To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

H8/300L Super Low Power Series

Addition of Single-Precision Floating-Point Numbers (FADD)

Introduction

The software FADD adds single-precision floating-point numbers placed in general-purpose registers and places the result of addition in general-purpose registers.

Target Device

H8/38024

Contents

1. Arguments

Description		Memory area	Data length (bytes)
Input	Augend	R0, R1	4
	Addend	R2, R3	4
Output	Result of addition	R0, R1	4

2. Changes to Internal Registers and Flags

R0	R1	R2	R3	R4	R5	R6	R7
○	○	×	×	×	×	×	—

I	U	H	U	N	Z	V	C
—	—	×	—	×	×	×	

Legend
—: No change
×: Undefined
○: Result

3. Specifications

Program memory (bytes)
280
Data memory (bytes)
0
Stack (bytes)
0
Clock cycle count
268
Reentrant
Possible
Relocation
Possible
Interrupt
Possible

4. Notes

The clock cycle count (268) in the specifications is for the example shown in figure 1.

For the format of floating-point numbers, see "About Single-precision floating-point Numbers <Reference>."

5. Description

5.1 Details of functions

1. The following arguments are used with the software FADD:
   a. Input arguments:
      R0: Sets the upper 2 bytes of a single-precision floating-point as augend.
      R1: Sets the lower 2 bytes of a single-precision floating-point as augend.
      R2: Sets the upper 2 bytes of a single-precision floating-point as addend.
      R3: Sets the lower 2 bytes of a single-precision floating-point as addend.
   b. Output arguments:
      R0: The upper 2 bytes of a single-precision floating-point are placed here as the result of addition.
      R1: The lower 2 bytes of a single-precision floating-point are placed here as the result of addition.
2. The following figure illustrates the execution of the software FADD. When the input arguments are set as shown in (1), the result of addition is placed in R0 and R1 as shown in (2).

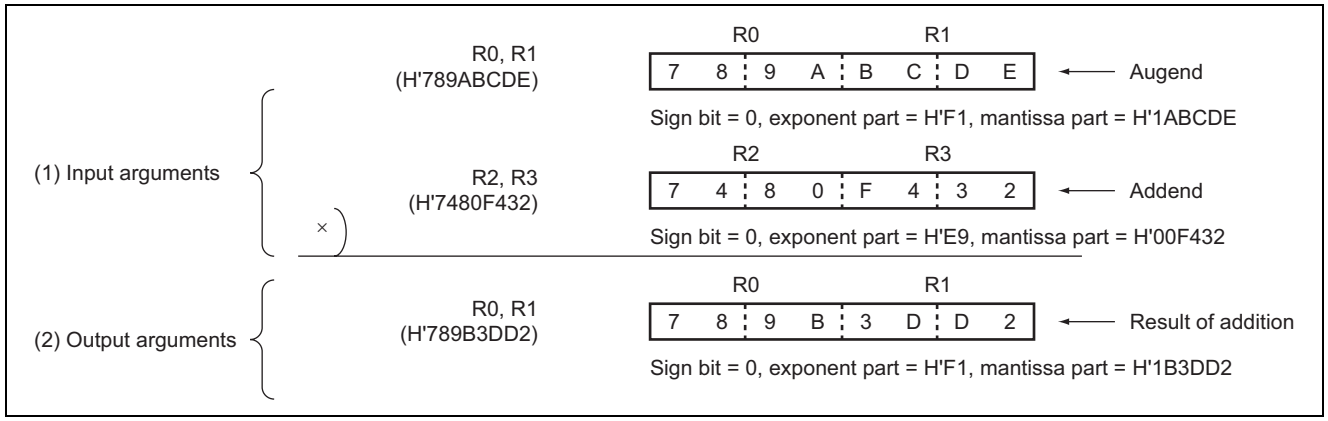


Figure 1 Example of Software FADD Execution

5.2 Notes on usage

1. The maximum and minimum values that can be handled by the software FADD are as follows:

   Positive maximum H'7F800000
   Positive minimum H'00000001

   Negative maximum H'80000001
   Negative minimum H'FF800000

2. All positive single-precision floating-point numbers H'7F800001 to H'7FFFFFFF are treated as a maximum value (H'7F800000). All negative single-precision floating-point numbers H'FF800000 to H'FFFFFFFF are treated as a minimum value (H'FF800000).
3. As a maximum value is treated as infinity (∞), the result of ∞ + 100 or ∞ - 100 becomes infinite. (See table 1.)

Table 1 Examples of Operation with Maximum Values Used as Arguments

Augend	Addend	Result
H'7F800000 to H”7FFFFFFF	********	H'7F800000
Not H'7F800000 to H’FFFFFFFF	H'7F800000 to H'7FFFFFFF	H'7F800000
H'FF800000 to H’FFFFFFFF	********	H'FF800000
Not H'7F800000 to H”7FFFFFFF	H'FF800000 to H'FFFFFFFF	H'FF800000

Note: * represents a hexadecimal number.

4. H'80000000 is treated as H'00000000 (zero).
5. After execution of the software FADD, the augend and addend data will be lost. When the input arguments are still needed after software FADD execution, save them in memory.

5.3 Description of data memory

The software FADD uses no data memory.

5.4 Example of usage

Set an augend and an addend in the general-purpose registers and call the software FADD as a subroutine.

```
WORK1    . RES. W    2
WORK2    . RES. W    2
WORK3    . RES. W    2

         MOV. W      @WORK1, R0
         MOV. W      @WORK1+2, R1

         MOV. W      @WORK2, R2
         MOV. W      @WORK2+2, R3

         JSR         @FADD

         MOV. W      R0, @WORK3
         MOV. W      R1  @WORK3+2
             .
             .
             .
```

Reserve a data memory area in which the user program places an augend, an addend, the result of addition.

Place the augend set by the user program in R0 and R1.

Place the addend set by the user program in R2 and R3.

Call the software FADD as a subroutine.

Save the result of addition set in the output argument in R0 and R1 .

5.5 Operation

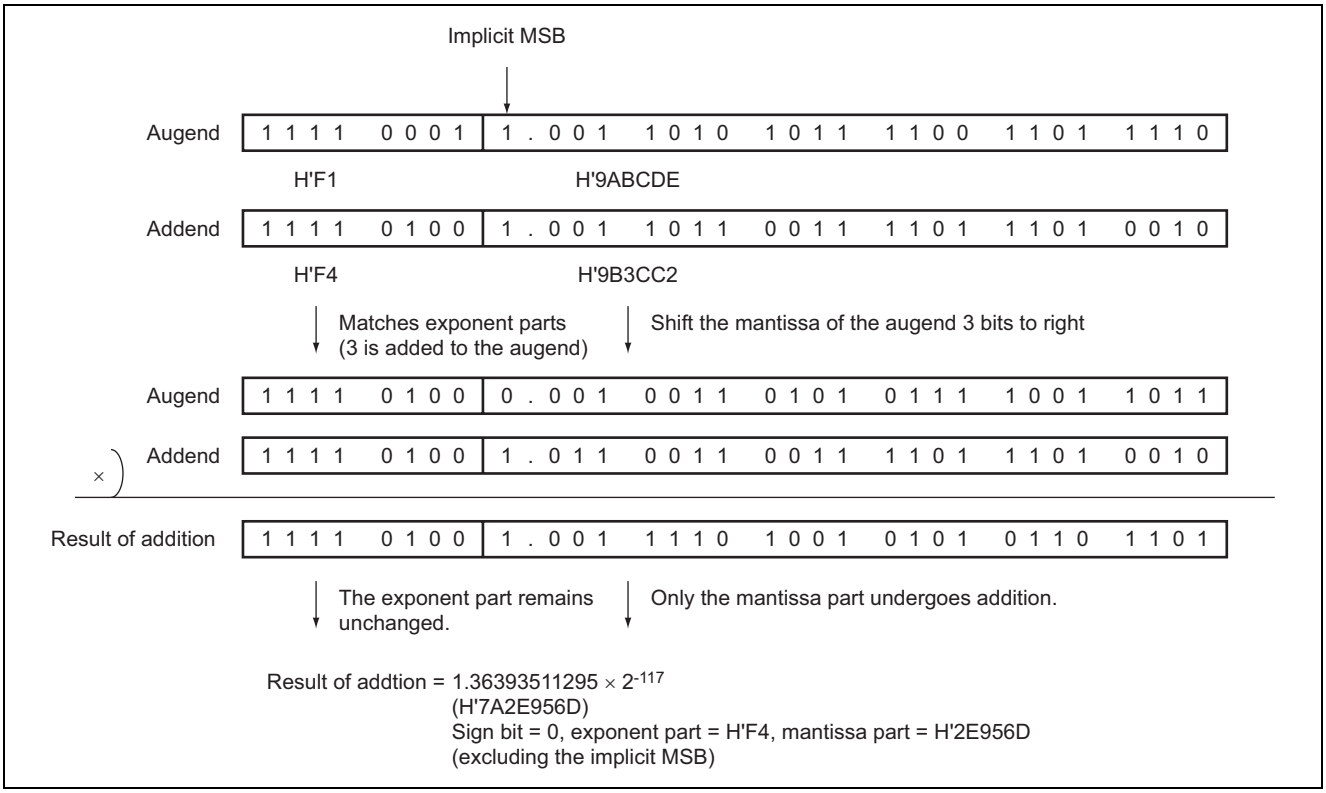
Addition of single-precision floating-point numbers is done in the following steps:

1. The software checks whether the augend and addend are +∞ or -∞ .
   a. When the exponent of the augend is H'FF, either of the following values is output depending on the state of the sign bit:

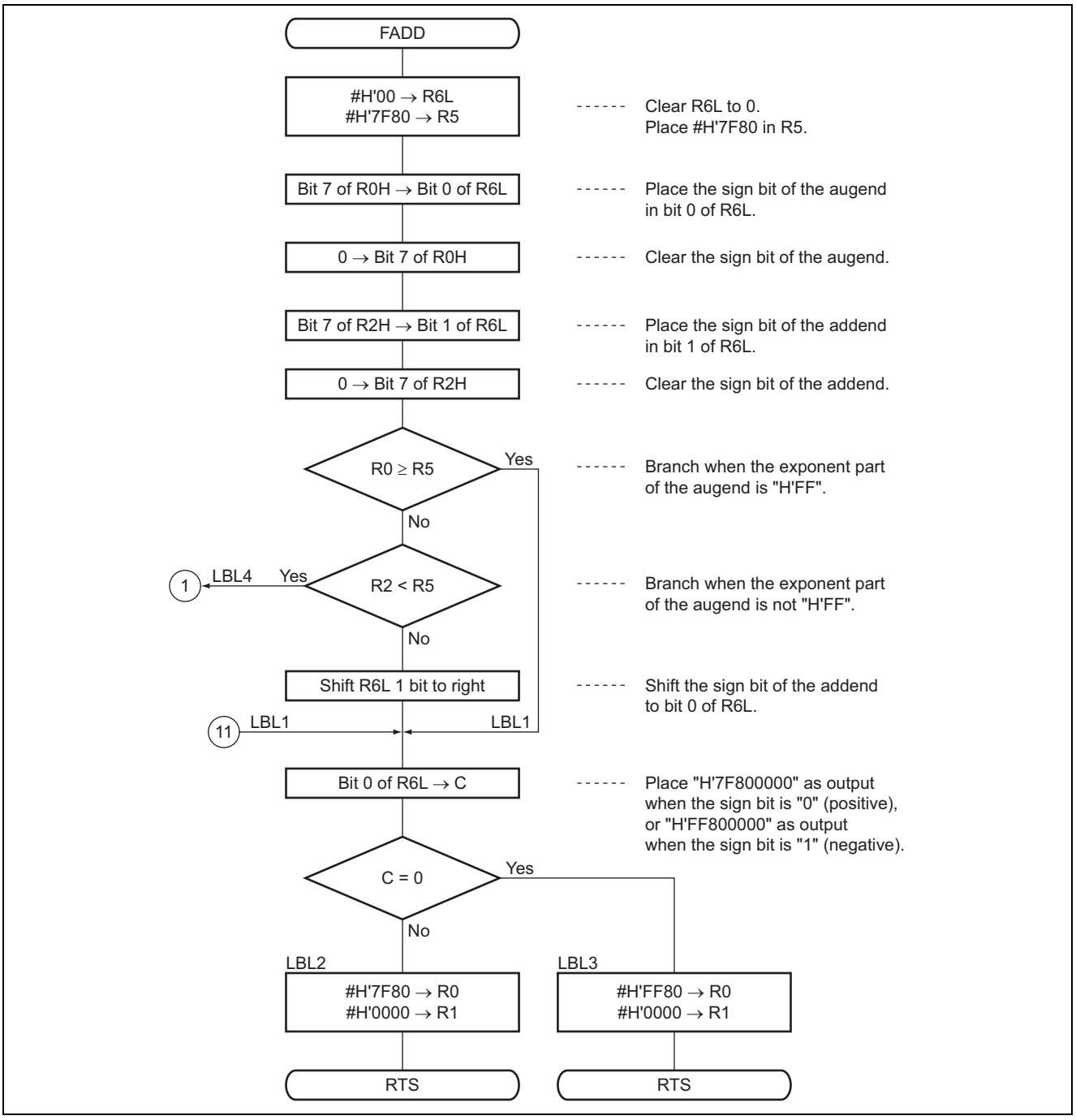
Sign bit	Output value
0 (positive)	H'7F800000 (+∞)
1 (negative)	H'FF800000 (–∞)

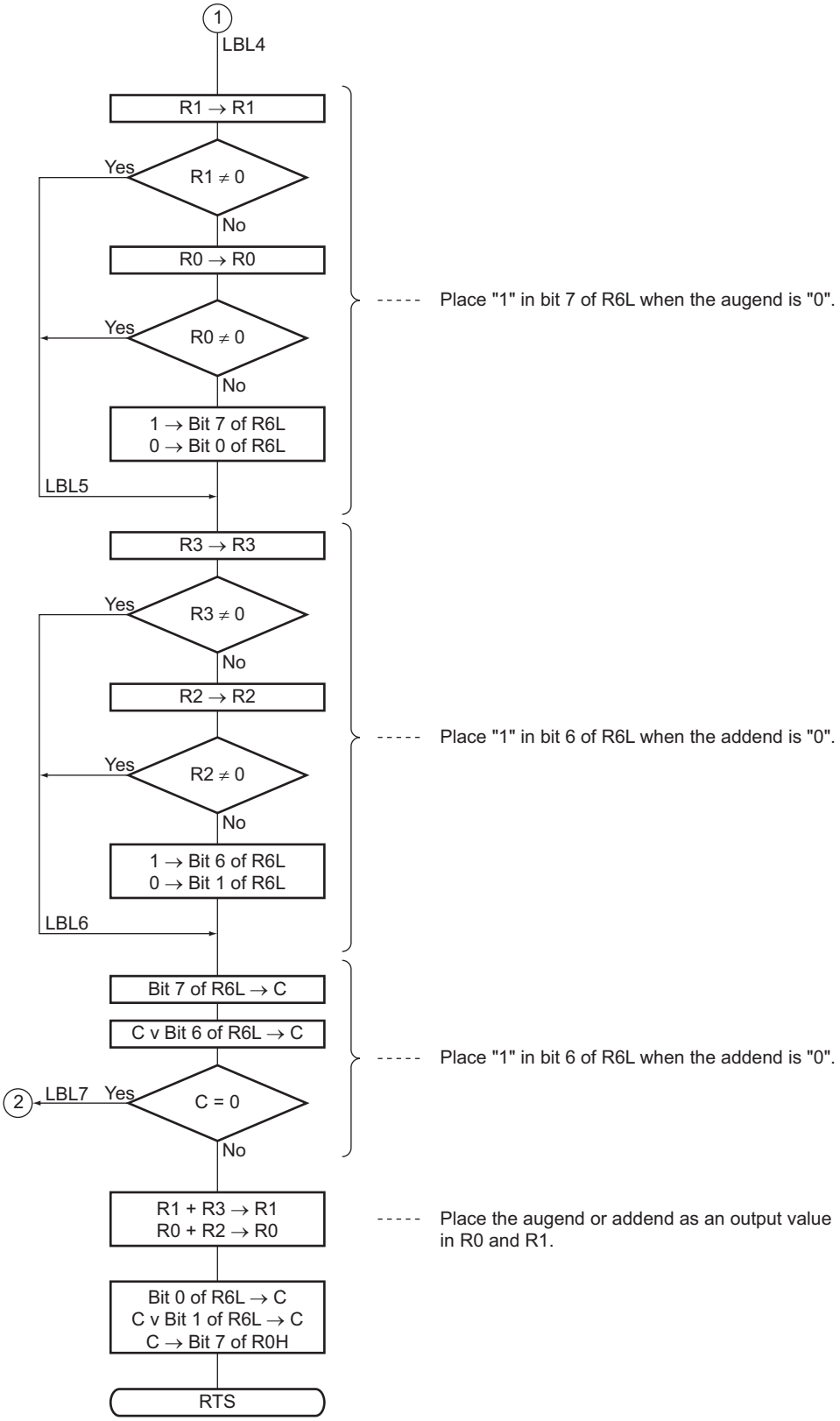
   b. The table above also applies when the augend is neither + ∞ nor – ∞ and the exponent of the addend is H'FF.
2. The software checks whether the augend and addend are "0".
   a. If either the augend or addend is "0", the other number is output (if both are "0", "H'00000000" is output).
3. The software attempts to match the exponent of the augend with that of the addend.
   a. The smaller number of the exponent is incremented and, at the same time, the mantissa (including the implicit MSB) is shifted digit by digit to right until the exponent of the augend matches that of the addend. (In the case of the denormalized format, 1 is added to the exponent and the MSB of the mantissa is taken as implicitly being zero.
4. The mantissas are added.

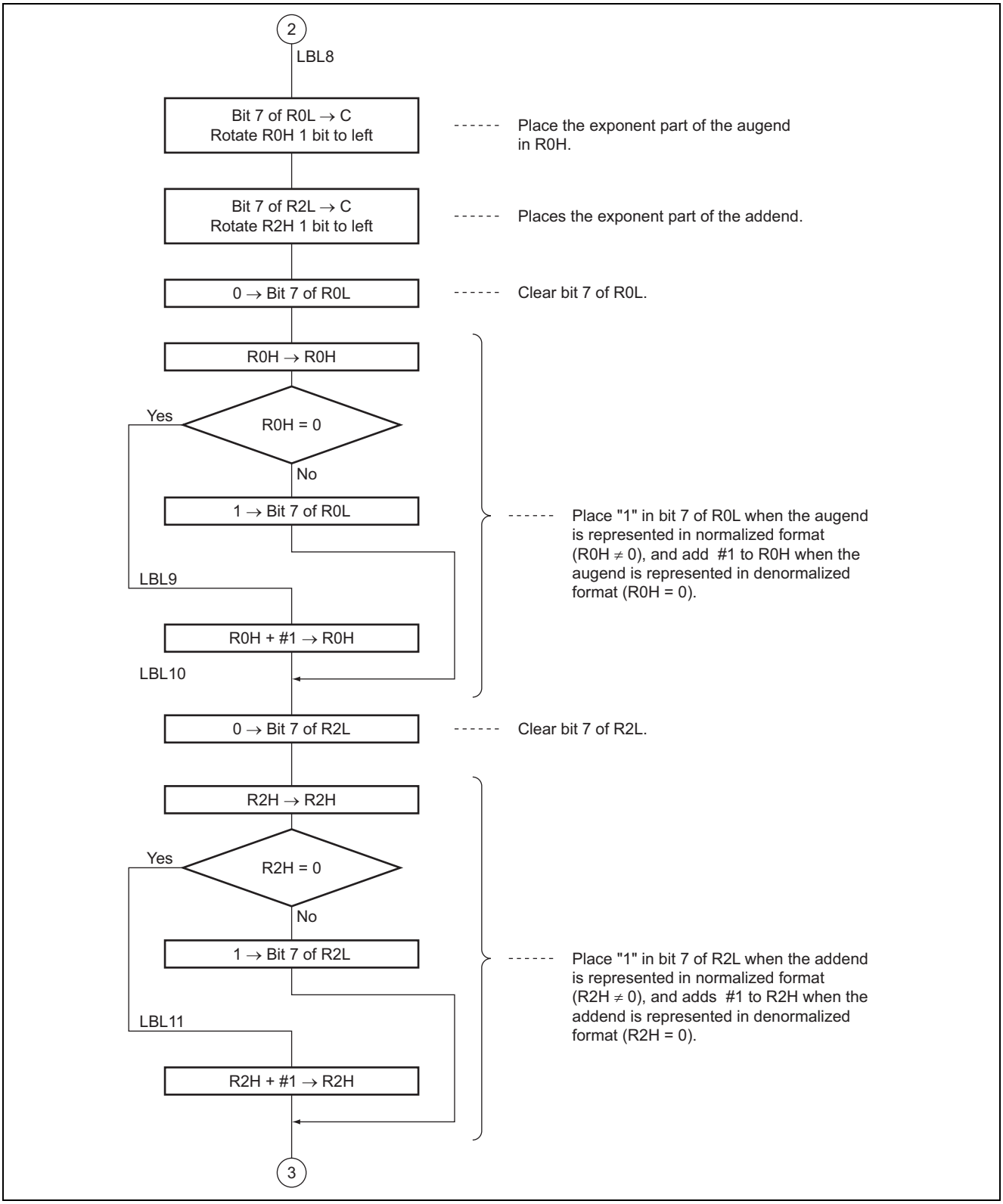
5. The result of addition is corrected to produce a number in the floating-point data format.
   (Example)
   Augend = $1.20888876915 \times 2^{114}$
   (H'789ABCDE)
   Sign bit = 0, exponent = H'F1, mantissa = H'1ABCDE
   (implicit MSB is not included)
   Addend = $1.21282410622 \times 2^{-117}$
   (H'7A1B3DD2)
   Sign bit = 0, exponent = H'F4, mantissa = H'1B3DD2
   (implicit MSB is not included)

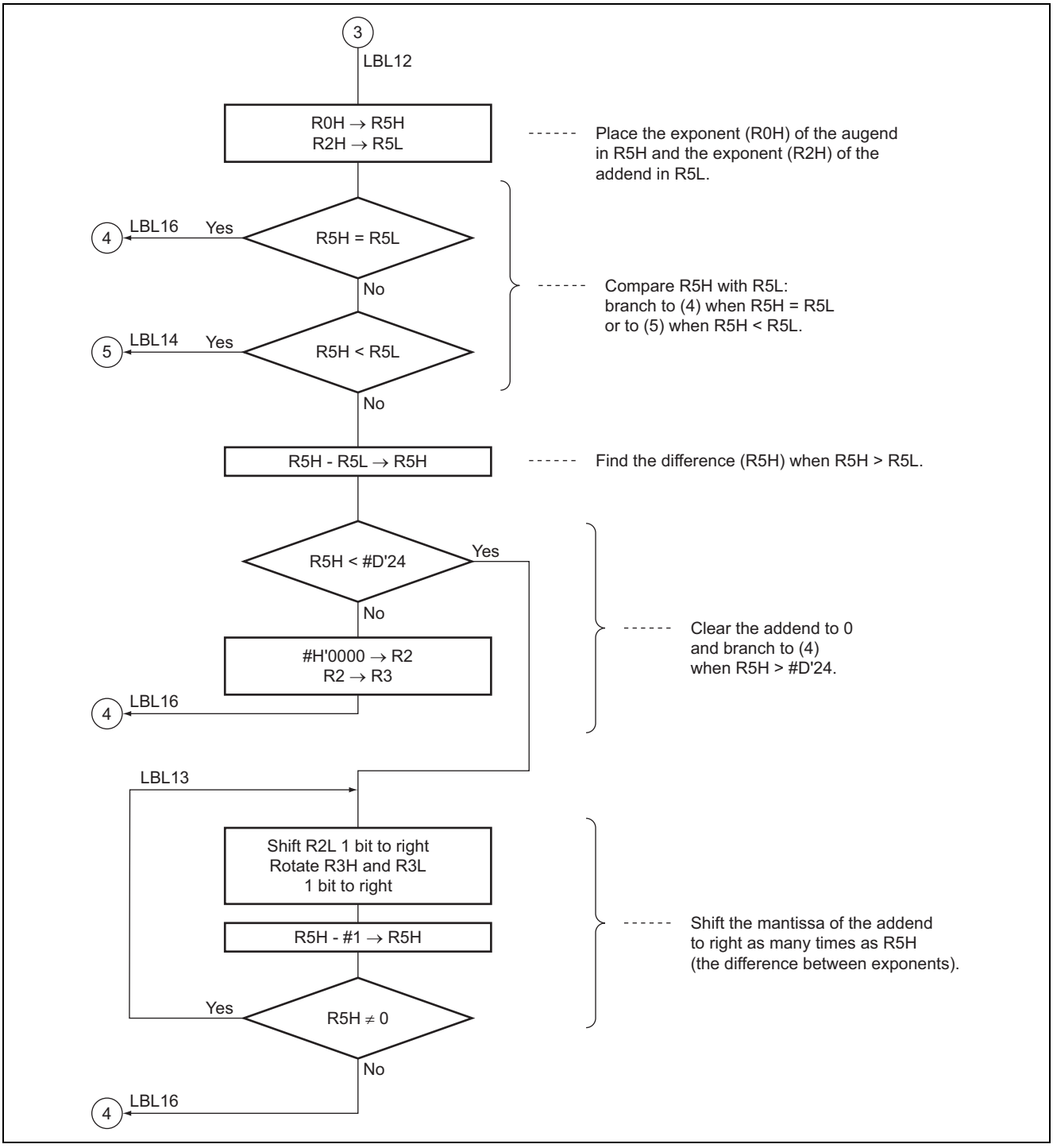


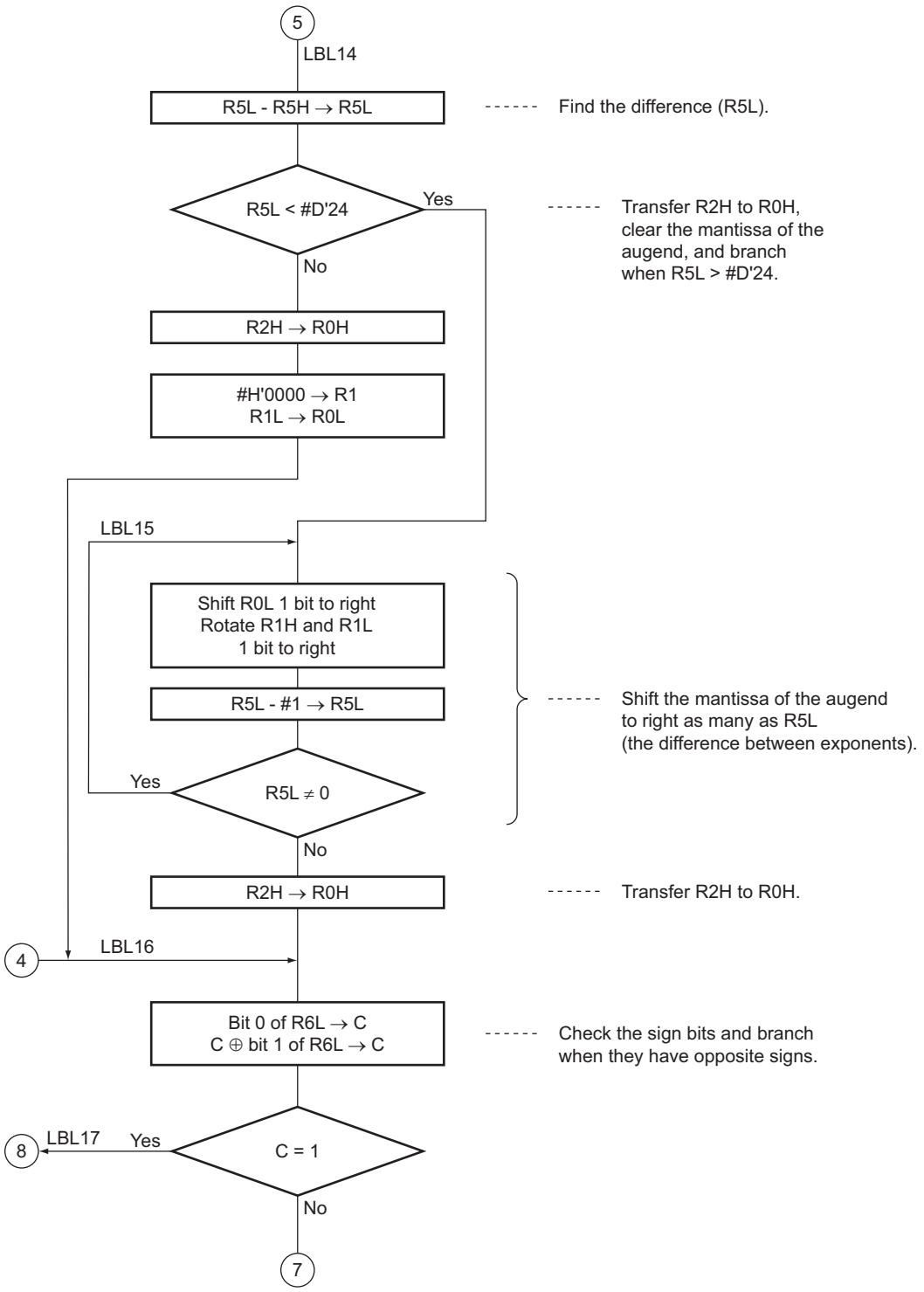
6. Flowchart

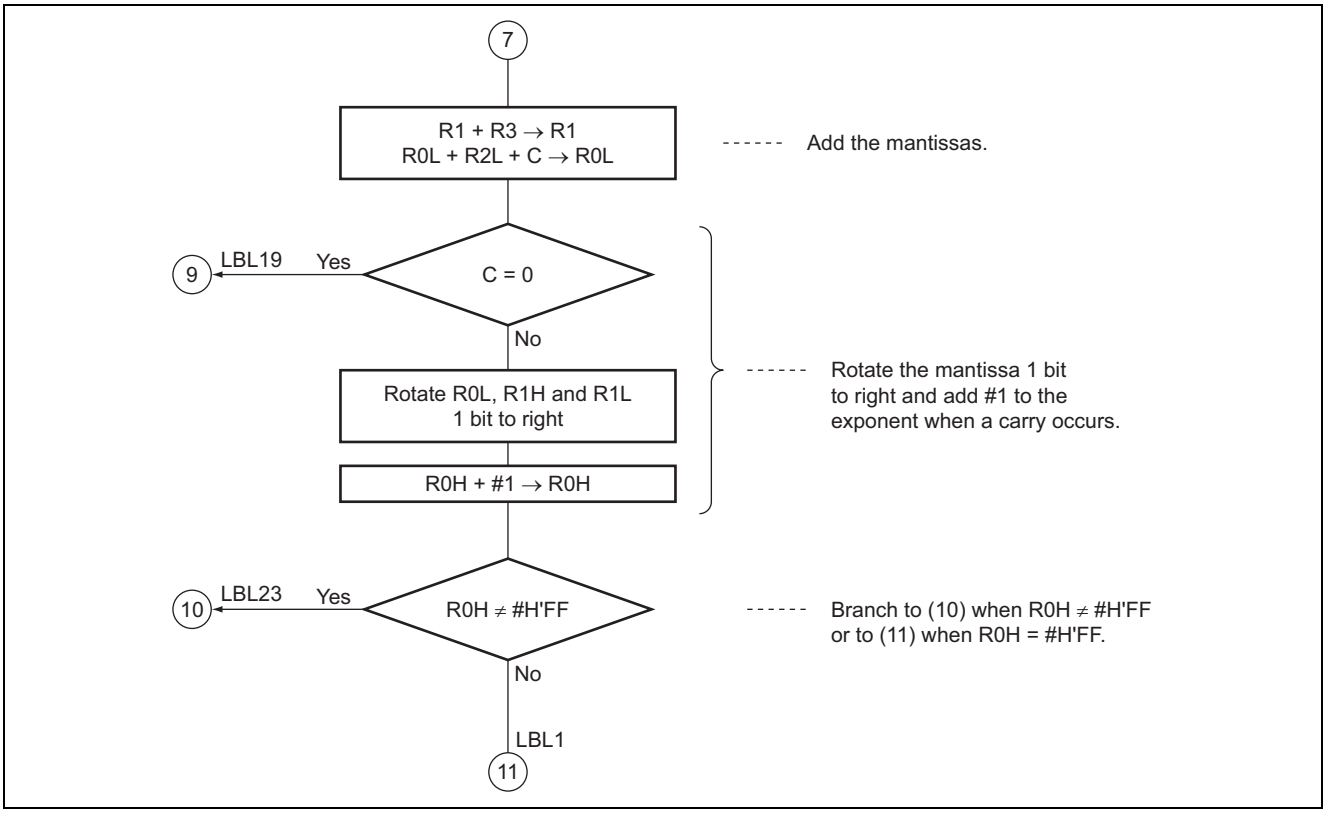

FADD
#H'00 → R6L
#H'7F80 → R5
Clear R6L to 0.
Place #H'7F80 in R5.
Bit 7 of R0H → Bit 0 of R6L
Place the sign bit of the augend in bit 0 of R6L.
0 → Bit 7 of R0H
Clear the sign bit of the augend.
Bit 7 of R2H → Bit 1 of R6L
Place the sign bit of the addend in bit 1 of R6L.
0 → Bit 7 of R2H
Clear the sign bit of the addend.
R0 ≥ R5
Yes
No
Branch when the exponent part of the augend is "H'FF".
1
LBL4
Yes
R2 < R5
No
Branch when the exponent part of the augend is not "H'FF".
Shift R6L 1 bit to right
Shift the sign bit of the addend to bit 0 of R6L.
11
LBL1
LBL1
Bit 0 of R6L → C
Place "H'7F800000" as output when the sign bit is "0" (positive), or "H'FF800000" as output when the sign bit is "1" (negative).
C = 0
Yes
No
LBL2
#H'7F80 → R0
#H'0000 → R1
LBL3
#H'FF80 → R0
#H'0000 → R1
RTS
RTS

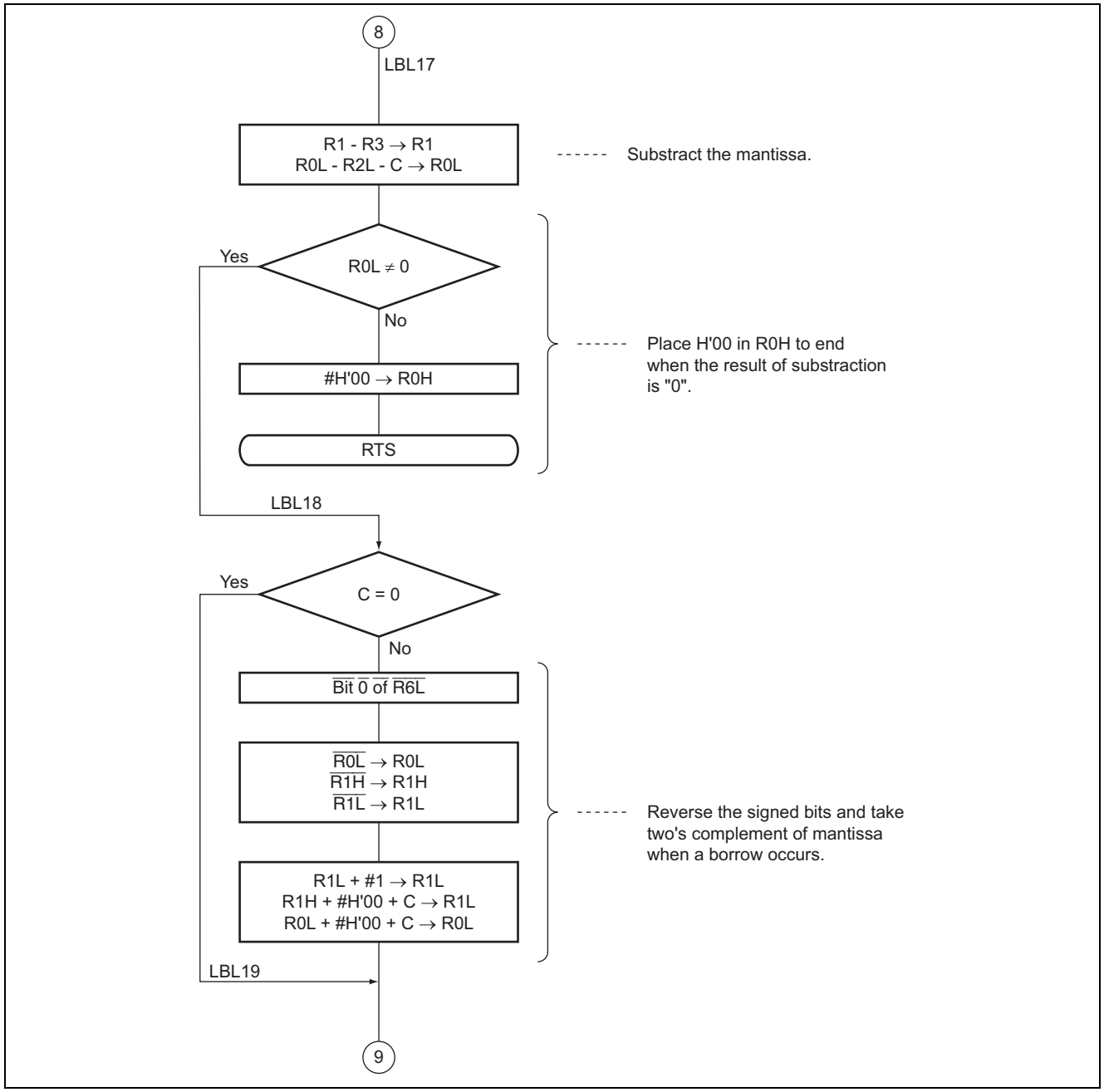

1
LBL4
R1 → R1
Yes
R1 ≠ 0
No
R0 → R0
Yes
R0 ≠ 0
No
1 → Bit 7 of R6L
0 → Bit 0 of R6L
LBL5
Place "1" in bit 7 of R6L when the augend is "0".
R3 → R3
Yes
R3 ≠ 0
No
R2 → R2
Yes
R2 ≠ 0
No
1 → Bit 6 of R6L
0 → Bit 1 of R6L
LBL6
Place "1" in bit 6 of R6L when the addend is "0".
Bit 7 of R6L → C
C v Bit 6 of R6L → C
2
LBL7
Yes
C = 0
No
Place "1" in bit 6 of R6L when the addend is "0".
R1 + R3 → R1
R0 + R2 → R0
Place the augend or addend as an output value in R0 and R1.
Bit 0 of R6L → C
C v Bit 1 of R6L → C
C → Bit 7 of R0H
RTS

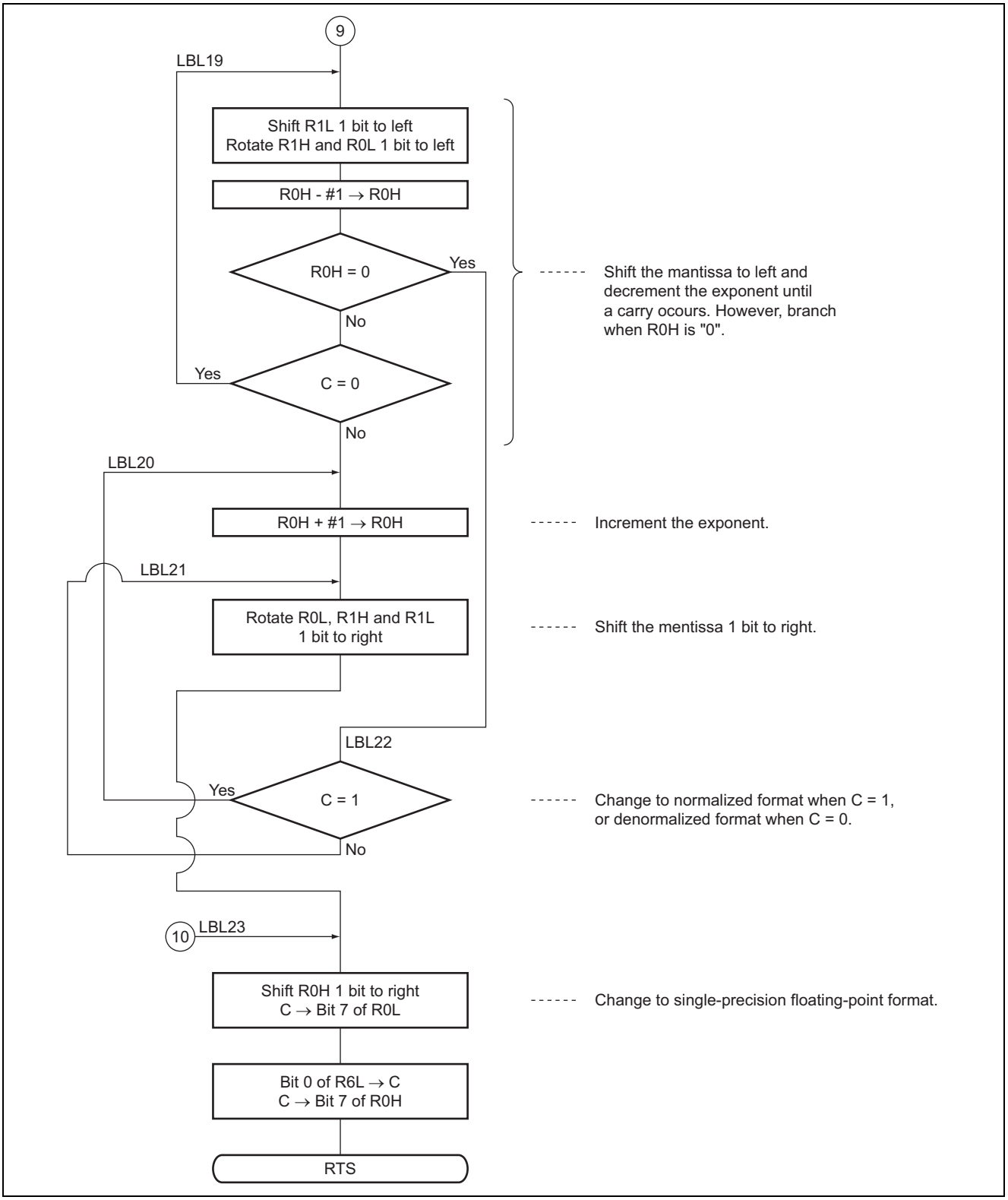

2
LBL8
Bit 7 of R0L → C
Rotate R0H 1 bit to left
Place the exponent part of the augend in R0H.
Bit 7 of R2L → C
Rotate R2H 1 bit to left
Places the exponent part of the addend.
0 → Bit 7 of R0L
Clear bit 7 of R0L.
R0H → R0H
Yes
R0H = 0
No
1 → Bit 7 of R0L
Place "1" in bit 7 of R0L when the augend is represented in normalized format (R0H ≠ 0), and add #1 to R0H when the augend is represented in denormalized format (R0H = 0).
LBL9
R0H + #1 → R0H
LBL10
0 → Bit 7 of R2L
Clear bit 7 of R2L.
R2H → R2H
Yes
R2H = 0
No
1 → Bit 7 of R2L
Place "1" in bit 7 of R2L when the addend is represented in normalized format (R2H ≠ 0), and adds #1 to R2H when the addend is represented in denormalized format (R2H = 0).
LBL11
R2H + #1 → R2H
3


3
LBL12
R0H → R5H
R2H → R5L
Place the exponent (R0H) of the augend in R5H and the exponent (R2H) of the addend in R5L.
4
LBL16
Yes
R5H = R5L
No
Compare R5H with R5L:
branch to (4) when R5H = R5L
or to (5) when R5H < R5L.
5
LBL14
Yes
R5H < R5L
No
R5H - R5L → R5H
Find the difference (R5H) when R5H > R5L.
R5H < #D'24
Yes
No
#H'0000 → R2
R2 → R3
4
LBL16
Clear the addend to 0
and branch to (4)
when R5H > #D'24.
LBL13
Shift R2L 1 bit to right
Rotate R3H and R3L
1 bit to right
R5H - #1 → R5H
Yes
R5H ≠ 0
No
4
LBL16
Shift the mantissa of the addend
to right as many times as R5H
(the difference between exponents).


5
LBL14
R5L - R5H → R5L
Find the difference (R5L).
R5L < #D'24
Yes
No
Transfer R2H to R0H, clear the mantissa of the augend, and branch when R5L > #D'24.
R2H → R0H
#H'0000 → R1
R1L → R0L
LBL15
Shift R0L 1 bit to right
Rotate R1H and R1L 1 bit to right
R5L - #1 → R5L
R5L ≠ 0
Yes
No
Shift the mantissa of the augend to right as many as R5L (the difference between exponents).
R2H → R0H
Transfer R2H to R0H.
4
LBL16
Bit 0 of R6L → C
C ⊕ bit 1 of R6L → C
Check the sign bits and branch when they have opposite signs.
8
LBL17
Yes
C = 1
No
7


7
R1 + R3 → R1
R0L + R2L + C → R0L
Add the mantissas.
9
LBL19
Yes
C = 0
No
Rotate R0L, R1H and R1L
1 bit to right
R0H + #1 → R0H
Rotate the mantissa 1 bit
to right and add #1 to the
exponent when a carry occurs.
10
LBL23
Yes
R0H ≠ #H'FF
No
Branch to (10) when R0H ≠ #H'FF
or to (11) when R0H = #H'FF.
LBL1
11


8
LBL17
R1 - R3 → R1
R0L - R2L - C → R0L
Substract the mantissa.
Yes
R0L ≠ 0
No
#H'00 → R0H
RTS
Place H'00 in R0H to end when the result of substraction is "0".
LBL18
Yes
C = 0
No
Bit 0 of R6L
R0L → R0L
R1H → R1H
R1L → R1L
R1L + #1 → R1L
R1H + #H'00 + C → R1L
R0L + #H'00 + C → R0L
Reverse the signed bits and take two's complement of mantissa when a borrow occurs.
LBL19
9


9
LBL19
Shift R1L 1 bit to left
Rotate R1H and R0L 1 bit to left
R0H - #1 → R0H
R0H = 0
Yes
No
Yes
C = 0
No
Shift the mantissa to left and decrement the exponent until a carry ocours. However, branch when R0H is "0".
LBL20
R0H + #1 → R0H
Increment the exponent.
LBL21
Rotate R0L, R1H and R1L 1 bit to right
Shift the mentissa 1 bit to right.
LBL22
Yes
C = 1
No
Change to normalized format when C = 1, or denormalized format when C = 0.
10
LBL23
Shift R0H 1 bit to right
C → Bit 7 of R0L
Change to single-precision floating-point format.
Bit 0 of R6L → C
C → Bit 7 of R0H
RTS

7. Program List

```
*** H8/300 ASSEMBLER VER 1.0B ** 08/18/92 10:20:43
PROGRAM NAME =
  1                                ;*********************************************************
  2                                ;*
  3                                ;*       00 - NAME   :FLOATING POINT ADDITION (FADD)
  4                                ;*
  5                                ;*********************************************************
  6                                ;*
  7                                ;*       ENTRY       :R0 (UPPER WORD OF SUMMAND)
  8                                ;*                    R1 (LOWER WORD OF SUMMAND)
  9                                ;*                    R2 (UPPER WORD OF ADDEND)
 10                                ;*                    R3 (LOWER WORD OF ADDEND)
 11                                ;*
 12                                ;*       RETURNS     :R0 (UPPER WORD OF RESULT)
 13                                ;*                    R1 (LOWER WORD OF RESULT)
 14                                ;*
 15                                ;*********************************************************
 16                                ;
 17 FADD_cod C  0000                        .SECTION    FADD_code,CODE,ALIGN=2
 18                                         .EXPORT     FADD
 19                                ;
 20 FADD_cod C         00000000  FADD  .EQU $              ;Entry point
 21 FADD_cod C  0000   FE00            MOV.B    #H'00,R6L   ;Clear R6L
 22 FADD_cod C  0002   79057F80        MOV.W    #H'7F80,R5  ;Set "H'7F80"
 23                                ;
 24 FADD_cod C  0006   7770            BLD      #7,R0H
 25 FADD_cod C  0008   670E            BST      #0,R6L      ;Set sign bit to bit 0 of R6L
 26 FADD_cod C  000A   7270            BCLR     #7,R0H      ;Bit clear bit 7 of R0H
 27                                ;
 28 FADD_cod C  000C   7772            BLD      #7,R2H
 29 FADD_cod C  000E   671E            BST      #1,R6L      ;Set sign bit to bit 1 of R6L
 30 FADD_cod C  0010   7272            BCLR     #7,R2H      ;Bit clear bit 7 of R2H
 31                                ;
 32 FADD_cod C  0012   1D05            CMP.W    R0,R5
 33 FADD_cod C  0014   4306            BLS      LBL1        ;Branch if "exponent of
                                                             summand"="H'FF"
 34 FADD_cod C  0016   1D25            CMP.W    R2,R5
 35 FADD_cod C  0018   421A            BHI      LBL4        ;Branch if not "exponent of
                                                             summand"="H'FF"
 36 FADD_cod C  001A   110E            SHLR     R6L         ;Shift R6L 1 bit right
 37 FADD_cod C  001C           LBL1
 38 FADD_cod C  001C   770E            BLD      #0,R6L      ;Bit load sign bit
 39 FADD_cod C  001E   450A            BCS      LBL3        ;Branch if sign bit=1
 40 FADD_cod C  0020           LBL2
 41 FADD_cod C  0020   79007F80        MOV.W    #H'7F80,R0  ;Set plus maximum number
 42 FADD_cod C  0024   79010000        MOV.W    #H'0000,R1
 43 FADD_cod C  0028   5470            RTS
 44 FADD_cod C  002A           LBL3
 45 FADD_cod C  002A   7900FF80        MOV.W    #H'FF80,R0  ;Set minus minimum number
 46 FADD_cod C  002E   79010000        MOV.W    #H'0000,R1
 47 FADD_cod C  0032   5470            RTS
 48                                ;
```

```
 49 FADD_cod C  0034           LBL4
 50 FADD_cod C  0034   0D11           MOV.W    R1,R1       ;
 51 FADD_cod C  0036   4608           BNE      LBL5        ;Branch if Z=0
 52 FADD_cod C  0038   0D00           MOV.W    R0,R0
 53 FADD_cod C  003A   4604           BNE      LBL5        ;Branch if Z=0
 54 FADD_cod C  003C   707E           BSET     #7,R6L      ;Bit set bit 7 of R6L
 55 FADD_cod C  003E   720E           BCLR #0,R6L          ;Bit clear bit 0 of R6L
 56 FADD_cod C  0040           LBL5
 57 FADD_cod C  0040   0D33           MOV.W    R3,R3
 58 FADD_cod C  0042   4608           BNE      LBL6        ;Branch if Z=0
 59 FADD_cod C  0044   0D22           MOV.W    R2,R2
 60 FADD_cod C  0046   4604           BNE      LBL6        ;Branch if Z=0
 61 FADD_cod C  0048   706E           BSET     #6,R6L      ;Bit set bit 6 of R6L
 62 FADD_cod C  004A   721E           BCLR     #1,R6L      ;Bit clear bit 1 of R6L
 63 FADD_cod C  004C           LBL6
 64 FADD_cod C  004C   777E           BLD      #7,R6L
 65 FADD_cod C  004E   746E           BOR      #6,R6L
 66 FADD_cod C  0050   440C           BCC      LBL8        ;Branch if not summand=addend=0
 67 FADD_cod C  0052   0931           ADD.W    R3,R1       ;Set summand and addend to result
 68 FADD_cod C  0054   0920           ADD.W    R2,R0
 69 FADD_cod C  0056   770E           BLD      #0,R6L
 70 FADD_cod C  0058   741E           BOR      #1,R6L
 71 FADD_cod C  005A   6770           BST      #7,R0H      ;Set sign bit
 72 FADD_cod C  005C   5470           RTS
 73                            ;
 74 FADD_cod C  005E           LBL8
 75 FADD_cod C  005E   7778           BLD      #7,R0L
 76 FADD_cod C  0060   1200           ROTXL    R0H         ;Set exponent of summand to R0H
 77                            ;
 78 FADD_cod C  0062   777A           BLD      #7,R2L
 79 FADD_cod C  0064   1202           ROTXL    R2H         ;Set exponent of addend to R0L
 80                            ;
 81 FADD_cod C  0066   7278           BCLR     #7,R0L
 82 FADD_cod C  0068   0C00           MOV.B    R0H,R0H
 83 FADD_cod C  006A   4704           BEQ      LBL9        ;Branch if summand is normalized
 84 FADD_cod C  006C   7078           BSET     #7,R0L      ;Set implicit MSB to summand
 85 FADD_cod C  006E   4002           BRA      LBL10       ;Branch always
 86 FADD_cod C  0070           LBL9
 87 FADD_cod C  0070   8001           ADD.B    #H'01,R0H
 88 FADD_cod C  0072           LBL10
 89 FADD_cod C  0072   727A           BCLR     #7,R2L
 90 FADD_cod C  0074   0C22           MOV.B    R2H,R2H
 91 FADD_cod C  0076   4704           BEQ      LBL11       ;Branch if addend is normalized
 92 FADD_cod C  0078   707A           BSET     #7,R2L      ;Set implicit MSB to addend
 93 FADD_cod C  007A   4002           BRA      LBL12       ;Branch always
 94 FADD_cod C  007C           LBL11
 95  FADD_cod C        007C 8201               ADD.B       #H'01,R2H
 96                            ;
 97 FADD_cod C  007E           LBL12
 98 FADD_cod C  007E   0C05           MOV.B    R0H,R5H
 99 FADD_cod C  0080   0C2D           MOV.B    R2H,R5L
100 FADD_cod C  0082   1CD5           CMP.B    R5L,R5H
101 FADD_cod C  0084   4738           BEQ      LBL16       ;Branch if R5H=R5L
102 FADD_cod C  0086   451A           BCS      LBL14       ;Branch if R5H<R5L
```

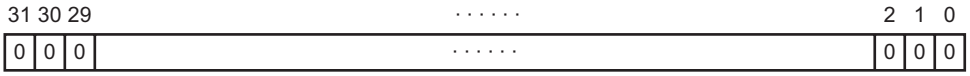
```
103                             ;
104 FADD_cod C  0088   18D5             SUB.B    R5L,R5H
105 FADD_cod C  008A   A518             CMP.B    #D'24,R5H   ;Set bit counter
106 FADD_cod C  008C   4508             BCS      LBL13       ;Branch if R5H<D'24
107 FADD_cod C  008E   79020000         MOV.W    #H'0000,R2  ;Clear addend
108 FADD_cod C  0092   0D23             MOV.W    R2,R3
109 FADD_cod C  0094   4028             BRA      LBL16       ;Branch always
110 FADD_cod C  0096             LBL13
111 FADD_cod C  0096   110A             SHLR     R2L         ;Shift mantissa of addend 1 bit left
112 FADD_cod C  0098   1303             ROTXR    R3H
113 FADD_cod C  009A   130B             ROTXR    R3L
114 FADD_cod C  009C   1A05             DEC.B    R5H         ;Decrement bit counter
115 FADD_cod C  009E   46F6             BNE      LBL13       ;Branch Z=0
116 FADD_cod C  00A0   401C             BRA      LBL16       ;Branch always
117                             ;
118 FADD_cod C  00A2             LBL14
119 FADD_cod C  00A2   185D             SUB.B    R5H,R5L
120 FADD_cod C  00A4   AD18             CMP.B    #D'24,R5L
121 FADD_cod C  00A6   450A             BCS      LBL15       ;Branch if R5L<D'24
122 FADD_cod C  00A8   0C20             MOV.B    R2H,R0H
123 FADD_cod C  00AA   79010000         MOV.W    #H'0000,R1  ;Clear summand
124 FADD_cod C  00AE   0C98             MOV.B    R1L,R0L
125 FADD_cod C  00B0   400C             BRA      LBL16       ;Branch always
126 FADD_cod C  00B2             LBL15
127 FADD_cod C  00B2   1108             SHLR     R0L         ;Shift mantissa of summand 1 bit right
128 FADD_cod C  00B4   1301             ROTXR    R1H
129 FADD_cod C  00B6   1309             ROTXR    R1L
130 FADD_cod C  00B8   1A0D             DEC.B    R5L         ;Decrement bit counter
131 FADD_cod C  00BA   46F6             BNE      LBL15       ;Branch if Z=0
132 FADD_cod C  00BC   0C20             MOV.B    R2H,R0H
133                             ;
134 FADD_cod C  00BE             LBL16
135 FADD_cod C  00BE   770E             BLD      #0,R6L
136 FADD_cod C  00C0   751E             BXOR     #1,R6L
137 FADD_cod C  00C2   4516             BCS      LBL17       ;Branch if different sign bit
138                             ;
139 FADD_cod C  00C4   0931             ADD.W    R3,R1       ;Addition mantissa
140 FADD_cod C  00C6   0EA8             ADDX.B   R2L,R0L
141 FADD_cod C  00C8   442A             BCC      LBL19       ;Branch if C = 0
142 FADD_cod C  00CA   1308             ROTXR    R0L         ;Rotate mantissa 1 bit right
143 FADD_cod C  00CC   1301             ROTXR    R1H
144 FADD_cod C  00CE   1309             ROTXR    R1L
145 FADD_cod C  00D0   8001             ADD.B    #H'01,R0H   ;Increment exponent
146 FADD_cod C  00D2   A0FF             CMP.B    #H'FF,R0H
147 FADD_cod C  00D4   4638             BNE      LBL23       ;Branch if not exponent=H'FF
148 FADD_cod C  00D6   5A000000         JMP      @LBL1       ;Jump
149                             ;
150 FADD_cod C  00DA             LBL17
151 FADD_cod C  00DA   1931             SUB.W    R3,R1       ;Substruct mantissa
152 FADD_cod C  00DC   1EA8             SUBX.B   R2L,R0L
153 FADD_cod C  00DE   4604             BNE      LBL18       ;Branch if Z=0
154 FADD_cod C  00E0   F000             MOV.B    #H'00,R0H   ;Clear R0H
155 FADD_cod C  00E2   5470             RTS
```

```
156 FADD_cod C  00E4           LBL18
157 FADD_cod C  00E4   440E           BCC      LBL19       ;Branch if C = 0
158 FADD_cod C  00E6   710E           BNOT     #0,R6L      ;Bit not sign bit
159 FADD_cod C  00E8   1708           NOT       R0L        ;2's complement mantissa
160 FADD_cod C  00EA   1701           NOT      R1H
161 FADD_cod C  00EC   1709           NOT      R1L
162 FADD_cod C  00EE   8901           ADD.B    #H'01,R1L
163 FADD_cod C  00F0   9100           ADDX.B   #H'00,R1H
164 FADD_cod C  00F2   9800           ADDX.B   #H'00,R0L
165                            ;
166 FADD_cod C  00F4           LBL19
167 FADD_cod C  00F4   1009           SHLL     R1L         ;Shift mantissa 1 bit left
168 FADD_cod C  00F6   1201           ROTXL    R1H
169 FADD_cod C  00F8   1208           ROTXL    R0L
170 FADD_cod C  00FA   1A00           DEC.B    R0H         ;Decrement exponemt
171 FADD_cod C  00FC   470C           BEQ      LBL22       ;Branch if exponent=0
172 FADD_cod C  00FE   44F4           BCC      LBL19       ;Branch if exponent>0
173 FADD_cod C  0100           LBL20
174 FADD_cod C  0100   0A00           INC.B    R0H         ;Increment exponent
175 FADD_cod C  0102           LBL21
176 FADD_cod C  0102   1308           ROTXR    R0L         ;Rotate mantissa 1 bit right
177 FADD_cod C  0104   1301           ROTXR    R1H
178 FADD_cod C  0106   1309           ROTXR    R1L
179 FADD_cod C  0108   4004           BRA      LBL23       ;Branch always
180 FADD_cod C  010A           LBL22
181 FADD_cod C  010A   45F4           BCS      LBL20       ;Branch if C = 1
182 FADD_cod C  010C   40F4           BRA      LBL21       ;Branch always
183                            ;
184 FADD_cod C  010E           LBL23                        ;Chage floating point format
185 FADD_cod C  010E   1100           SHLR     R0H
186 FADD_cod C  0110   6778           BST      #7,R0L
187 FADD_cod C  0112   770E           BLD      #0,R6L
188 FADD_cod C  0114   6770           BST      #7,R0H
189 FADD_cod C  0116   5470           RTS
190                            ;
191                                   .END
*****TOTAL ERRORS 0
*****TOTAL WARNINGS 0
```

About Single-Precision Floating-Point Numbers <Reference>

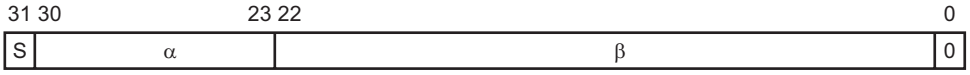
Single-Precision Floating-Point Formats:

1. Internal representation of single-precision floating-point numbers
   In this Application Note, the following formats are applied to single-precision floating-point numbers depending on their values (R = real number):
   A. Internal representation for R = 0



   All of the 32 bits are 0's.

   B. Normalized format

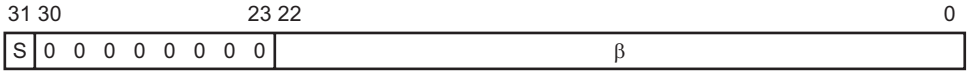


   α is an exponent whose field is 8 bits long. β is a mantissa whose field is 23 bits long. The value of R can be represented by the following equation (on conditions that $1 \leq \alpha \leq 254$) :

   $R = 2^S \times 2^{\alpha-127} \times (1 + 2^{-1} \times \beta_{22} + 2^{-2} \times \beta_{21} + \ldots\ldots + 2^{-23} \times \beta_0)$

   where βi is the value of the i-th bit ($0 \leq i \leq 22$) and S is the sign bit.

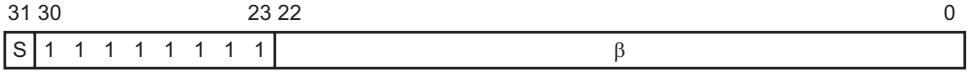
   C. Denormalized format



   where β is a mantissa whose field is 23 bits long. This format is used to represent a real number too small to be represented in the normal format. In this format, R can be represented by the following equation:

   $R = 2^S \times 2^{-126} \times (2^{-1} \times \beta_{22} + 2^{-2} \times \beta_{21} + \ldots\ldots + 2^{-23} \times \beta_0)$

   D. Infinity



   where β is a mantissa whose field is 23 bits long. In this Application Note, however, the following rules apply if all exponents are 1's;

   Positive infinity when S = 0

   $R = +\infty$

   Negative infinity when S = 1

   $R = -\infty$

2. Example of internal representation

If S = B'0 (binary)

α = B'10000011 (binary)

β = B'1011100……0 (binary)

Then the corresponding real number is as follows:

$R = 2^{0} \times 2^{131-127} \times (1 + 2^{-1} + 2^{-3} + 2^{-4} + 2^{-5})$

$= 16 + 8 + 2 + 1 + 0.5 = 27.5$

A. Maximum and minimum values

The maximum value (R_{MAX}) and minimum value (R_{MIN}), in terms of the absolute value, are as follows:

$RMAX = 2^{254\ -\ 127} \times (1 + 2^{-1} + 2^{-2} + 2^{-3}$ ……$+ 2^{-23})$

$= 3.37 \times 10^{38}$

$RMIN = 2^{-126} \times 2^{-23} = 2^{-140} = 1.40 \times 10^{-45}$

The absolute values within the above range can be represented.

Website and Support

Renesas Technology Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry
csc@renesas.com

Revision Record

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Sep.18.03	—	First edition issued
2.00	Nov.30.06	All pages	Content correction

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.