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.

RENESAS

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

Conversion from Signed 32-bit Binary to Single-Precision Floating-Point (KFTR)

Introduction

Converts a signed 32-bit binary number in general registers to a single-precision floating-point number.

Target Device

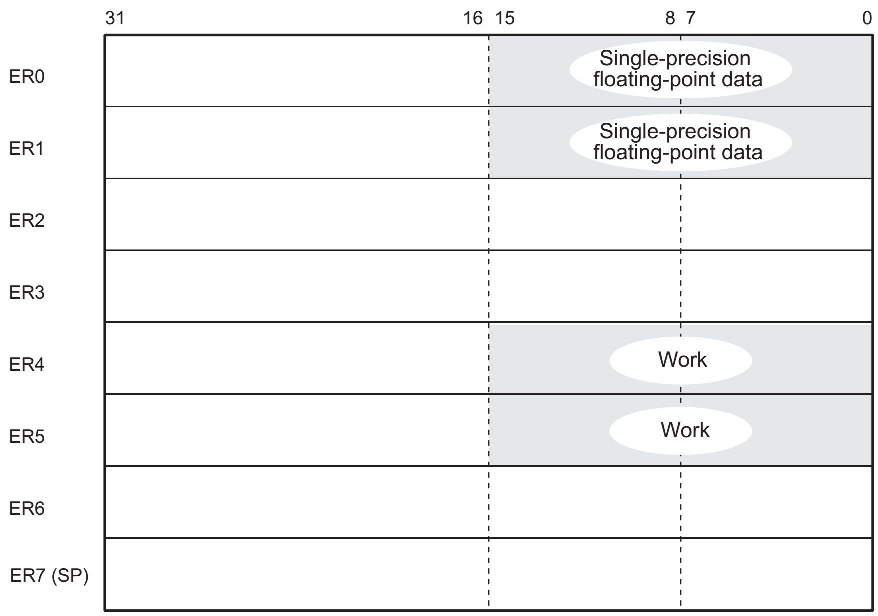
H8/300H Tiny Series

Contents

1. Arguments

Contents		Storage Location	Data Length (Bytes)
Input	Signed 32-bit binary number	R0, R1	4
Output	Single-precision floating-point number	R0, R1	4

2. Changes to Internal Registers and Flags



I	UI	H	U	N	Z	V	C
—	—	↕	—	↕	↕	↕	↕

— : No change

↕ : Varies

0 : Fixed to 0

1 : Fixed to 1

3. Programming Specifications

Program memory (bytes)	98
Data memory (bytes)	0
Stack (bytes)	0
Number of cycles	346
Re-entrant	Yes
Relocatable	Yes
Interrupts during execution	Yes

4. Notes

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

5. Descriptions

5.1 Descriptions of Functions

1. The arguments are listed below.
   1) Set the input argument.
      R0: higher-order two bytes of the signed 32-bit binary number
      R1: lower-order two bytes of the signed 32-bit binary number
   2) The KFTR subroutine places the output argument in the following registers.
      R0: higher-order two bytes of the single-precision floating-point number
      R1: lower-order two bytes of the single-precision floating-point number

2. The following figure illustrates the execution of the KFTR subroutine. When the subroutine is called with the input arguments set as shown below, it places the result of conversion in R0 and R1.

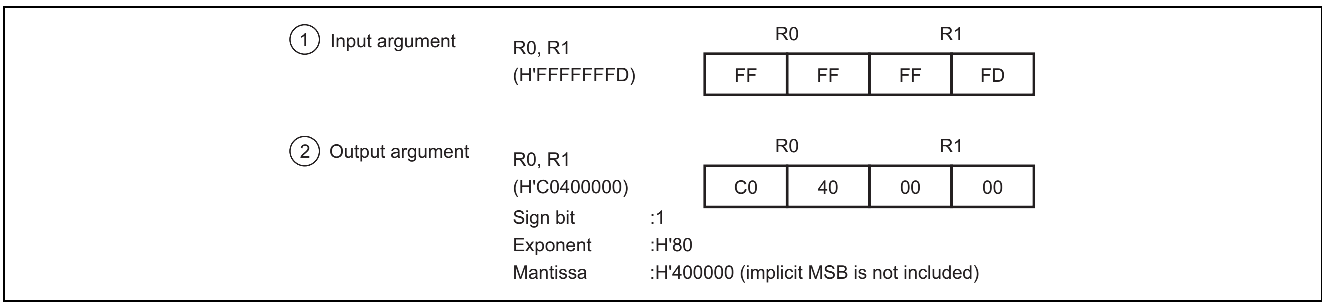


Figure 1 Example of KFTR Execution

5.2 Usage Notes

The signed 32-bit binary number stored in R0 and R1 is lost in the execution of KFTR because the result of conversion is output to R0 and R1. When you will still require this number, save it elsewhere in memory before executing KFTR.

5.3 Description of Data Memory

No data memory is used by KFTR.

5.4 Example of Usage

After setting the signed 32-bit binary number in the general registers, call the KFTR subroutine.

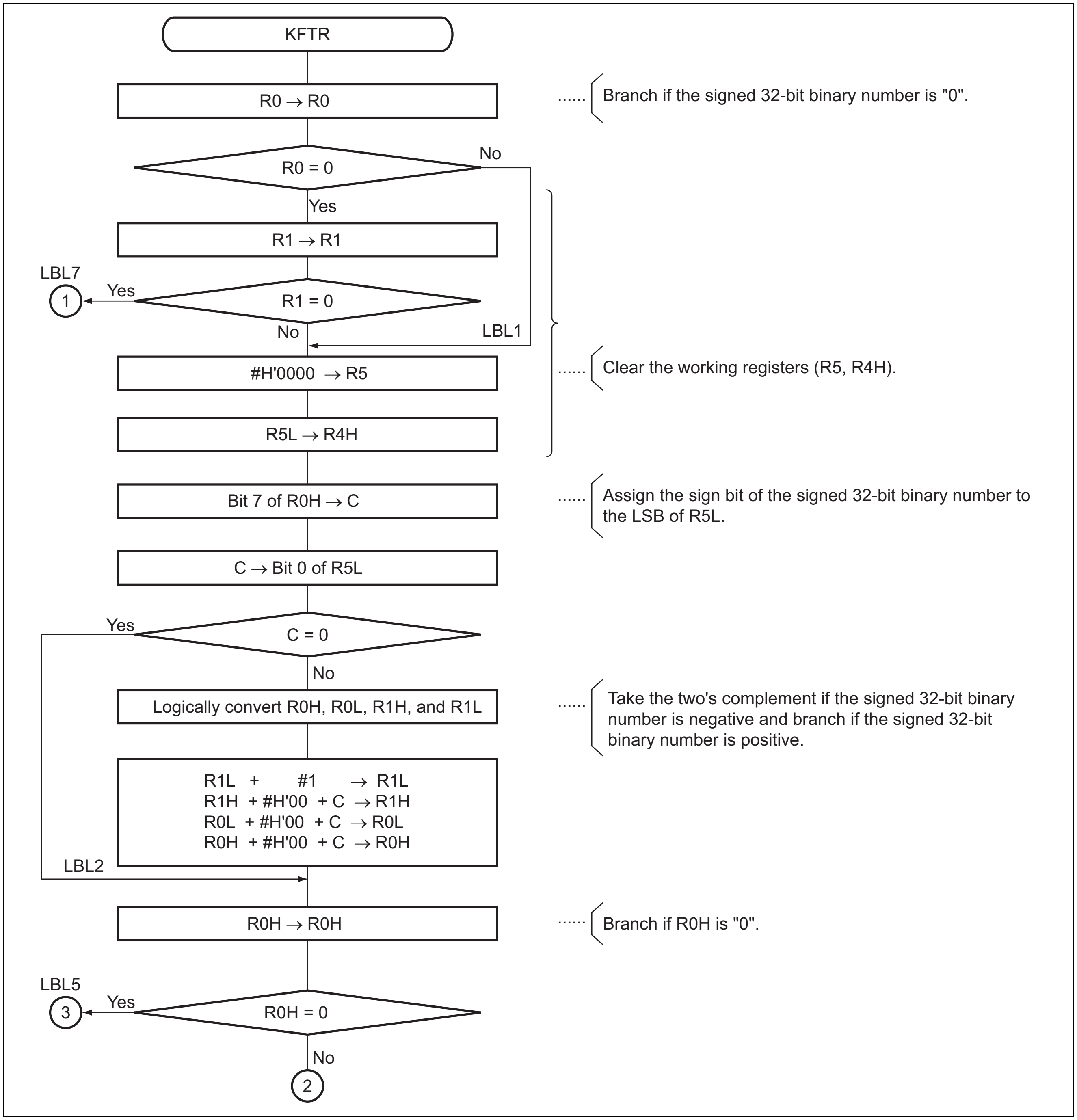
```
WORK1   . RES. W 2,0          ········· Reservation of the data memory area for setting of the signed 32-bit binary number by the user
                                        program.
WORK2   . RES. W 2,0          ········· Reservation of the data memory area where the single-precision floating-point number will be set.
            .
            .
            .
        MOV. W @WORK1, R0     ········· Sets, as the input argument, the signed 32-bit binary number specified by the user program
        MOV. W @WORK1+2, R1
        JSR    @KFTR          ········· Subroutine call of KFTR
        MOV. W R0, @WORK2     ········· Transfers the single-precision floating-point result from the output argument to data memory.
        MOV. W R1, @WORK2+2
```

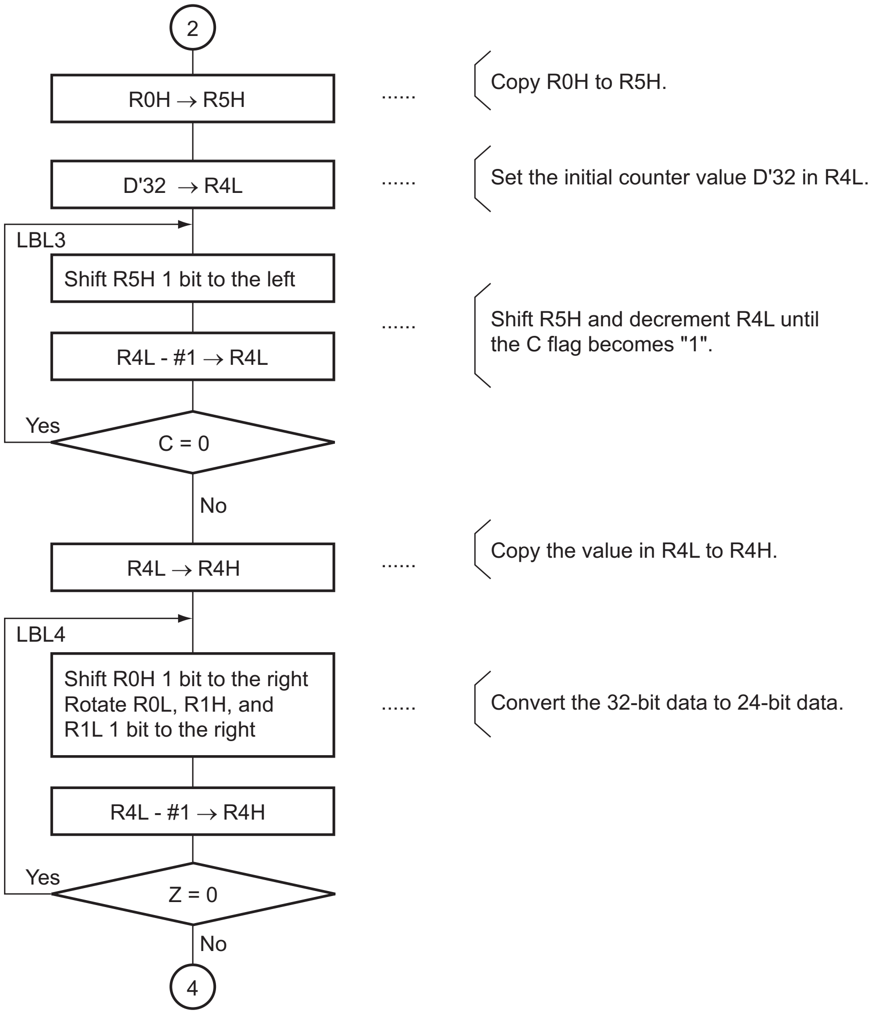
5.5 Principles of Operation

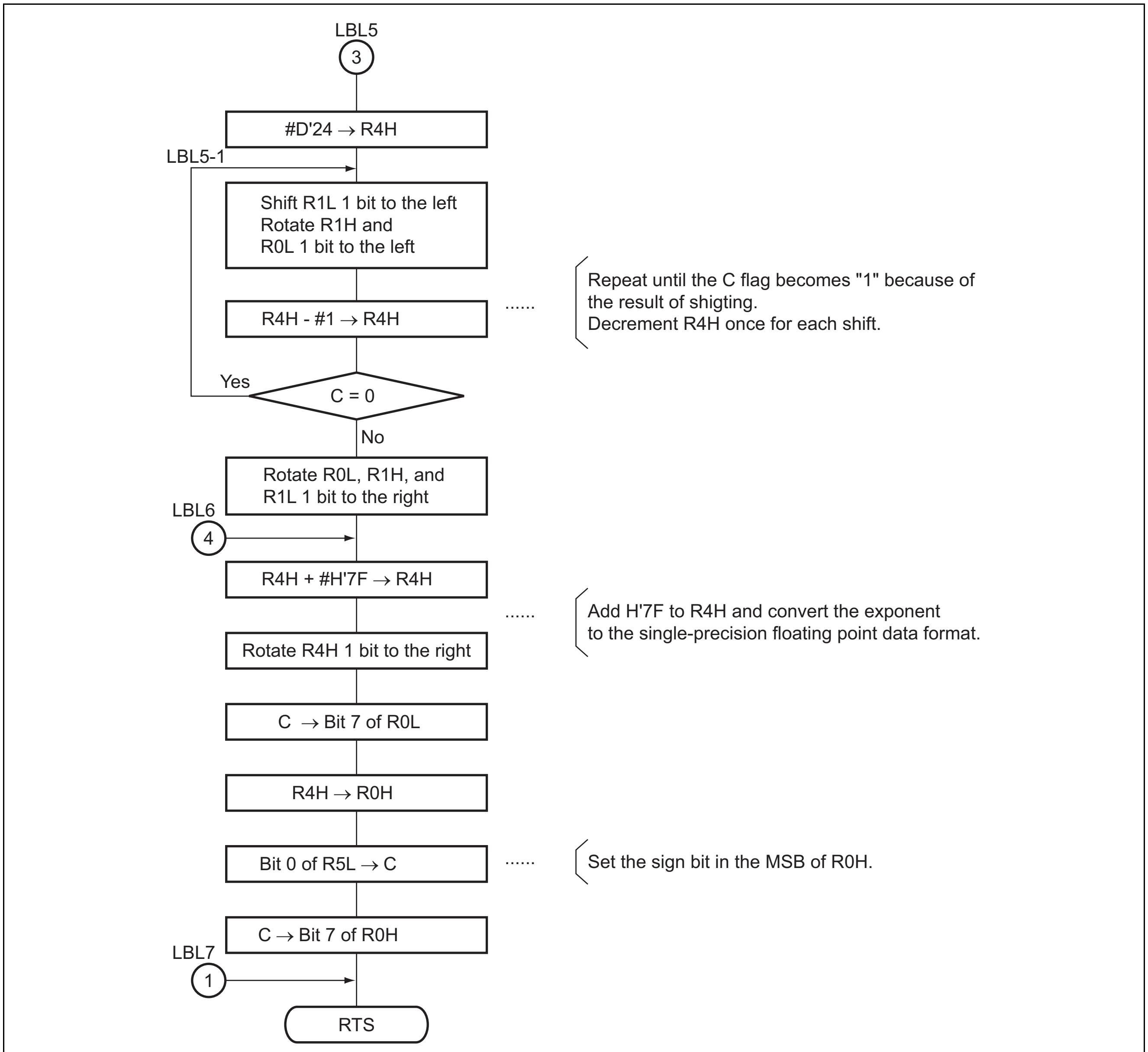
The KFTR subroutine checks whether the signed 32-bit binary number is negative or positive, and takes the two's complement if the number is negative. The eight higher-order bits are then checked for a value of H'00.

1. If these bits don’t contain H'00, KFTR shifts the exponent to the right and obtains a 24-bit binary number.
2. If the bits contain H'00, KFTR shifts the exponent to the left, sets the MSB of the 24 lower-order bits to 1, and adds H'7F to the exponent to convert into the floating-point data format.

6. Flowchart


KFTR
R0 → R0
R0 = 0
No
Yes
R1 → R1
LBL7
Yes
1
R1 = 0
No
LBL1
#H'0000 → R5
R5L → R4H
Bit 7 of R0H → C
C → Bit 0 of R5L
Yes
C = 0
No
Logically convert R0H, R0L, R1H, and R1L
R1L + #1 → R1L
R1H + #H'00 + C → R1H
R0L + #H'00 + C → R0L
R0H + #H'00 + C → R0H
LBL2
R0H → R0H
LBL5
Yes
3
R0H = 0
No
2
...... Branch if the signed 32-bit binary number is "0".
...... Clear the working registers (R5, R4H).
...... Assign the sign bit of the signed 32-bit binary number to the LSB of R5L.
...... Take the two's complement if the signed 32-bit binary number is negative and branch if the signed 32-bit binary number is positive.
...... Branch if R0H is "0".


2
R0H → R5H
Copy R0H to R5H.
D'32 → R4L
Set the initial counter value D'32 in R4L.
LBL3
Shift R5H 1 bit to the left
R4L - #1 → R4L
Shift R5H and decrement R4L until the C flag becomes "1".
Yes
C = 0
No
R4L → R4H
Copy the value in R4L to R4H.
LBL4
Shift R0H 1 bit to the right Rotate R0L, R1H, and R1L 1 bit to the right
Convert the 32-bit data to 24-bit data.
R4L - #1 → R4H
Yes
Z = 0
No
4


LBL5
3
#D'24 → R4H
LBL5-1
Shift R1L 1 bit to the left
Rotate R1H and
R0L 1 bit to the left
R4H - #1 → R4H
......
Repeat until the C flag becomes "1" because of
the result of shigting.
Decrement R4H once for each shift.
Yes
C = 0
No
Rotate R0L, R1H, and
R1L 1 bit to the right
LBL6
4
R4H + #H'7F → R4H
......
Add H'7F to R4H and convert the exponent
to the single-precision floating point data format.
Rotate R4H 1 bit to the right
C → Bit 7 of R0L
R4H → R0H
Bit 0 of R5L → C
......
Set the sign bit in the MSB of R0H.
C → Bit 7 of R0H
LBL7
1
RTS

7. Program Listing

```
                      ;************************************************************************
                      ;*                                                                      *
                      ;*          NAME : CHANGE 32 BIT BINARY TO FLOATING POINT               *
                      ;*                 (KFTR)                                               *
                      ;*                                                                      *
                      ;************************************************************************
                      ;*                                                                      *
                      ;*        ENTRY:  R0     (UPPER WORD OF 32 BIT BINARY)                  *
                      ;*                R1     (LOWER WORD OF 32 BIT BINARY)                  *
                      ;*                                                                      *
                      ;*       RETURNS: R0     (UPPER WORD OF FLOATING BINARY)                *
                      ;*                R1     (LOWER WORD OF FLOATING BINARY)                *
                      ;*                                                                      *
                      ;************************************************************************
                      ;
                              .CPU        300HN
0000                          .SECTION    KFTR_code,CODE,ALIGN=2
                              .EXPORT     KFTR
                      ;
     00000000         KFTR    .EQU        $           ;Entry point
0000 0D00                     MOV.W       R0,R0
0002 4604                     BNE         LBL1
0004 0D11                     MOV.W       R1,R1
0006 4758                     BEQ         LBL7        ;Branch if R0=R1=0
0008                  LBL1
0008 79050000                 MOV.W       #H'0000,R5  ;Clear R5
000C 0CD4                     MOV.B       R5L,R4H     ;Clear R4H
000E 7770                     BLD         #7,R0H
0010 670D                     BST         #0,R5L      ;Set sign bit to bit 0 of R5L
0012 4410                     BCC         LBL2        ;Branch if 32 bit binary is negative
0014 1700                     NOT         R0H         ;2's complement 32 bit binary
0016 1708                     NOT         R0L
0018 1701                     NOT         R1H
001A 1709                     NOT         R1L
001C 8901                     ADD.B       #H'01,R1L
001E 9100                     ADDX.B      #H'00,R1H
0020 9800                     ADDX.B      #H'00,R0L
0022 9000                     ADDX.B      #H'00,R0H
0024                  LBL2
0024 0C00                     MOV.B       R0H,R0H
0026 471A                     BEQ         LBL5        ;Branch if R0H=0
0028 0C05                     MOV.B       R0H,R5H
002A FC20                     MOV.B       #D'32,R4L   ;Set bit counter1
002C                  LBL3
002C 1005                     SHLL.B      R5H         ;Shift R5H 1 bit left
002E 1A0C                     DEC.B       R4L         ;Decrement R4L
0030 4410                     BCC         LBL5        ;Branch if C=0
0032 0CC4                     MOV.B       R4L,R4H     ;Push R4L to R4H
0034                  LBL4
0034 1100                     SHLR.B      R0H         ;Change 32 bit binary to mantissa
0036 1208                     ROTXL.B     R0L
0038 1201                     ROTXL.B     R1H
003A 1209                     ROTXL.B     R1L
003C 1A0C                     DEC.B       R4L         ;Decrement bit counter 1
003E 46F4                     BNE         LBL4        ;Branch if Z=0
0040 4012                     BRA         LBL6        ;Branch always
```

```
                  ;
     0042         LBL5
     0042 F418            MOV.B     #D'24,R4H   ;Set bit counter 2
     0044         LBL5_1
     0044 1009            SHLL.B    R1L         ;Change 32 bit binary to mantissa
     0046 1201            ROTXL.B   R1H
     0048 1208            ROTXL.B   R0L
     004A 1A04            DEC.B     R4H         ;Decrement bit counter 2
     004C 44F6            BCC       LBL5_1
     004E 1308            ROTXR.B   R0L         ;Rotate mantissa 1 bit right
     0050 1301            ROTXR.B   R1H
     0052 1309            ROTXR.B   R1L
     0054         LBL6
     0054 847F            ADD.B     #H'7F,R4H   ;Biased exponent
     0056 1104            SHLR.B    R4H         ;Change floating point format
     0058 6778            BST       #7,R0L
     005A 0C40            MOV.B     R4H,R0H
     005C 770D            BLD       #0,R5L
     005E 6770            BST       #7,R0H
     0060         LBL7
     0060 5470            RTS
                  ;
                          .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

31	30	29	......	2	1	0
0	0	0	......	0	0	0

   All the 32 bits are 0.

   2) Normalized Format

31	30 … 23	22 … 0
S	α	β

   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 $1 \le \alpha \le 254$).

   ↓Implicit MSB

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

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

   3) Denormalized Format

31	30 … 23	22 … 0
S	0 0 0 0 0 0 0 0	β

   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.

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

   4) Infinity

31	30 … 23	22 … 0
S	1 1 1 1 1 1 1 1	β

   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 (binary)
   α = B'10000011 (binary)
   β = B'1011100……0 (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.

   $$R_{MAX} = 2^{254-127} \times (1 + 2^{-1} + 2^{-2} + 2^{-3} + \ldots\ldots + 2^{-23})$$
   $$\approx 3.27 \times 10^{38}$$
   $$R_{MIN} = 2^{-126} \times 2^{-23} = 2^{-140} \approx 1.40 \times 10^{-45}$$

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	7, 8	Error correction