANA         BYTE MANTISSA A.
05F1 1864               BCTR,Z  ERR3       IF ZERO DIV. ERROR (3)
05F3 0C06ED        DM2  LODA,R0 SMAB       LOAD SIGN MANTISSA B.
05F6 EC06E5             COMA,R0 SMAA       COMPARE WITH SIGN A.
05F9 1804               BCTR,EQ DM3        IF EQUAL BRANCH FOR POS. SIGN
05FB 04F0               LODI,R0 H'F0'        ELSE LOAD NEG. SIGN.
05FD 1B01               BCTR,UN DM4
05FF 20            DM3  EORZ    R0         CLEAR R0 (POS. SIGN MANTISSA)
0600 CC06ED        DM4  STRA,R0 SMAB       STORE R0 (SIGN) IN SIGN MAN B
                   *
0603 EE06EB             COMA,R2 OPB        COMPARE SIGN EXPA AND EXPB.
0606 980C               BCFR,EQ DMU
0608 0C06E4             LODA,R0 EXPA       LOAD EXP. A
060B 8465               ADDI,R0 H'65'      ADD OFFSET (C=1)
060D 8C06EC             ADDA,R0 EXPB       ADD EXPB+A
0610 94                 DAR,R0             DEC. ADJUST RESULT.
0611 D1                 RRL,R1             SHIFT CARRY IN R1
0612 1B1E               BCTR,UN DMU4
0614 0C06EC        DMU  LODA,R0 EXPB       LOAD EXPONENT B.
0617 EC06E4             COMA,R0 EXPA       COMPARE WITH EXP. A.
061A 190C               BCTR,GT DMU2       BRANCH IF GREATER B-A
061C 9802               BCFR,EQ DMU1       IF NOT EQUAL A-B
061E 0600               LODI,R2 0          LOAD POS SIGN INDICATOR.
0620 0C06E4        DMU1 LODA,R0 EXPA       LOAD EXP. A FOR SUBTRACTION.
0623 AC06EC             SUBA,R0 EXPB
0626 1B06               BCTR,UN DMU3
0628 0E06EB        DMU2 LODA,R2 OPB        LOAD SIGN OF OPERAND B IN R2.
062B AC06E4             SUBA,R0 EXPA       SUBTRACT  EXP A FROM B.
062E 94            DMU3 DAR,R0             DEC. ADJUST RESULT.
062F CE06EB             STRA,R2 OPB        STORE RESULT OF SIGN EXP.
0632 CC06EC        DMU4 STRA,R0 EXPB       STORE RESULT EXP.
0635 CD06F8             STRA,R1 EEXP       STORE CARRY FROM ADDITION
0638 7501               CPSL    C          CLEAR CARRY.
063A 0406               LODI,R0 <OPA       LOAD HIGH AND LOW ORDER
063C 05E3               LODI,R1 >OPA         ADDR OPERAND A.
063E 0607               LODI,R2 LENG       LOAD SHIFT LENGTH INDEX.
0640 3B3F               BSTR,UN SHFR      *CALL SUBR. SHIFT RIGHT
0642 5F050F             BRNA,R3 DIV        IF R3 NOT ZERO DIVISION ELSE
                   *                         MULTIPLICATION.
0645 070A          MUL  LODI,R3 LEN+LEN+2  LOAD SHIFT COUNTER.
0647 0406          MUL1 LODI,R0 <OPB       LOAD HIGH AND LOW ORDER
0649 05EB               LODI,R1 >OPB         ADDR OPERAND B.
064B 060C               LODI,R2 LENG+LEN+1 LOAD SHIFT LENGTH INDEX.
064D 3B32               BSTR,UN SHFR      *CALL SUBR SHIFT RIGHT.
064F FB76               BDRR,R3 MUL1       IF NOT READY BRANCH.
0651 0708               LODI,R3 LEN+LEN    LOAD MULTIPLICATION COUNTER.
0653 3B35          MUL2 BSTR,UN SHF       *CALL SUBR SHIFT RIGHT.
0655 0C06F7        MUL3 LODA,R0 MANB+LEN+LEN+1 LOAD LS BYTE EXT. MAN.
0658 180A               BCTR,Z  MUL4       IF ZERO BRANCH.
065A A40F               SUBI,R0 H'0F'      SUBTRACT H'10' FROM LS BYTE
065C CC06F7             STRA,R0 MANB+LEN+LEN+1 STORE RESULT IN B.
065F 3F04DA             BSTA,UN ADD       *CALL SUBR. ADDITION.
0662 1B71               BCTR,UN MUL3
                   *
0664 FB6D          MUL4 BDRR,R3 MUL2       IF NOT READY BRANCH.
                   *                         GO TO NORMALIZE.
0666 1F0539        MUDI BCTA,UN RDY        MULTIPLICATION/DIVISION READY
                   *
                   *
```

```
LINE ADDR  OBJECT  E SOURCE

0437                 *    ROUNDING OF OPERAND B (OPTIONAL).
0438                 *
0439 0669 0C0448     RND  LODA,R0 ROUN       LOAD ROUNDING CONSTANT.
0440 066C 0505            LODI,R1 LEN+1      LOAD INDEX COUNTER.
0441 066E 8D46EE     RND1 ADDA,R0 MANB,R1,-  ADD BYTE MANTISSA B.
0442 0671 94              DAR,R0             DECIMAL ADJUST RESULT.
0443 0672 CD66EE          STRA,R0 MANB,R1    STORE RESULT IN MANB.
0444 0675 0466            LODI,R0 OFST       LOAD R0 WITH OFFSET TO ADD C.
0445 0677 5975            BRNR,R1 RND1
0446 0679 B501            TPSL    C
0447 067B 16              RETC,NO             IF NO CARRY RETURN.
0448 067C 3B0C            BSTR,UN SHF        *CALL SUBR SHIFT RIGHT.
0449 067E 3B1F            BSTR,UN INCR       *CALL SUBR INCREMENT EXP.
0450 0680 17              RETC,UN
0451                 *
0452                 *    SHIFT RIGHT SUBROUTINE.
0453                 *    THIS SUBROUTINE SHIFTS A REGISTER DEFINED
0454                 *     BY PTR ONE DIGIT RIGHT
0455                 *
0456 0681 CE06E2     SHFR STRA,R2 CNTR       STORE END OF INDEX
0457 0684 CC06E0     SHF0 STRA,R0 PTR        STORE HIGH AND LOW ORDER
0458 0687 CD06E1          STRA,R1 PTR+1        ADDRESS OF OPERAND
0459 068A 0504       SHF  LODI,R1 4          LOAD BIT SHIFT COUNTER
0460 068C 0602       SHF1 LODI,R2 2          LOAD START OF INDEX.
0461 068E 0EA6E0     SHF2 LODA,R0 *PTR,R2,+  LOAD BYTE OF OPERAND AND
0462 0691 50              RRR,R0               ROTATE RIGHT.
0463 0692 CEE6E0          STRA,R0 *PTR,R2    STORE RESULT IN OPERAND.
0464 0695 EE06E2          COMA,R2 CNTR       COMPARE COUNTER WITH END OF
```

```
LINE ADDR  OBJECT  E SOURCE

0465 0698 9874            BCFR,EQ SHF2         INDEX, BRANCH IF NOT READY.
0466 069A 7501            CPSL    C          CLEAR CARRY
0467 069C F96E            BDRR,R1 SHF1       DECREMENT  BIT COUNTER.
0468 069E 17              RETC,UN
0469                 *
0470                 *    INCREMENT EXPONENT OF OPERAND SUBROUTINE.
0471                 *    CONDITION: (R1)=0 AND CARRY C=0.
0472                 *    THIS SUBROUTINE INCREMENTS THE EXPONENT OF AN
0473                 *    OPERAND INCLUDING SIGN.
0474                 *
0475 069F 0DA6E0     INCR LODA,R0 *PTR,R1,+  LOAD EXPONENT OF OPERAND
0476 06A2 0E86E0          LODA,R2 *PTR       LOAD SIGN EXPONENT.
0477 06A5 980E            BCFR,Z  INC2       BRANCH IF SIGN NEGATIVE.
0478 06A7 8467            ADDI,R0 H'67'      INCREMENT EXPONENT.
0479 06A9 94              DAR,R0             DECIMAL ADJUST.
0480 06AA 9805            BCFR,Z  INC1       BRANCH IF NO OVERFLOW.
0481 06AC 0501            LODI,R1 1          IF OVERFLOW PUT 1 IN EXT.  EXP
0482 06AE CD06F8          STRA,R1 EEXP
0483 06B1 CDE6E0     INC1 STRA,R0 *PTR,R1    STORE RESULT OF EXPONENT.
0484 06B4 17              RETC,UN
0485 06B5 5803       INC2 BRNR,R0 INC3       IF EXP.  NOT ZERO BRANCH.
0486 06B7 CC06F8          STRA,R0 EEXP       CLEAR EXTENDED EXP.
0487 06BA A400       INC3 SUBI,R0 0          DECREMENT NEGATIVE EXPONENT.
0488 06BC 94              DAR,R0
0489 06BD 5872            BRNR,R0 INC1       IF NOT ZERO BRANCH ELSE
0490 06BF CC86E0          STRA,R0 *PTR        POSITIVE SIGN IN SIGN BYTE.
0491 06C2 1B6D            BCTR,UN INC1
0492                 *
```

ROUNDING SUBROUTINE

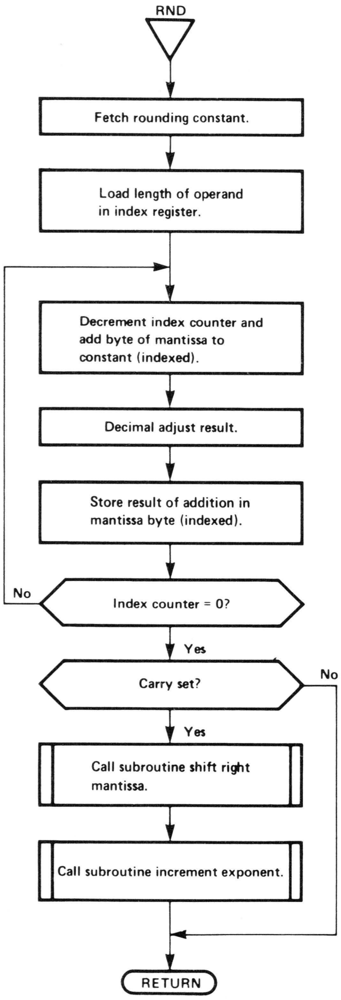

RND
Fetch rounding constant.
Load length of operand in index register.
Decrement index counter and add byte of mantissa to constant (indexed).
Decimal adjust result.
Store result of addition in mantissa byte (indexed).
No
Index counter = 0?
Yes
Carry set?
No
Yes
Call subroutine shift right mantissa.
Call subroutine increment exponent.
RETURN

Figure 12

SHIFT MANTISSA RIGHT SUBROUTINE

Shifts mantissa of operand addressed by pointer PTR one digit right.
Length of operand is indicated by R2.

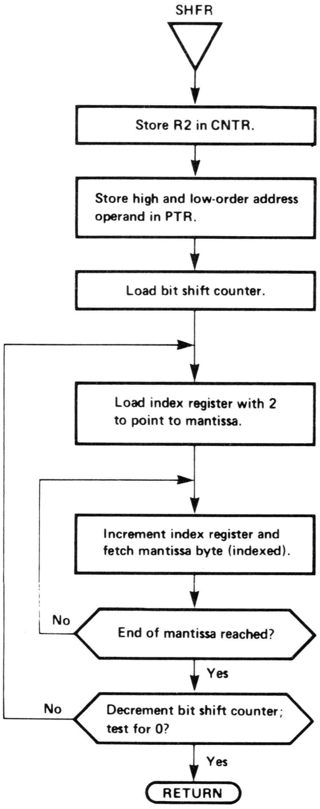

SHFR
Store R2 in CNTR.
Store high and low-order address operand in PTR.
Load bit shift counter.
Load index register with 2 to point to mantissa.
Increment index register and fetch mantissa byte (indexed).
No
End of mantissa reached?
Yes
No
Decrement bit shift counter; test for 0?
Yes
RETURN

Figure 13

INCREMENT EXPONENT SUBROUTINE

Increment exponent of operand addressed by pointer PTR.
Sign included.
Index register and carry must be zero.

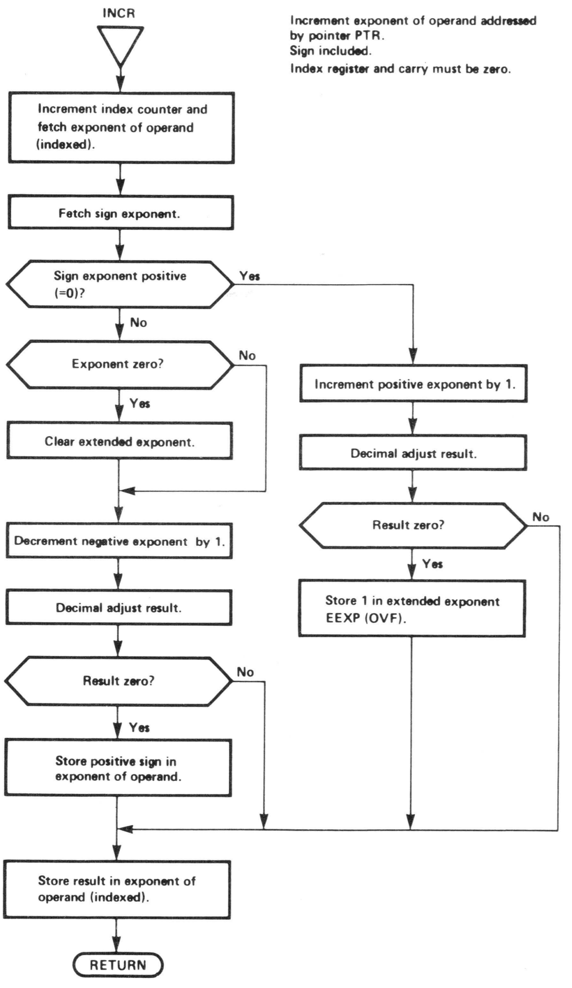

INCR
Increment index counter and fetch exponent of operand (indexed).
Fetch sign exponent.
Sign exponent positive (=0)?
Yes
No
Exponent zero?
No
Yes
Clear extended exponent.
Increment positive exponent by 1.
Decimal adjust result.
Result zero?
No
Yes
Store 1 in extended exponent EEXP (OVF).
Decrement negative exponent by 1.
Decimal adjust result.
Result zero?
No
Yes
Store positive sign in exponent of operand.
Store result in exponent of operand (indexed).
RETURN

Figure 14

```
LINE ADDR  OBJECT  E SOURCE

                   ****************************************************
                   *
                   * BCD FLOATING POINT ARITHMETIC TEST ROUTINE
                   *         FOR USE WITH PIPBUG
                   *
                   * THIS ROUTINE INPUTS AND OUTPUTS THE OPERANDS, THE
                   * OPERATOR AND THE RESULT. IT DETECTS THE OPERATOR
                   * AND CALLS THE ARITHMETIC ROUTINE.
                   *
                   *   INPUT SEQUENCE:
                   *           1' INPUT OPERAND 1
                   *           2' INPUT THE OPERATION CHAR '+','-','*', OR ':'
                   *           3' INPUT OPERAND 2
                   *
                   ****************************************************
                   *
                   * DEFINITIONS OF PROGRAM DEFINED SYMBOLS
                   *
     0020          SP'CE EQU    H'20'    SPACE CHARACTER
     008A          CRLF EQU    H'008A'  PIPBUG CR AND LF OUTPUT ROUTINE
     02B4          COUT EQU    H'02B4'  PIPBUG CHARACTER UUTPUT ROUTINE
     0286          CHIN EQU    H'0286'  PIPBUG CHARACTER INPUT ROUTINE
     0269          BOUT EQU    H'0269'  PIPBUG 2 HEX DIGITS OUTPUT ROUTINE
     0224          BIN  EQU    H'0224'  PIPBUG 2 HEX DIGITS INPUT ROUTINE
                   *
     06C4 7509     TEST CPSL    WC+C       OPERATION WITHOUT CARRY/
                   *                         CLEAR CARRY.
     06C6 7640          PPSU    F
     06C8 06FF          LODI,R2 -1         LOAD INDEX REGISTER.
     06CA 3B34          BSTR,UN IN11       INPUT SIGN EXPONENT.
     06CC CEA440        STRA,R0 *PTR1,R2,+ STORE IN OP1
     06CF 3F0224        BSTA,UN BIN        INPUT EXPONENT.
     06D2 01            LODZ    R1         SAVE IN R0.
     06D3 CEA440        STRA,R0 *PTR1,R2,+ STORE IN OP1
     06D6 3F0269        BSTA,UN BOUT       ECHO EXPONENT.
     06D9 3B25          BSTR,UN IN11       INPUT SIGN MANTISSA.
     06DB CEA440        STRA,R0 *PTR1,R2,+ STORE IN OP1.
     06DE 3F0224   TST1 BSTA,UN BIN        INPUT MANTISSA BYTE.
     06E1 01            LODZ    R1
     06E2 CEA440        STRA,R0 *PTR1,R2,+ STORE IN OP1.
     06E5 3F0269        BSTA,UN BOUT       ECHO MANTISSA BYTE.
     06E8 E606          COMI,R2 LENG-1     END OF MANTISSA REACHED.
     06EA 9872          BCFR,EQ TST1
                   *
     06EC 3B21          BSTR,UN SPAC       PRINT SPACE.
     06EE 3F0286        BSTA,UN CHIN       INPUT OPERATION CHARACTER.
     06F1 0700          LODI,R3 0          CLEAR R3.
     06F3 0504          LODI,R1 4          SET INDEX
     06F5 ED478E   TST2 COMA,R0 SRCH,R1,-  COMPARE CHAR. WITH LIST.
     06F8 181B          BCTR,EQ TST3       IF CHAR. FOUND BRANCH ELSE
     06FA 8703          ADDI,R3 3            ADD DISPLACEMENT.
     06FC 5977          BRNR,R1 TST2
     06FE 1B44          BCTR,UN TEST       ERROR RETURN.
                   *
```

PROGRAM TITLE

TEST ROUTINE FOR BCD ARITHMETIC FLOATING POINT ROUTINES

FUNCTION

Inputs and echoes operands and operator and
outputs the result of the operation via a teletype.

SPECIAL REQUIREMENTS

Hardware: Terminal and PC1001
Software: PIPBUG (PC1001)

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

RAM REQUIRED (BYTES): 1

ROM REQUIRED (BYTES): 205

EXECUTION TIME: Variable

MAXIMUM SUBROUTINE
NESTING LEVELS: 3

ASSEMBLER/COMPILER USED: TWIN VER 2.0

```
LINE ADDR  OBJECT  E SOURCE

0548 0700 3F0286    IN11 BSTA,UN CHIN       INPUT CHARACTER.
0549 0703 C1             STRZ    R1         SAVE CHAR. IN R1.
0550 0704 3F02B4         BSTA,UN COUT       ECHO CHARACTER.
0551 0707 04F0           LODI,R0 H'F0'      LOAD NEGATIVE SIGN INDICATION
0552 0709 E52D           COMI,R1 A'-'       COMPARE WITH A'-' SIGN.
0553 070B 1801           BCTR,EQ IN12       BRANCH IF EQUAL.
0554 070D 20             EORZ    R0         LOAD POS. SIGN INDICATION.
0555 070E 17        IN12 RETC,UN
0556                *
0557 070F 0420      SPAC LODI,R0 SPCE       PRINT SPACE.
0558 0711 3F02B4         BSTA,UN COUT
0559 0714 17             RETC,UN
0560                *
0561 0715 CF078D    TST3 STRA,R3 OPRT       STORE INDEX IN OPERAND
0562 0718 3F02B4         BSTA,UN COUT       ECHO OPERATION CHAR.
0563 071B 3B72           BSTR,UN SPAC       PRINT SPACE.
0564                *
0565 071D 06FF           LODI,R2 -1         LOAD INDEX REGISTER.
0566 071F 3B5F           BSTR,UN IN11       INPUT SIGN EXPONENT.
0567 0721 CEA442         STRA,R0 *PTR2,R2,+ STORE IN OP1
0568 0724 3F0224         BSTA,UN BIN        INPUT EXPONENT.
0569 0727 01             LODZ    R1         SAVE IN R0.
0570 0728 CEA442         STRA,R0 *PTR2,R2,+ STORE IN OP1
0571 072B 3F0269         BSTA,UN BOUT       ECHO EXPONENT.
0572 072E 3B50           BSTR,UN IN11       INPUT SIGN MANTISSA.
0573 0730 CEA442         STRA,R0 *PTR2,R2,+ STORE IN OP1
0574 0733 3F0224    TST4 BSTA,UN BIN
0575 0736 01             LODZ    R1
0576 0737 CEA442         STRA,R0 *PTR2,R2,+ STORE IN OP1
0577 073A 3F0269         BSTA,UN BOUT       ECHO EXPONENT.
0578 073D E606           COMI,R2 LENG-1     END OF MANTISSA REACHED.
0579 073F 9872           BCFR,EQ TST4
0580                *
0581 0741 3B4C           BSTR,UN SPAC       PRINT SPACE.
0582 0743 043D           LODI,R0 A'='       PRINT = CHARACTER.
0583 0745 3F02B4         BSTA,UN COUT
0584 0748 3B45           BSTR,UN SPAC
0585 074A 7640           PPSU    F
0586 074C 0F078D         LODA,R3 OPRT       LOAD OPERATION CHAR ->R0
0587 074F BF0772         BSXA    OPRR,R3    CALL OPERATION INDEX.
```

```
LINE ADDR  OBJECT  E SOURCE

0588                *
0589 0752 7508           CPSL    WC
0590 0754 06FF           LODI,R2 -1         LOAD INDEX COUNTER.
0591 0756 3B26           BSTR,UN OT11       PRINT SIGN EXPONENT.
0592 0758 0EA446         LODA,R0 *PTRR,R2,+ LOAD RESULT BYTE IN R0.0
0593 075B C1             STRZ    R1
0594 075C 3F0269         BSTA,UN BOUT       ECHO CHARACTER.
0595 075F 3B1D           BSTR,UN OT11       PRINT SIGN MANTISSA
0596 0761 0EA446    TST5 LODA,R0 *PTRR,R2,+ LOAD RESULT BYTE
0597 0764 C1             STRZ    R1
0598 0765 3F0269         BSTA,UN BOUT       ECHO CHARACTER.
0599 0768 E606           COMI,R2 LENG-1     END OF REGISTER
0600 076A 9875           BCFR,EQ TST5       IF NOT READY BRANCH.
0601 076C 3F000A         BSTA,UN CRLF       OUTPUT CARRIAGE RETURN LF
0602 076F 1F06C4         BCTA,UN TEST       RETURN TO TEST.
0603                *
0604                *
0605 0772 1F05D8    OPRR BCTA,UN DDIV       PERFORM DIVISION.
0606 0775 1F05E1         BCTA,UN DMUL       PERFORM MULTIPLICATION.
0607 0778 1F0465         BCTA,UN DSUB       PERFORM SUBTRACTION.
0608 077B 1F046C         BCTA,UN DADD       PERFORM ADDITION.
0609                *
0610 077E 0EA446    OT11 LODA,R0 *PTRR,R2,+ LOAD RESULT BYTE IN R0
0611 0781 1804           BCTR,Z  OT12       TEST FOR ZERO  (POS).
0612 0783 042D           LODI,R0 A'-'       OUTPUT A '-' SIGN.
0613 0785 1B02           BCTR,UN OT13
0614 0787 042B      OT12 LODI,R0 A'+'       OUTPUT A '+' SIGN
0615 0789 3F02B4    OT13 BSTA,UN COUT       OUTPUT SIGN.
0616 078C 17             RETC,UN
0617                *
0618                * RAM STORAGE
0619                *
0620 078D           OPRT RES     1          OPERATION SYMBOL STORAGE LOCATION.
0621                *
0622                * TABLE OF OPERATION SYMBOLS
0623                *
0624 078E 2B2D2A3A  SRCH DATA    A'+-*:'
0625 06C4                END     TEST

 TOTAL ASSEMBLY ERRORS = 0000
```

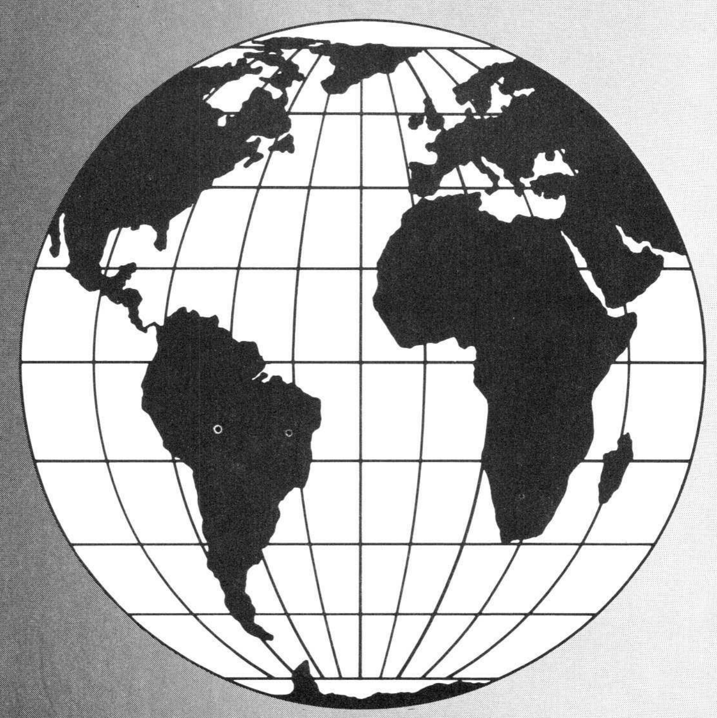
Related 2650 publications

no.	title	summary
TN 064	Digital cassette interface for a 2650 microprocessor system	Interface hardware and software for the Philips DCR digital cassette drive.
TN 069	2650 Microprocessor keyboard interfaces	Simple interfaces for low-cost keyboard systems.
TN 072	Introducing the Signetics 2651 PCI Terminology and operation modes	Description of the 2651 Programmable Communications Interface IC.
TN 083	Using the Signetics 2651 PCI with popular microprocessors	Simple hardware interfaces to use the 2651 Programmable Communications Interface with various microprocessors.
TN 084	Using seven-segment LED display with the 2650 microprocessor	Interfaces for single and multi-digit LED displays.
TN 085	Cyclic redundancy check by software	A short routine to encode and decode CRC check characters for the 2650.
TN 086	Introducing the Signetics 2655 PPI	Description of the 2655 Programmable Peripheral Interface.
TN 087	Audio cassette recorder interface for the 2650 microprocessor	Economical alternatives to the digital cassette recorder.
TN 089	CRT display using a standard TV monitor for 2650-based microcomputers	Economical solution for a visual display unit.
TN 092	2650 sorting routines	Sorting routines for single and multiple byte numbers.
TN 093	2650 binary floating point routines	Arithmetic routines for binary floating point numbers.

Electronic components and materials

for professional, industrial and consumer uses

from the world-wide Philips Group of Companies



Argentina: FAPESA I.y.C., Av. Crovara 2550, Tablada, Prov. de BUENOS AIRES, Tel. 652-7438/7478.
Australia: PHILIPS INDUSTRIES HOLDINGS LTD., Elcoma Division, 67 Mars Road, LANE COVE, 2066, N.S.W., Tel. 427 08 88.
Austria: ÖSTERREICHISCHE PHILIPS BAUELEMENTE Industrie G.m.b.H., Triester Str. 64, A-1101 WIEN, Tel. 62 91 11.
Belgium: M.B.L.E., 80, rue des Deux Gares, B-1070 BRUXELLES, Tel 523 00 00.
Brazil: IBRAPE, Caixa Postal 7383, Av. Paulista 2073-S/Loja, SAO PAULO, SP, Tel. 284-4511.
Canada: PHILIPS ELECTRONICS LTD., Electron Devices Div., 601 Milner Ave., SCARBOROUGH, Ontario, M1B 1M8, Tel. 292-5161.
Chile: PHILIPS CHILENA S.A., Av. Santa Maria 0760, SANTIAGO, Tel. 39-40 01.
Colombia: SADAPE S.A., P.O. Box 9805, Calle 13, No. 51 + 39, BOGOTA D.E. 1., Tel. 600 600.
Denmark: MINIWATT A/S, Emdrupvej 115A, DK-2400 KØBENHAVN NV., Tel. (01) 69 16 22.
Finland: OY PHILIPS AB, Elcoma Division, Kaivokatu 8, SF-00100 HELSINKI 10, Tel. 1 72 71.
France: R.T.C. LA RADIOTECHNIQUE-COMPELEC, 130 Avenue Ledru Rollin, F-75540 PARIS 11, Tel. 355-44-99.
Germany: VALVO, UB Bauelemente der Philips G.m.b.H., Valvo Haus, Burchardstrasse 19, D-2 HAMBURG 1, Tel. (040) 3296-1.
Greece: PHILIPS S.A. HELLENIQUE, Elcoma Division, 52, Av. Syngrou, ATHENS, Tel. 915 311.
Hong Kong: PHILIPS HONG KONG LTD., Comp. Dept., Philips Ind. Bldg., Kung Yip St., K.C.T.L. 289, KWAI CHUNG, N.T. Tel. 12-24 51 21.
India: PHILIPS INDIA LTD., Elcoma Div., Band Box House, 254-D, Dr. Annie Besant Rd., Prabhadevi, BOMBAY-25-DD, Tel. 457 311-5.
Indonesia: P.T. PHILIPS-RALIN ELECTRONICS, Elcoma Division, 'Timah' Building, Jl. Jen. Gatot Subroto, JAKARTA, Tel. 44 163.
Ireland: PHILIPS ELECTRICAL (IRELAND) LTD., Newstead, Clonskeagh, DUBLIN 14, Tel. 69 33 55.
Italy: PHILIPS S.p.A., Sezione Elcoma, Piazza IV Novembre 3, I-20124 MILANO, Tel. 2-6994.
Japan: NIHON PHILIPS CORP., Shuwa Shinagawa Bldg., 26-33 Takanawa 3-chome, Minato-ku, TOKYO (108), Tel. 448-5611.
(IC Products) SIGNETICS JAPAN, LTD., TOKYO, Tel. (03) 230-1521.
Korea: PHILIPS ELECTRONICS (KOREA) LTD., Elcoma Division, Philips House, 260-199 Itaewon-dong, Yongsan-ku, SEOUL, Tel. 794-4202.
Mexico: ELECTRONICA S.A. de C.V., Varsovia No. 36, MEXICO 6, D.F., Tel. 533-11-80.
Netherlands: PHILIPS NEDERLAND B.V., Afd. Elonco, Boschdijk 525, NL 5600 PD EINDHOVEN, Tel. (040) 79 33 33.
New Zealand: PHILIPS Electrical Ind. Ltd., Elcoma Division, 2 Wagener Place, St. Lukes, AUCKLAND, Tel. 867 119.
Norway: NORSK A/S PHILIPS, Electronica Dept., Vitaminveien 11, Grefsen, OSLO 4, Tel. (02) 15 05 90.
Peru: CADESA, Rocca de Vergallo 247, LIMA 17, Tel. 62 85 99.
Philippines: ELDAC, Philips Industrial Dev. Inc., 2246 Pasong Tamo, MAKATI-RIZAL, Tel. 86-89-51 to 59.
Portugal PHILIPS PORTUGESA S.A.R.L., Av. Eng. Duharte Pacheco 6, LISBOA 1, Tel. 68 31 21.
Singapore: PHILIPS SINGAPORE PTE LTD., Elcoma Div., P.O.B. 340, Toa Payoh CPO, Lorong 1, Toa Payoh, SINGAPORE 12, Tel. 53 88 11.
South Africa: EDAC (Pty.) Ltd., South Park Lane, New Doornfontein, JOHANNESBURG 2001, Tel. 24/6701.
Spain: COPRESA S.A., Balmes 22, BARCELONA 7, Tel. 301 63 12.
Sweden: A.B. ELCOMA, Lidingövägen 50, S-10 250 STOCKHOLM 27, Tel. 08/67 97 80.
Switzerland: PHILIPS A.G., Elcoma Dept., Edenstrasse 20, CH-8027 ZÜRICH, Tel. 01/44 22 11.
Taiwan: PHILIPS TAIWAN LTD., 3rd Fl., San Min Building, 57-1, Chung Shan N. Rd, Section 2, P.O. Box 22978, TAIPEI, Tel. 5513101-5.
Turkey: TÜRK PHILIPS TICARET A.S., EMET Department, Inonu Cad. No. 78-80, ISTANBUL, Tel. 43 59 10.
United Kingdom: MULLARD LTD., Mullard House, Torrington Place, LONDON WC1E 7HD, Tel. 01-580 6633.
United States: (Active devices & Materials) AMPEREX SALES CORP., Providence Pike, SLATERSVILLE, R.I. 02876, Tel. (401) 762-9000.
(Passive devices) MEPCO/ELECTRA INC., Columbia Rd., MORRISTOWN, N.J. 07960, Tel. (201) 539-2000.
(IC Products) SIGNETICS CORPORATION, 811 East Arques Avenue, SUNNYVALE, California 94086, Tel. (408) 739-7700.
Uruguay: LUZILECTRON S.A., Rondeau 1567, piso 5, MONTEVIDEO, Tel. 9 43 21.
Venezuela: IND. VENEZOLANAS PHILIPS S.A., Elcoma Dept., A. Ppal de los Ruices, Edif. Centro Colgate, CARACAS, Tel. 36 05 11.

This information is furnished for guidance, and with no guarantees as to its accuracy or completeness; its publication conveys no licence under any patent or other right, nor does the publisher assume liability for any consequence of its use; specifications and availability of goods mentioned in it are subject to change without notice; it is not to be reproduced in any way, in whole or in part, without the written consent of the publisher.

Printed in The Netherlands   Date of release: 31 August 1978   T.D.   9398 009 40011