

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.

H8/300H Tiny Series

Multiplication of Single-Precision Floating-Point Numbers (FMUL)

Introduction

Multiplies single-precision floating-point numbers set in general registers and stores the result in general registers.

Target Device

H8/300H Tiny Series

Contents

1. Function	2
2. Arguments.....	2
3. Changes to Internal Registers and Flags	2
4. Programming Specifications	3
5. Notes.....	3
6. Descriptions	4
7. Flowchart.....	7
8. Program Listing.....	15
<Reference> Description of Single-Precision Floating-Point Formats.....	20

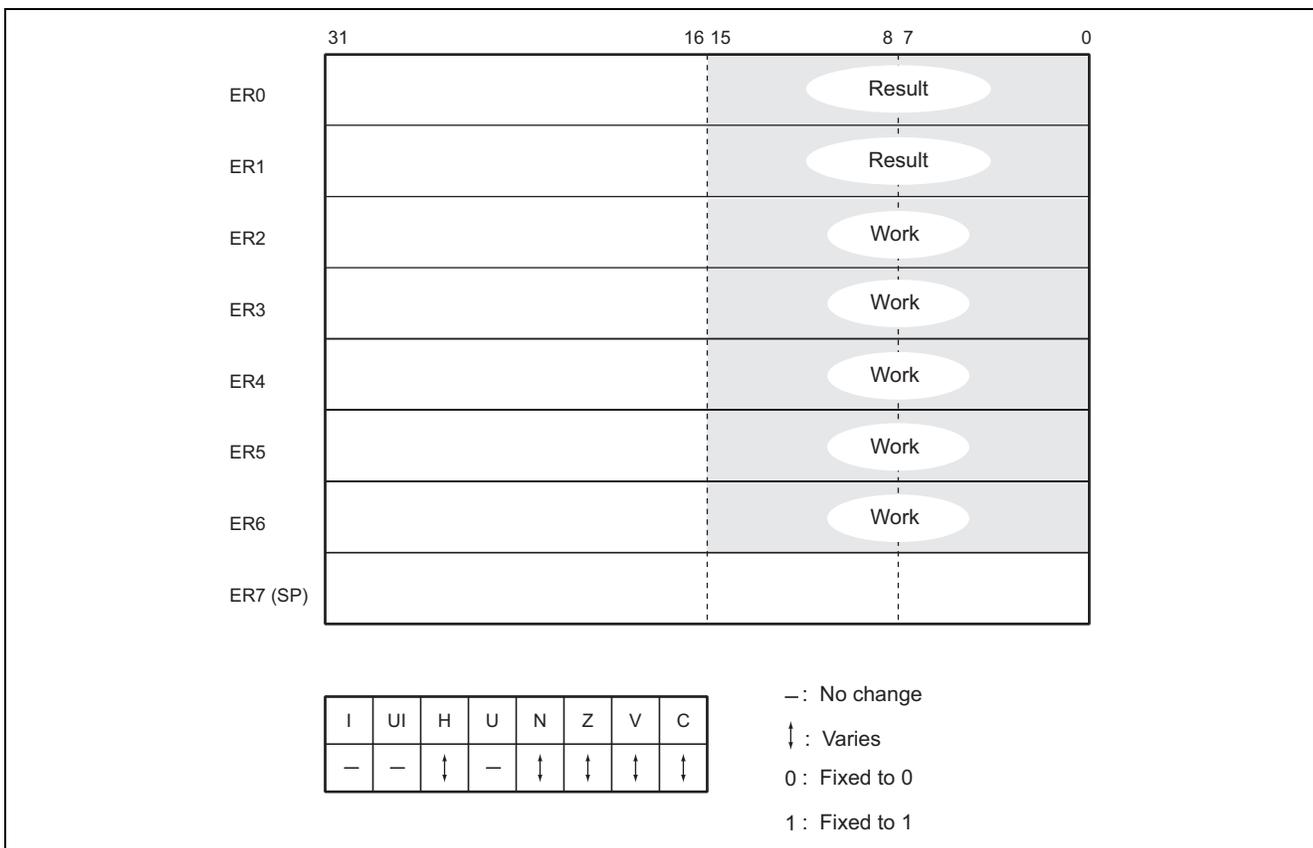
1. Function

1. Multiplies single-precision floating-point numbers set in general registers and stores the result in general registers.
2. The arguments are all in the single-precision floating-point data format.

2. Arguments

Description		Storage Location	Data Length (Bytes)
Input	Multiplicand	R0, R1	4
	Multiplier	R2, R3	4
Output	Result	R0, R1	4

3. Changes to Internal Registers and Flags



4. Programming Specifications

	Program memory (bytes)	
	348	
	Data memory (bytes)	
	0	
	Stack (bytes)	
	16	
	Number of cycles	
	1078	
	Re-entrant	
	Yes	
	Relocatable	
	Yes	
	Interrupts during execution	
	Yes	

5. Notes

The number of cycles in the programming specifications is the value for execution of the example in figure 1. For details on the floating-point data format, refer to Reference: Description of Single-Precision Floating-Point Formats.

6. Descriptions

6.1 Descriptions of Functions

1. The arguments are as follows.
 - 1) Set the input arguments as follows.
 - R0: higher-order two bytes of the multiplicand
 - R1: lower-order two bytes of the multiplicand
 - R2: higher-order two bytes of the multiplier
 - R3: lower-order two bytes of the multiplier
 - 2) The FMUL subroutine sets the following output arguments.
 - R0: higher-order two bytes of the multiplication result
 - R1: lower-order two bytes of the multiplication result

2. The following figure illustrates the execution of the FMUL subroutine. When the input arguments are set as shown below, the subroutine places the result of multiplication in R0 and R1.

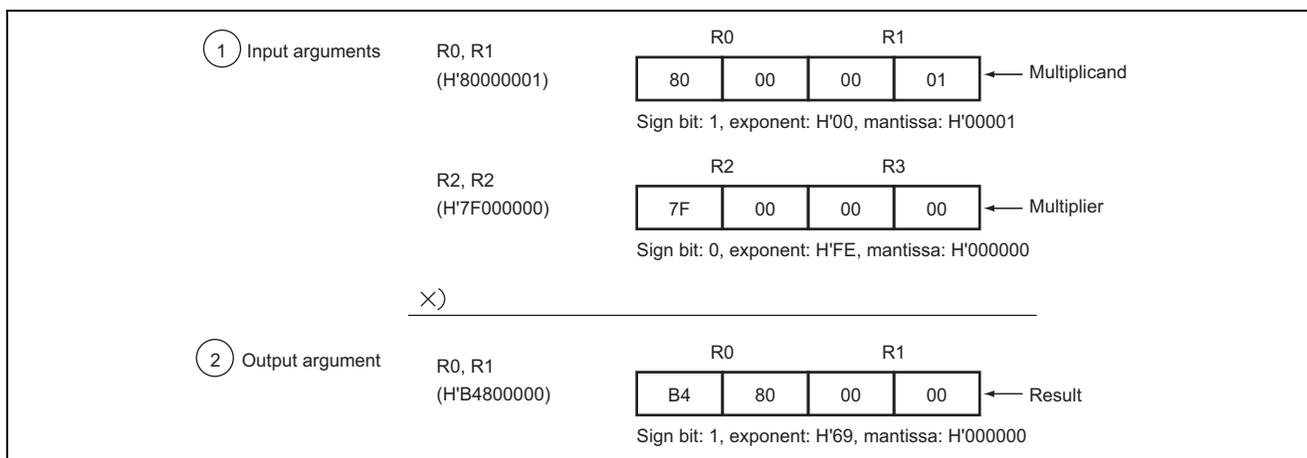


Figure 1 Example of FMUL Execution

6.2 Usage Notes

1. The maximum and minimum values handled by the FMUL subroutine are given below.
 Maximum positive value: H'7F80000
 Minimum positive value: H'00000001
 Maximum negative value: H'80000001
 Minimum negative value: H'FF800000
2. Positive single-precision floating-point numbers from H'7F800001 to H'7FFFFFFF are regarded as having the maximum value, H'7F800000. Negative single-precision floating-point numbers from H'FF800000 to H'FFFFFFF are regarded as having the minimum value, H'FF800000.
3. The maximum value is handled as infinity (∞), that is, the result of multiplying other numbers by the maximum value is the maximum value. For example, in multiplication by 100, $\infty \times 100 = \infty$ and $\infty \times (-100) = -\infty$ (see table 1).

Table 1 Examples of Results when Maximum Values are Specified as Arguments

Multiplicand	Multiplier	Result
> H'7F800000 ($+\infty$)	Positive value	H'7F800000 ($+\infty$)
	Negative value	H'FF800000 ($-\infty$)
< H'FF800000 ($-\infty$)	Positive value	H'FF800000 ($-\infty$)
	Negative value	H'7F800000 ($+\infty$)
Positive value	> H'7F800000 ($+\infty$)	H'7F800000 ($+\infty$)
	< H'FF800000 ($-\infty$)	H'FF800000 ($-\infty$)
Negative value	> H'7F800000 ($+\infty$)	H'FF800000 ($-\infty$)
	< H'FF800000 ($-\infty$)	H'7F800000 ($+\infty$)

4. H'80000000 is handled as H'00000000 (zero).
5. The multiplicand and multiplier stored in the general registers are lost through execution of FMUL. When you will still require the input arguments, save them elsewhere in memory beforehand.

6.3 Description of Data Memory

No data memory is used by the FMUL subroutine.

6.4 Example of Usage

After setting the multiplicand and multiplier in the general registers, call the FADD subroutine.

```

WORK1 . RES. W 2      ..... Reservation of the data memory area for setting of the multiplicand by the user program.
WORK2 . RES. W 2      ..... Reservation of the data memory area for setting of the multiplier by the user program.
WORK3 . RES. W 2      ..... Reservation of the data memory area where the product of multiplication will be stored by the user
      .
      .
      .
MOV. W @WORK1, R0     ..... Sets the multiplicand specified by the user program as an input argument.
MOV. W @WORK1+2, R1
MOV. W @WORK2, R2     ..... Sets the multiplier specified by the user program as an input argument.
MOV. W @WORK2+2, R3
JSR   @FMUL           ..... Subroutine call of FMUL
MOV. W R0, @WORK3     ..... Transfers the product set as the output argument to the data memory area of the user program.
MOV. W R0, @WORK3+2
    
```

6.5 Principles of Operation

Multiplication of the single-precision floating-point numbers is according to the following sequence.

1. The multiplicand and multiplier are checked for zero values.
If one or both holds a zero, H'00000000 is output.
2. The multiplicand and multiplier are checked for infinite (+∞ or -∞) values.
If one or both of them is infinite (+∞ or -∞), the result is as given in table 6.1.
3. The exponents of the multiplicand and multiplier are matched.
Let R1 be the multiplicand (sign bit = S1, exponent = α1, mantissa = β1) and R2 the multiplier (sign bit = S2, exponent = α2, mantissa = β2); R1 and R2 are then expressed as follows.

$$R1 = (-1)^{S1} \times 2^{\alpha1-127} \times \beta1$$

$$R2 = (-1)^{S2} \times 2^{\alpha2-127} \times \beta2$$

Multiplication is as follows.

$$R1 \times R2 = (-1)^{S1+S2} \times 2^{\alpha1+\alpha2-127-127} \times \beta1 \times \beta2$$

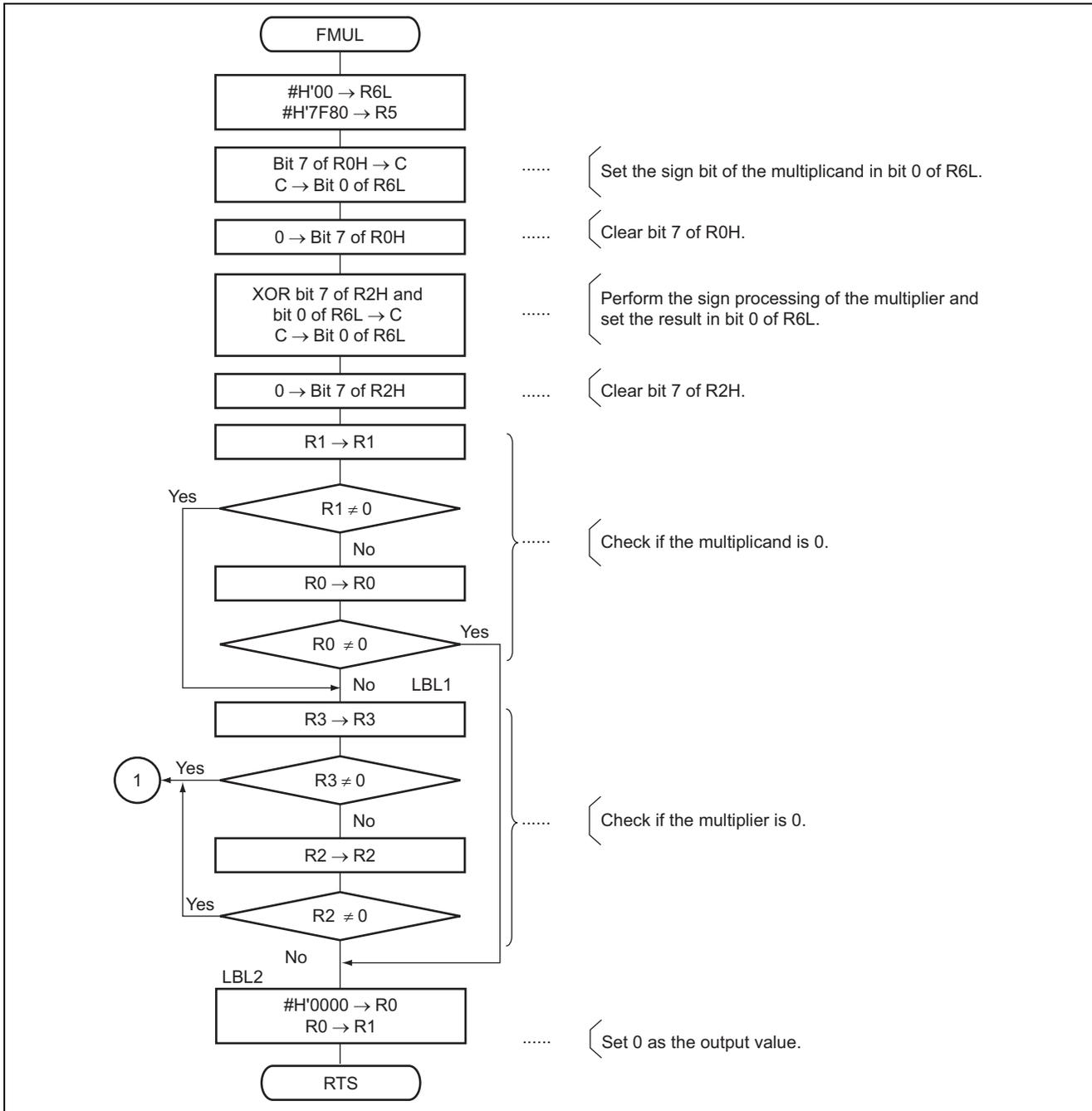
In the floating-point data format, H'7F (D'127) is added to the actual exponent, so the equation will be as follows.

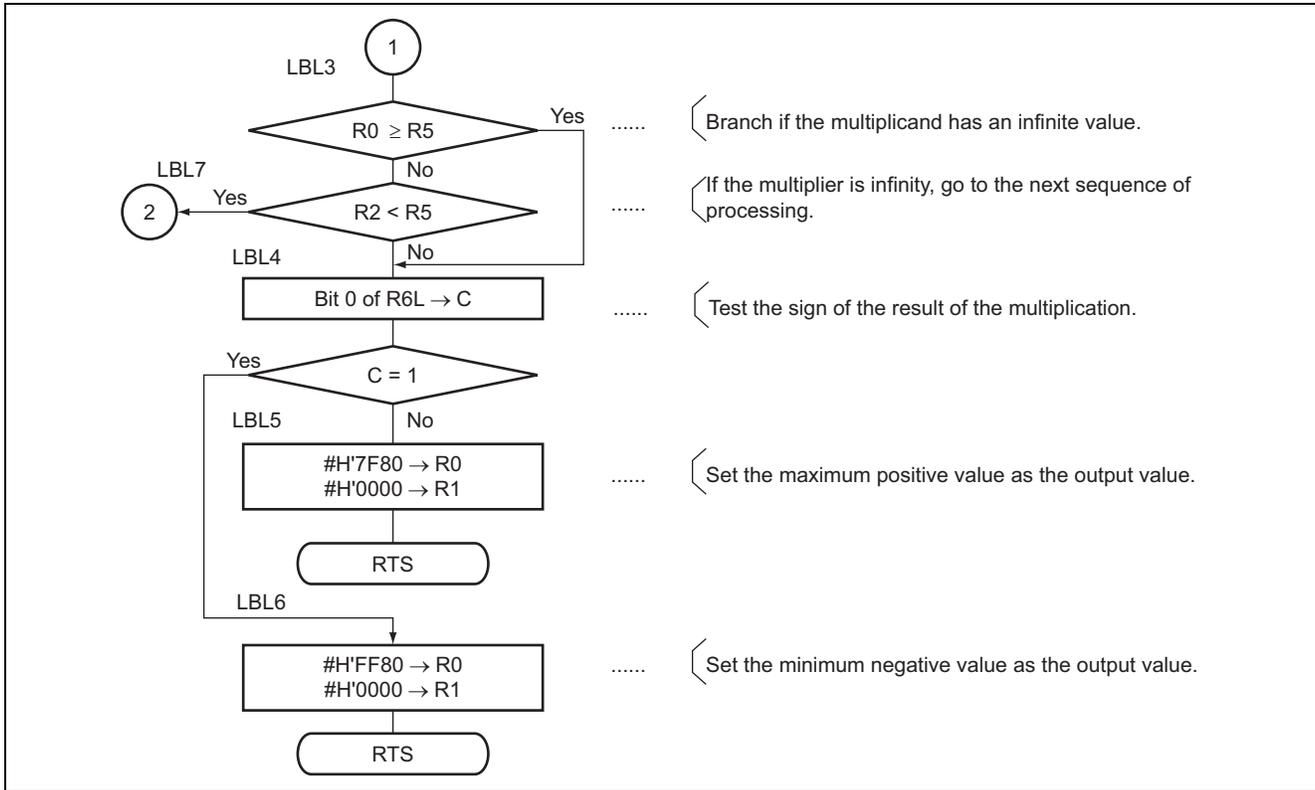
$$R1 \times R2 = (-1)^{S1+S2} \times 2^{\alpha1+\alpha2-127} \times \beta1 \times \beta2$$

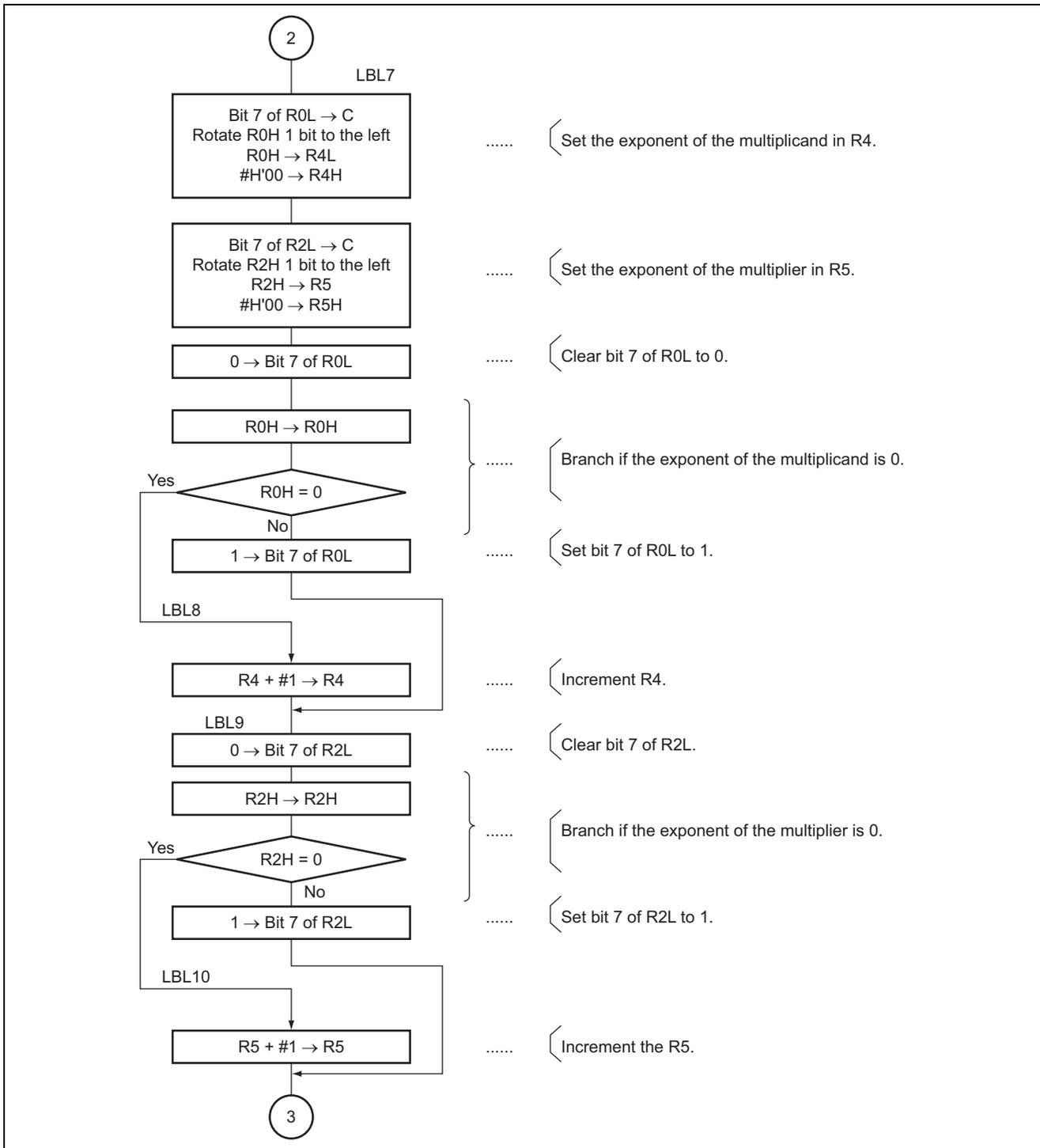
Multiplication according to this equation is carried out in the following sequence.

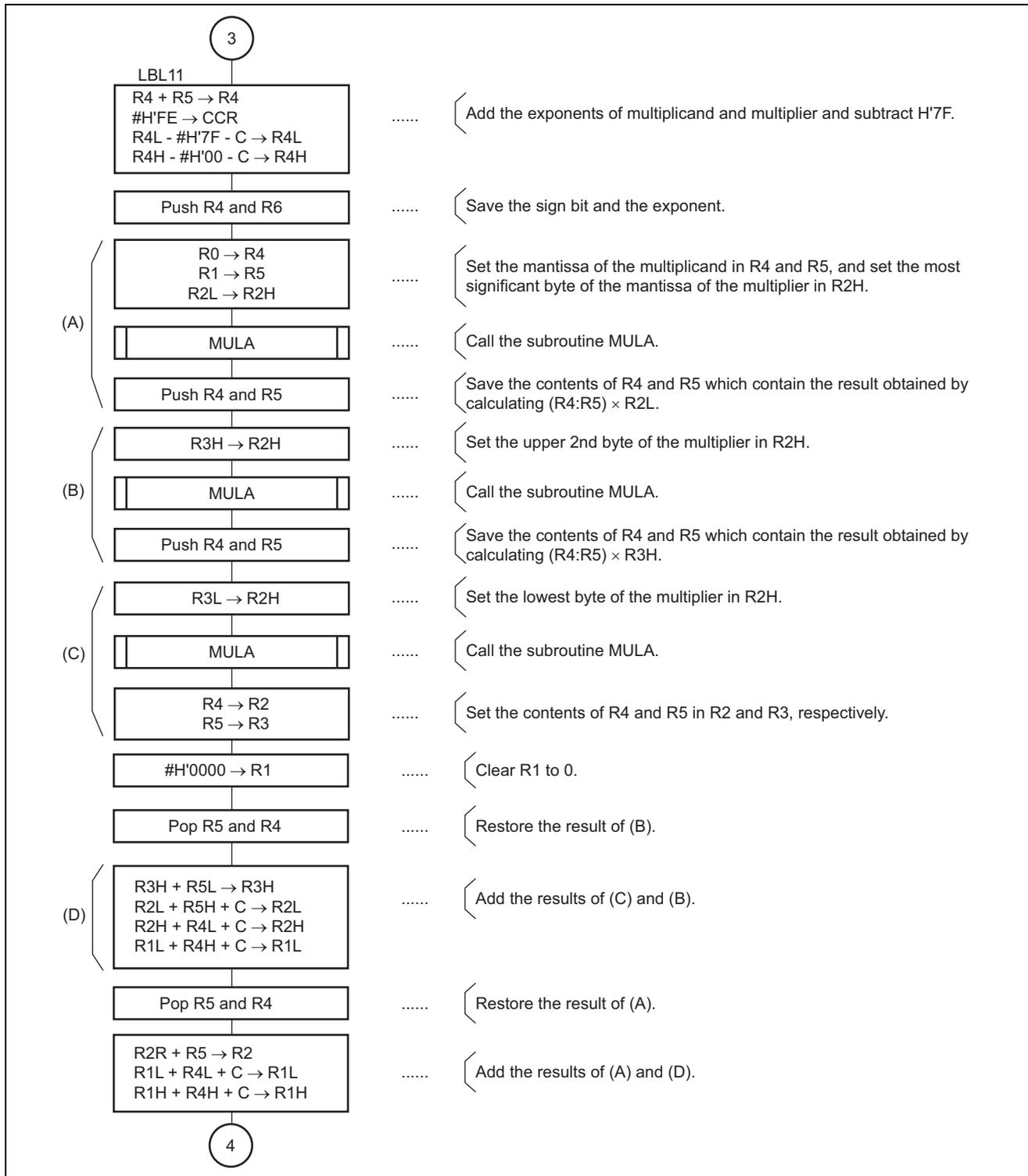
- 1) The exponents are added to each other.
H'7F (D'127) is added to the actual exponent of a number in the floating-point data format; H'7F (D'127) is thus subtracted from both α1 and α2, and H'7F (D'127) is added to the exponent of the result. The result may thus be expressed as follows.
 $(\alpha1-H'7F) + (\alpha2-H'7F) + H'7F = \alpha1 + \alpha2 -H'7F$
One is added to the exponent of a number in denormalized format before the calculation.
- 2) The mantissas are multiplied by each other.
The implicit MSB is included in the multiplication.
For a number in the denormalized format, the implicit MSB of the mantissa is taken to be zero.
- 3) The result of multiplication is corrected to produce a number in the floating-point data format.

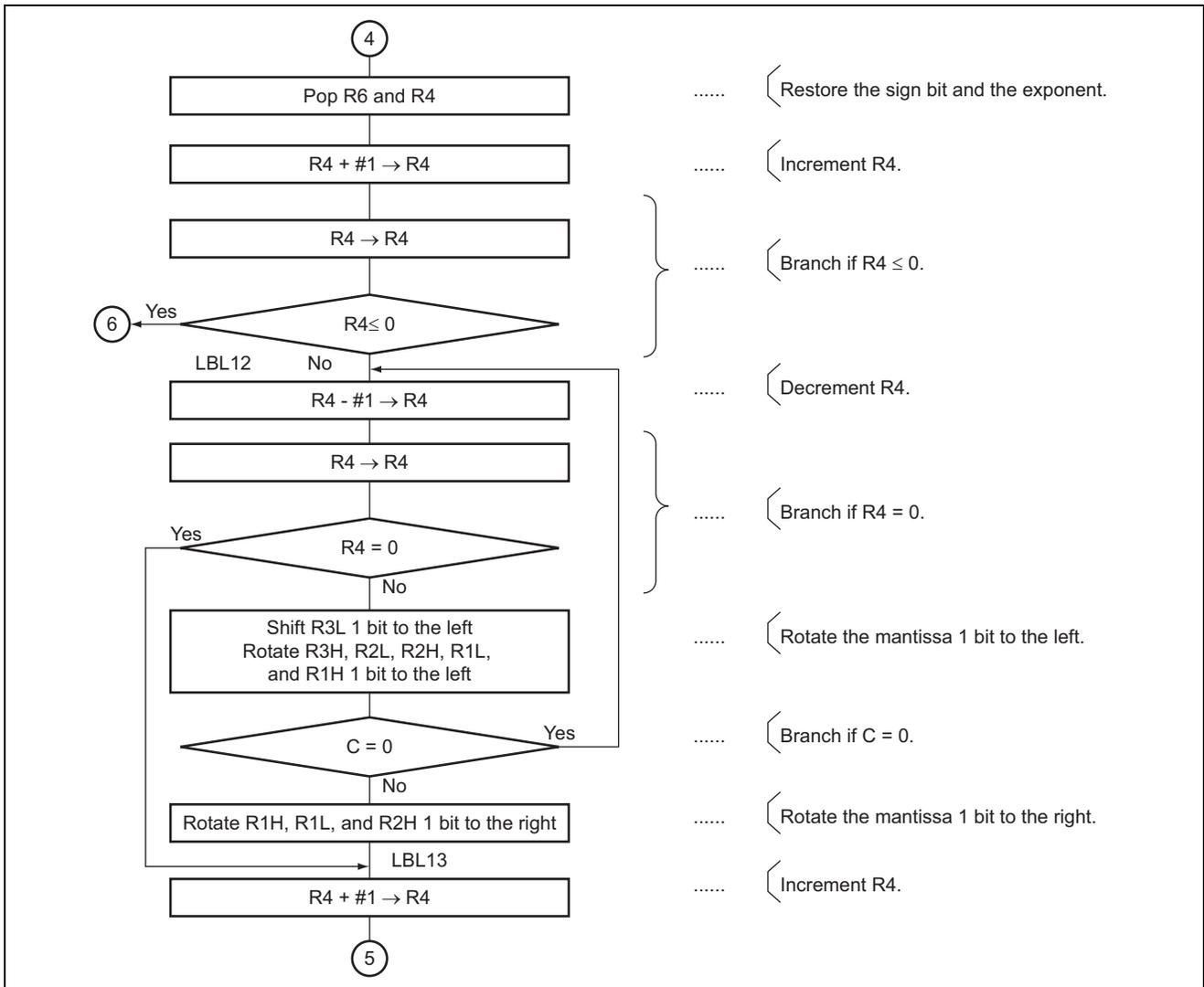
7. Flowchart

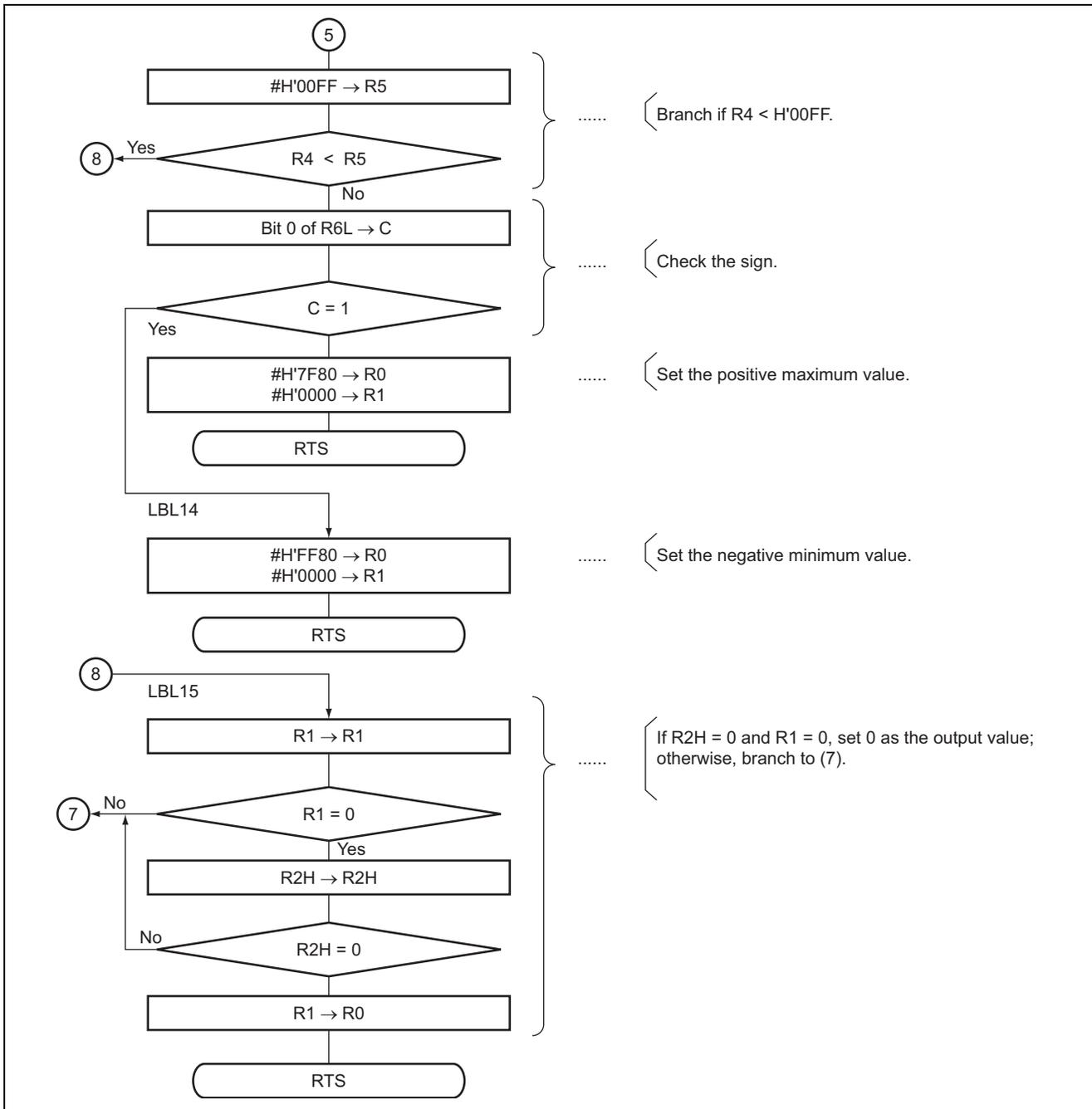


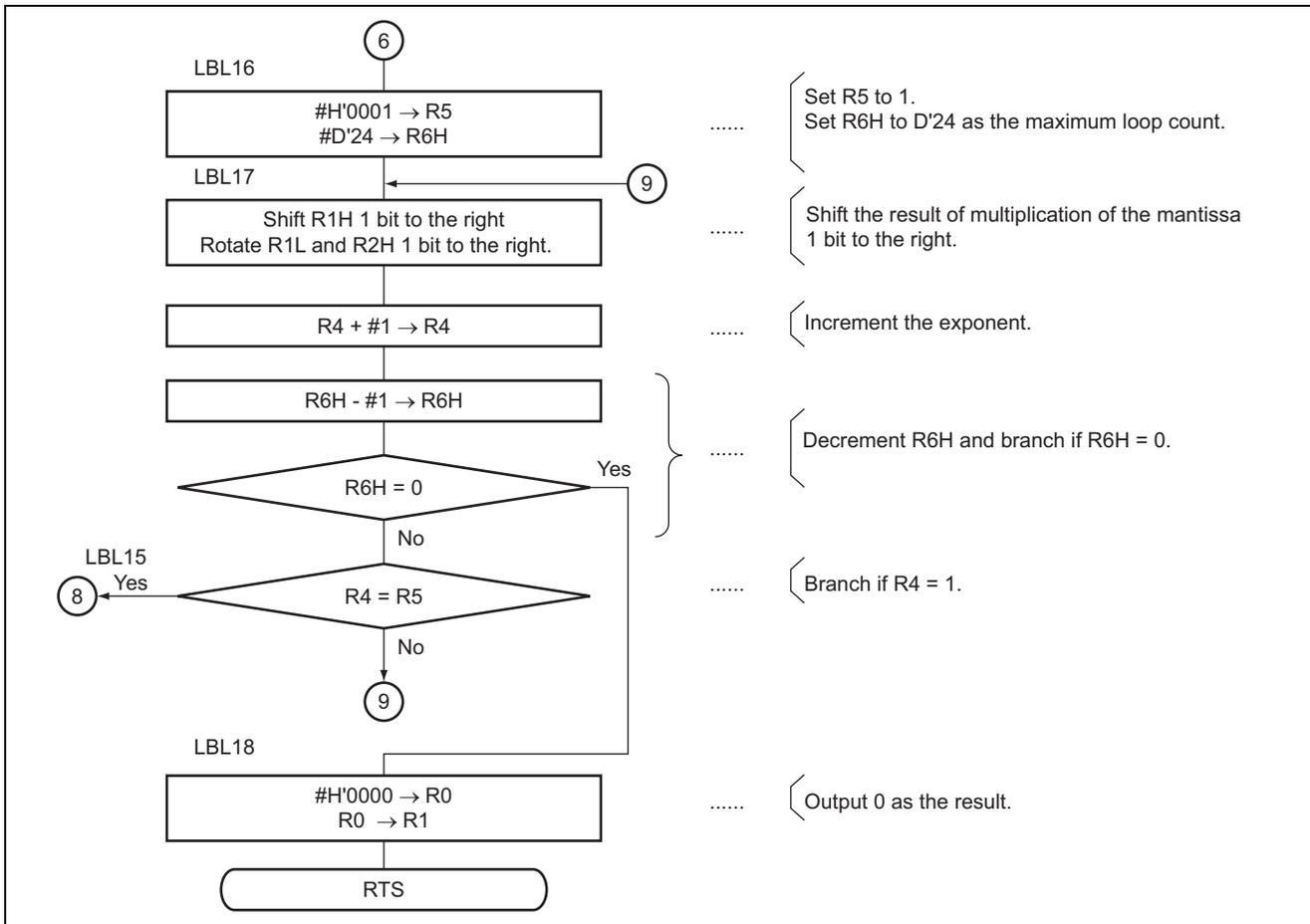


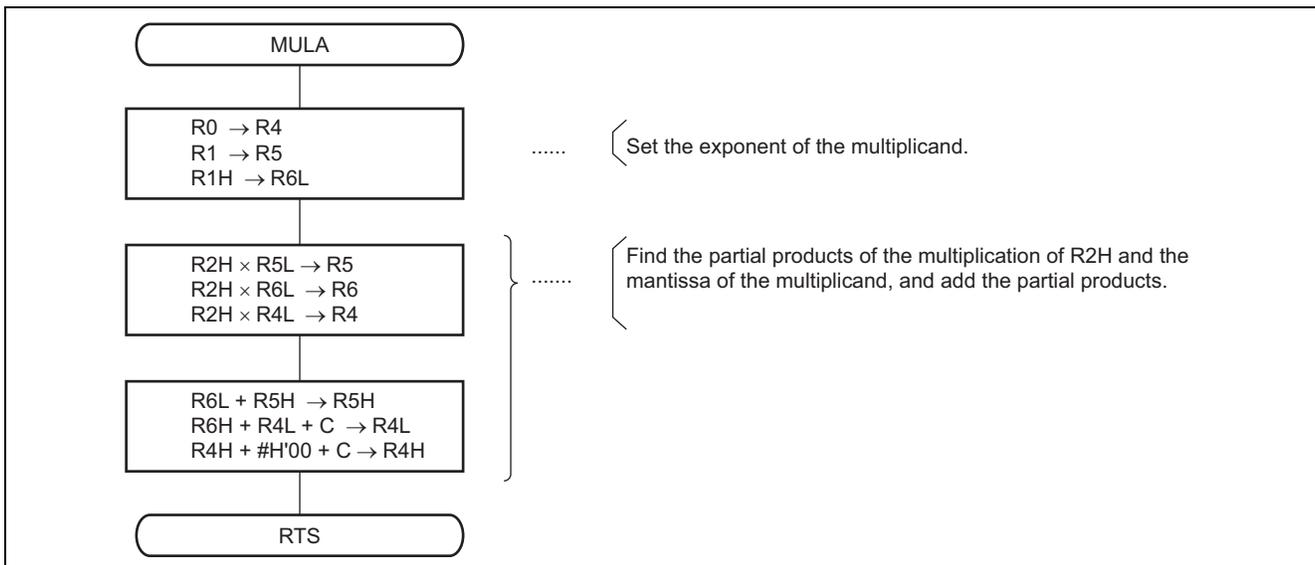
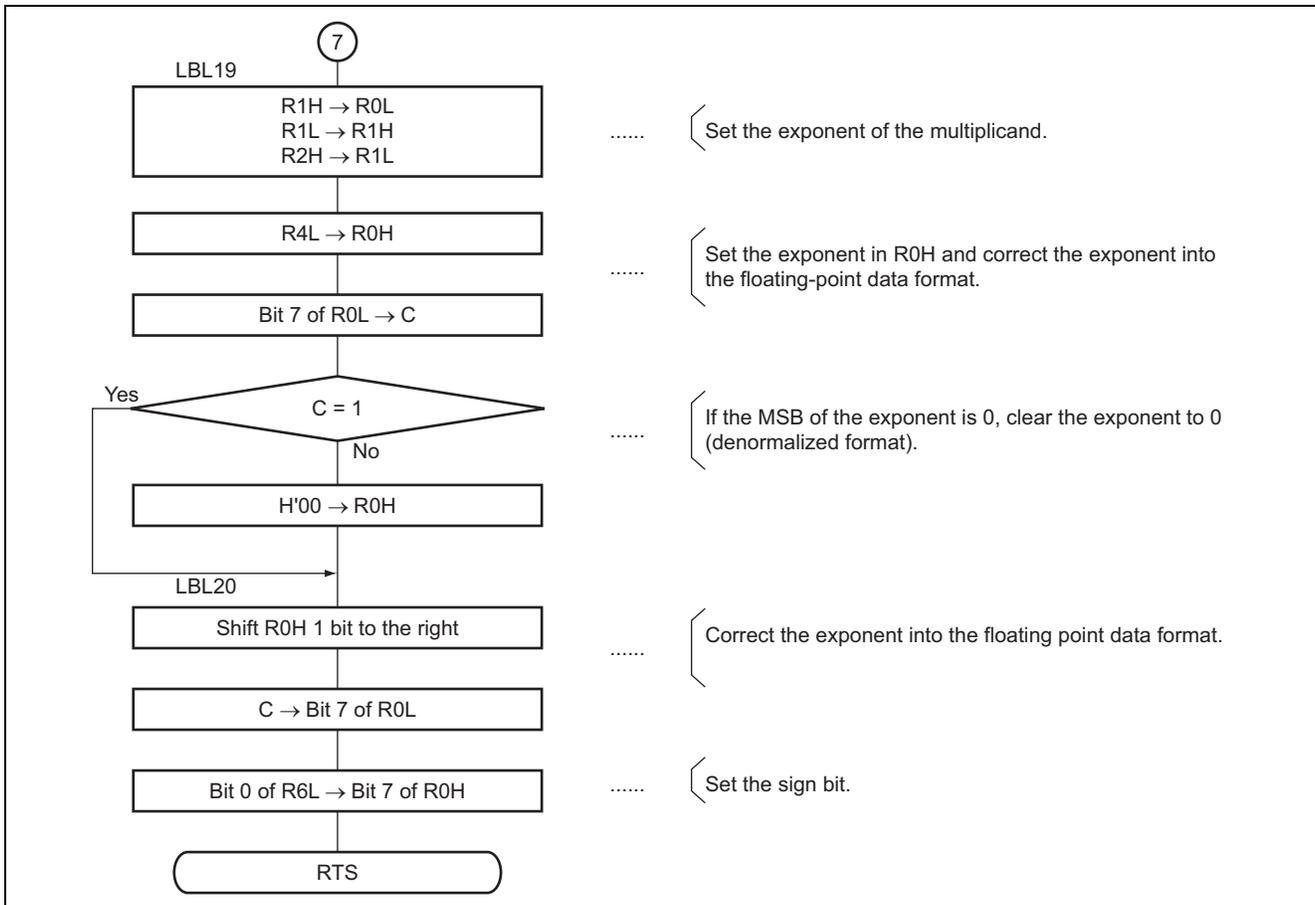












8. Program Listing

```

1          1          ;*****
2          2          ;*
3          3          ;*      NAME : FLOATING POINT MULTIPLICATION (FMUL)
4          4          ;*
5          5          ;*****
6          6          ;*
7          7          ;*      ENTRY:  R0      (HIGHER WORD OF MULTIPLICAND)
8          8          ;*      R1      (LOWER WORD OF MULTIPLICAND)
9          9          ;*      R2      (HIGHER WORD OF MULTIPLIER)
10         10         ;*      R3      (LOWER WORD OF MULTIPLIER)
11         11         ;*
12         12         ;*      RETURNS: R0      (HIGHER WORD OF RESULT)
13         13         ;*      R1      (LOWER WORD OF RESULT)
14         14         ;*
15         15         ;*****
16         16         ;
17         17         .CPU      300HN
18 0000     18         .SECTION  FMUL_code, CODE, ALIGN=2
19         19         .EXPORT   FMUL
20         20         ;
21         21         FMUL     .EQU      $          ;Entry point
22 0000 FE00     22         MOV.B    #H'00,R6L    ;Clear R6L
23 0002 79057F80 23         MOV.W    #H'7F80,R5    ;Set H'7F80
24         24         ;
25 0006 7770     25         BLD      #7,R0H      ;Set sign bit of multiplicand
26 0008 670E     26         BST      #0,R6L      ; to bit 0 of R6L
27 000A 7270     27         BCLR     #7,R0H      ;Bit clear bit 7 of R0H
28         28         ;
29 000C 7772     29         BLD      #7,R2H      ;
30 000E 750E     30         BXOR     #0,R6L      ;Set sign bit of result
31 0010 670E     31         BST      #0,R6L      ; to bit 0 of R6L
32 0012 7272     32         BCLR     #7,R2H      ;Bit clear bit 7 of R2H
33         33         ;
34 0014 0D11     34         MOV.W    R1,R1      ;
35 0016 4604     35         BNE      LBL1
36 0018 0D00     36         MOV.W    R0,R0
37 001A 4708     37         BEQ      LBL2      ;Branch if R1=R=0
38 001C         38         LBL1
39 001C 0D33     39         MOV.W    R3,R3      ;
40 001E 460C     40         BNE      LBL3      ;Branch if not R3 = 0
41 0020 0D22     41         MOV.W    R2,R2
42 0022 4608     42         BNE      LBL3      ;Branch if not R2 = 0
43 0024         43         LBL2
44 0024 79000000 44         MOV.W    #H'0000,R0    ;Set 0 as result
45 0028 0D01     45         MOV.W    R0,R1
46 002A 5470     46         RTS
47 002C         47         LBL3
48 002C 1D05     48         CMP.W    R0,R5
49 002E 4304     49         BLS      LBL4      ;Branch if R0>=R5
50 0030 1D25     50         CMP.W    R2,R5
51 0032 4218     51         BHI      LBL7      ;Branch if R2>=R5
52         52         ;

```

```

53  0034          53  LBL4
54  0034 770E    54          BLD      #0,R6L      ;Load sign bit
55  0036 450A    55          BCS      LBL6        ;Branch if C=1
56  0038          56  LBL5
57  0038 79007F80 57          MOV.W   #H'7F80,R0 ;Set #H'7F800000 as result
58  003C 79010000 58          MOV.W   #H'0000,R1
59  0040 5470    59          RTS
60                60  ;
61  0042          61  LBL6
62  0042 7900FF80 62          MOV.W   #H'FF80,R0 ;Set #H'FF800000 as result
63  0046 79010000 63          MOV.W   #H'0000,R1
64  004A 5470    64          RTS
65                65  ;
66  004C          66  LBL7
67  004C 7778    67          BLD      #7,R0L
68  004E 1200    68          ROTXL   R0H
69  0050 0C0C    69          MOV.B   R0H,R4L      ;Set exponent of multiplicand in R4
70  0052 F400    70          MOV.B   #H'00,R4H
71                71  ;
72  0054 777A    72          BLD      #7,R2L
73  0056 1202    73          ROTXL   R2H
74  0058 0C2D    74          MOV.B   R2H,R5L      ;Set exponent of multiplier in R5
75  005A F500    75          MOV.B   #H'00,R5H
76                76  ;
77  005C 7278    77          BCLR    #7,R0L      ;Clear bit 7 of R0L
78  005E 0C00    78          MOV.B   R0H,R0H
79  0060 4704    79          BEQ     LBL8        ;Branch if multiplier is denormalized
80  0062 7078    80          BSET    #7,R0L      ;Set implicit MSB
81  0064 4004    81          BRA     LBL9        ;Branch always
82  0066          82  LBL8
83  0066 79140001 83          ADD.W   #1,R4
84  006A          84  LBL9
85  006A 727A    85          BCLR    #7,R2L      ;Clear bit 7 of R2L
86  006C 0C22    86          MOV.B   R2H,R2H
87  006E 4704    87          BEQ     LBL10       ;Branch if multiplier is denormalized
88  0070 707A    88          BSET    #7,R2L      ;Set implicit MSB
89  0072 4004    89          BRA     LBL11       ;Branch always
90  0074          90  LBL10
91  0074 79150001 91          ADD.W   #1,R5
92                92  ;
93  0078          93  LBL11
94  0078 0954    94          ADD.W   R5,R4        ;addition exponents
95  007A 06FE    95          ANDC   #H'FE,CCR    ;Clear C flag of CCR
96  007C BC7F    96          SUBX.B #H'7F,R4L    ;R4L - #H'7F - C -> R4L
97  007E B400    97          SUBX.B #H'00,R4H
98                98  ;
99  0080 6DF4    99          PUSH   R4           ;Push R4
100 0082 6DF6   100         PUSH   R6           ;Push R6
101                101  ;
102 0084 0D04   102         MOV.W  R0,R4
103 0086 0D15   103         MOV.W  R1,R5
104                104  ;
105 0088 0CA2   105         MOV.B  R2L,R2H
106 008A 5E000000 106        JSR    @MULA        ;R2L * (R0L:R1) -> (R4:R5)

```

```

107 008E 6DF4      107          PUSH      R4          ;Push R4
108 0090 6DF5      108          PUSH      R5          ;Push R5
109
110 0092 0C32      110          MOV.B     R3H,R2H     ;
111 0094 5E000000   111          JSR      @MULA       ;R3L * (R0L:R1) -> (R4:R5)
112 0098 6DF4      112          PUSH      R4          ;Push R4
113 009A 6DF5      113          PUSH      R5          ;Push R5
114
115 009C 0CB2      115          MOV.B     R3L,R2H     ;R3L * (R0L:R1) -> (R4:R5)
116 009E 5E000000   116          JSR      @MULA       ;Push R4
117 00A2 0D42      117          MOV.W     R4,R2       ;Push R5
118 00A4 0D53      118          MOV.W     R5,R3
119
120 00A6 79010000   120          MOV.W     #H'0000,R1 ;Clear R1
121 00AA 6D75      121          POP       R5          ;Pop R5
122 00AC 6D74      122          POP       R4          ;Pop R4
123
124 00AE 08D3      124          ADD.B     R5L,R3H     ;R3H + R5L + C -> R3H
125 00B0 0E5A      125          ADDX.B    R5H,R2L     ;R2L + R5H + C -> R2L
126 00B2 0EC2      126          ADDX.B    R4L,R2H     ;R2H + R4L + C -> R2H
127 00B4 0E49      127          ADDX.B    R4H,R1L     ;R1L + R4H + C -> R1L
128
129 00B6 6D75      129          POP       R5          ;Pop R5
130 00B8 6D74      130          POP       R4          ;Pop R4
131 00BA 0952      131          ADD.W     R5,R2       ;R2 + R5 -> R2L
132 00BC 0EC9      132          ADDX.B    R4L,R1L     ;R1L + R4L + C -> R1L
133 00BE 0E41      133          ADDX.B    R4H,R1H     ;R1H + R4H + C -> R1H
134
135 00C0 6D76      135          POP       R6          ;Pop R6
136 00C2 6D74      136          POP       R4          ;Pop R4
137 00C4 79140001   137          ADD.W     #1,R4
138 00C8 0D44      138          MOV.W     R4,R4
139
140 00CA 474E      140          BEQ      LBL16       ;Branch if R4=0
141 00CC 4B4C      141          BMI      LBL16       ;Branch if R4<0
142 00CE
143 00CE 79340001   143          SUB.W     #1,R4
144 00D2 0D44      144          MOV.W     R4,R4
145 00D4 4714      145          BEQ      LBL13       ;Branch if R4=0
146 00D6 100B      146          SHLL     R3L         ;Shift mantissa 1 bit left
147 00D8 1203      147          ROTXL    R3H
148 00DA 120A      148          ROTXL    R2L
149 00DC 1202      149          ROTXL    R2H
150 00DE 1209      150          ROTXL    R1L
151 00E0 1201      151          ROTXL    R1H
152 00E2 44EA      152          BCC      LBL12       ;Branch if C=0
153 00E4 1301      153          ROTXR    R1H         ;Rotate mantissa 1 bit right
154 00E6 1309      154          ROTXR    R1L
155 00E8 1302      155          ROTXR    R2H
156 00EA
157 00EA 79140001   157          ADD.W     #1,R4
158
159 00EE 790500FF   159          MOV.W     #H'00FF,R5
160 00F2 1D45      160          CMP.W     R4,R5

```

```

161 00F4 4418          161          BCC      LBL15      ;Branch if R5>R4
162 00F6 770E          162          BLD      #0,R6L    ;Load sign bit
163 00F8 450A          163          BCS      LBL14      ;Branch if C=1
164 00FA 79007F80      164          MOV.W   #H'7F80,R0 ;Set H'7F800000 to result
165 00FE 79010000      165          MOV.W   #H'0000,R1
166 0102 5470          166          RTS
167                    167          ;
168 0104              168          LBL14
169 0104 7900FF80      169          MOV.W   #H'FF80,R0 ;Set H'FF800000 to product
170 0108 79010000      170          MOV.W   #H'0000,R1
171 010C 5470          171          RTS
172 010E              172          LBL15
173 010E 0D11          173          MOV.W   R1,R1
174 0110 462A          174          BNE     LBL19      ;Branch if not R1=0
175 0112 0C22          175          MOV.B   R2H,R2H
176 0114 4626          176          BNE     LBL19      ;Branch if not R2H=0
177 0116 0D10          177          MOV.W   R1,R0
178 0118 5470          178          RTS
179                    179          ;
180 011A              180          LBL16
181 011A 79050001      181          MOV.W   #H'0001,R5 ;Set #H'0001 to R5
182 011E F618          182          MOV.B   #D'24,R6H ;Set bit counter
183 0120              183          LBL17
184 0120 1101          184          SHLR   R1H        ;Shift mantissa 1 bit right
185 0122 1309          185          ROTXR  R1L
186 0124 1302          186          ROTXR  R2H
187 0126 79140001      187          ADD.W   #1,R4      ;Increment exponent
188 012A 1A06          188          DEC.B   R6H        ;Decrement bit counter
189 012C 4706          189          BEQ     LBL18      ;Branch if Z=1
190 012E 1D54          190          CMP.W   R5,R4
191 0130 47DC          191          BEQ     LBL15      ;Branch if R5=R4
192 0132 40EC          192          BRA     LBL17      ;Branch always
193 0134              193          LBL18
194 0134 79000000      194          MOV.W   #H'0000,R0 ;Clear result
195 0138 0D01          195          MOV.W   R0,R1
196 013A 5470          196          RTS
197                    197          ;
198 013C              198          LBL19
199 013C 0C18          199          MOV.B   R1H,R0L
200 013E 0C91          200          MOV.B   R1L,R1H
201 0140 0C29          201          MOV.B   R2H,R1L
202                    202          ;
203 0142 0CC0          203          MOV.B   R4L,R0H
204 0144 7778          204          BLD      #7,R0L
205 0146 4502          205          BCS      LBL20      ;Branch if C=1
206 0148 F000          206          MOV.B   #H'00,R0H
207 014A              207          LBL20              ;Correct into floating-point format
208 014A 1100          208          SHLR   R0H
209 014C 6778          209          BST     #7,R0L
210 014E 770E          210          BLD      #0,R6L
211 0150 6770          211          BST     #7,R0H
212 0152 5470          212          RTS
213                    213          ;
214                    214          ;-----

```

```

215          215      ;
216  0154          216      MULA          ;R2H * (R0L:R1) -> (R4:R5)
217  0154 0D04    217          MOV.W      R0,R4      ;R0  -> R4
218  0156 0D15    218          MOV.W      R1,R5      ;R1  -> R5
219  0158 0C1E    219          MOV.B      R1H,R6L     ;R1H -> R6L
220          220      ;
221  015A 5025    221          MULXU      R2H,R5      ;R2H * R5L -> R5
222  015C 5026    222          MULXU      R2H,R6      ;R2H * R6L -> R6
223  015E 5024    223          MULXU      R2H,R4      ;R2H * R4L -> R4
224          224      ;
225  0160 08E5    225          ADD.B      R6L,R5H      ;R5H + R6L      -> R5H
226  0162 0E6C    226          ADDX.B     R6H,R4L      ;R4L + R6H  + C -> R4L
227  0164 9400    227          ADDX.B     #H'00,R4H    ;R4H + #H'00 + C -> R4H
228  0166 5470    228          RTS
229          229      ;
230          230          .END
*****TOTAL ERRORS      0
*****TOTAL WARNINGS    0

```

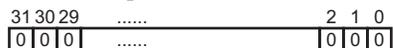
<Reference> Description of Single-Precision Floating-Point Formats

Single-Precision Floating-Point Formats:

1. Internal Representation of Single-Precision Floating Point Numbers

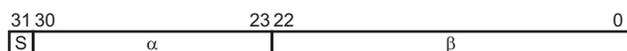
One of the following formats is used depending on the value of the single-precision floating-point data in this application note (a real number is indicated as R).

1) Internal Representation When R=0



All the 32 bits are 0.

2) Normalized Format



α is an index number with an 8-bit-long field. β is a mantissa with a 23-bit-long field. Here, the R value can be represented by the expression below (when $1 \leq \alpha \leq 254$).

↓Implicit MSB

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

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

3) Denormalized Format

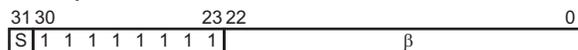


β is a mantissa with a 23-bit-long field. This format is used to represent a real number that is too small to be represented by the normalized format.

Here, the R value can be represented by the expression below.

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

4) Infinity



β is a mantissa with a 23-bit-long field. Note that if all the bits in the index part are 1, the R value is handled as follows, in this application note.

When S = 0: Plus infinity

$$R = +\infty$$

When S = 1: Minus infinity

$$R = -\infty$$

2. Internal Representation Examples

$$S = B'0 \quad (\text{binary})$$

$$\alpha = B'10000011 \quad (\text{binary})$$

$$\beta = B'1011100\dots\dots 0 \quad (\text{binary})$$

Under the above conditions, the corresponding R value is represented as follows.

$$R = 2^0 \times 2^{131-126} \times (1 + 2^{-1} + 2^{-3} + 2^{-4} + 2^{-5})$$

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

1) Maximum and Minimum Values

Here, the maximum and minimum values are absolute values. The maximum value is indicated as R_{MAX} and the minimum value is indicated as R_{MIN} . Up to the following values can be represented.

$$R_{MAX} = 2^{254-127} \times (1 + 2^{-1} + 2^{-2} + 2^{-3} + \dots + 2^{-23})$$

$$\approx 3.27 \times 10^{38}$$

$$R_{MIN} = 2^{-126} \times 2^{-23} = 2^{-140} \approx 1.40 \times 10^{-45}$$

Revision Record

Rev.	Date	Description	
		Page	Summary
2.00	Feb.28.06	—	Format has been changed from Hitachi version to Renesas version.
3.00	Jun.12.06	6	Error correction

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage. Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
 The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
 Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (<http://www.renesas.com>).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
 Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.