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 Five-Digit BCD to Two-Byte Hexadecimal (BCD)

Introduction

Converts the three-byte five-digit BCD (binary coded decimal) number set in general registers to a two-byte hexadecimal number, and places the result in another general register.

Target Device

H8/300H Tiny Series

Contents

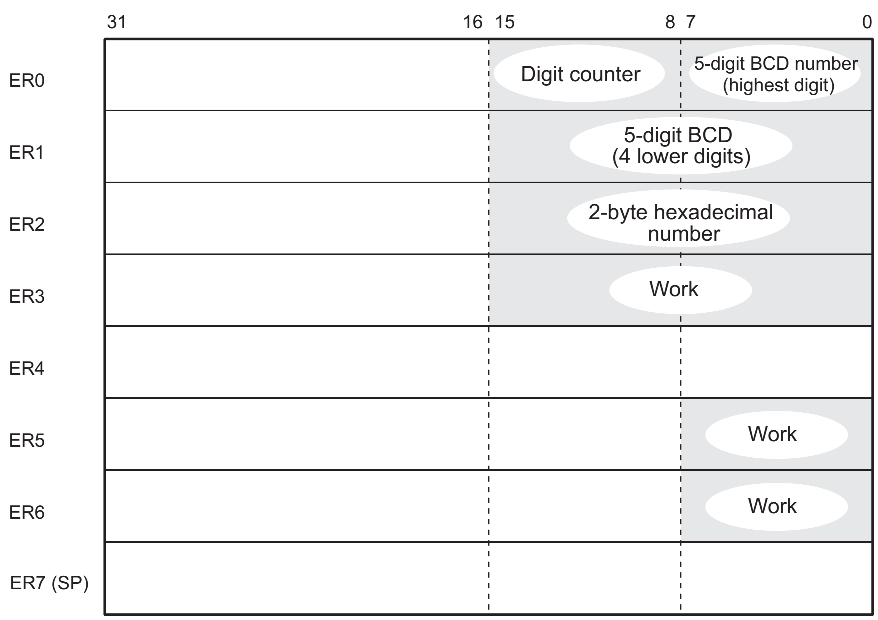
1. Function

1. Converts a three-byte five-digit BCD (binary coded decimal) number in general registers to two-byte hexadecimal, and places the result in another general register.
2. Data operations are entirely on the general registers.
3. The maximum acceptable five-digit BCD number is D'65535.

2. Arguments

Contents		Storage Location	Data Length (Bytes)
Input	5-digit BCD number (highest-order digit)	R0L	1
	5-digit BCD number (4 lower-order digits)	R1	2
Output	2-byte hexadecimal number	R2	2

3. 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

4. Programming Specifications

Program memory (bytes)	64
Data memory (bytes)	0
Stack (bytes)	2
Number of cycles	210
Re-entrant	Yes
Relocatable	Yes
Interrupts during execution	Yes

5. Description

5.1 Description of Functions

1. The arguments are as follows.
   R0L: Set the highest-order digit (byte) of the five-digit BCD number as part of the input argument.
   R1: Set the lower-order four digits (two bytes) of the BCD number as part of the input argument.
   R2: The two-byte hexadecimal number is set here as the output argument.
   Figure 1 shows the format of the input and output arguments.

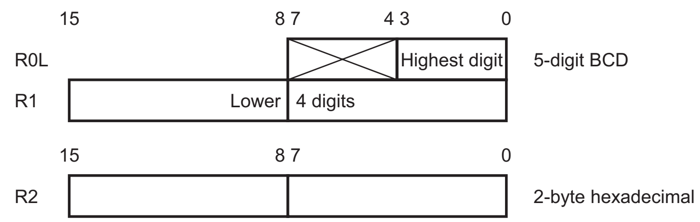


Figure 1 Input and Output Arguments

2. Figure 2 illustrates the execution of the BCD subroutine. When the input argument is set as shown below, the subroutine sets the corresponding two-byte hexadecimal number in R2.

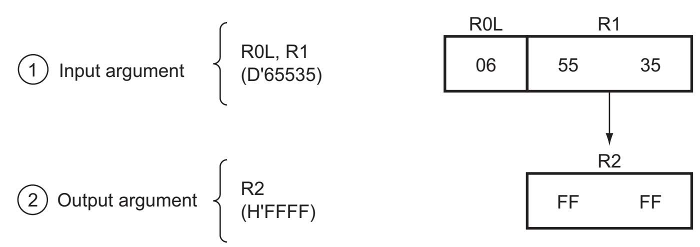


Figure 2 Example of BCD Execution

5.2 Usage Notes

1. The values of bits 4 to 7 in R0L, which hold the highest-order digit of the five-digit BCD number, are not converted and are cleared to "0", regardless of their initial value, by the execution of this subroutine.
2. D'65535 is thus the highest possible five-digit BCD number.
3. Any higher-order digits of the five-digit BCD number that are not used must be explicitly set to"0". If this is not done, the correct result might not be obtained because undefined data in the higher-order digits is included in the operation.

5.3 Description of Data Memory

No data memory is used by BCD.

5.4 Example of Usage

After setting the five-digit BCD number as an input argument, call the BCD subroutine.

```
WORK1   . RES. B 3            ......... Reservation of the data memory area for setting of the 5-digit BCD number (3 bytes) by the
                                        user program.
WORK2   . RES. B 2            ......... Reservation of the data memory area where the 2-byte hexadecimal number will be set for
                                        the user program.
            .
            .
            .
        MOV. B @WORK1, R0L    ......... Sets, as the input argument, the 5-digit BCD number specified by the user program.
        MOV. B @WORK1+1, R1H
        MOV. B @WORK1+2, R1L
        JSR    @BCD           ......... Subroutine call of BCD.
        MOV. B R2H, @WORK2    ......... Transfers the 2-byte hexadecimal number set as the output argument to the data memory area
                                        of the user program.
        MOV. B R2L, @WORK2+1
            .
            .
            .
```

5.5 Principles of Operation

1. The BCD subroutine has two sequences of processing.
   1) Extraction of the individual digits from the five-digit BCD number.
   2) Conversion of the extracted data to hexadecimal in four-bit units.

2. The processing of one digit (four bits) of input data is described below with reference to figure 3.

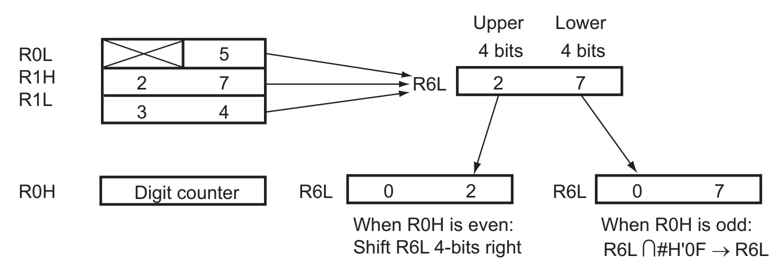


Figure 3 Dividing One Byte of Data in a General Register in Two

The BCD subroutine

1) sets H'04 in R0H to count the execution of processing for five digits;
2) transfers the current byte from the five-digit BCD number (R0L, R1H, R1L) to R6L in sequence from the highest-order byte;
3) decrements R0H.
4) selects the higher-order or lower-order four bits of that byte on the basis of whether the counter value is even or odd:
   - when R0H is odd, takes the logical AND of R6L and H'0F to extract the four lower-order bits,
   - when R0H is even, shifts R6L four bits to the right to extract the four higher-order bits; and

3. BCD-to-hexadecimal conversion is carried out in the following way.
   1) A four-digit BCD number given as $D_3D_2D_1D_0$ may be expressed as shown below.

$$D_3D_2D_1D_0 = D_3 \times 10^3 + D_2 \times 10^2 + D_1 \times 10^1 + D_0 \times 10^0 \quad \text{......Formula (1)}$$

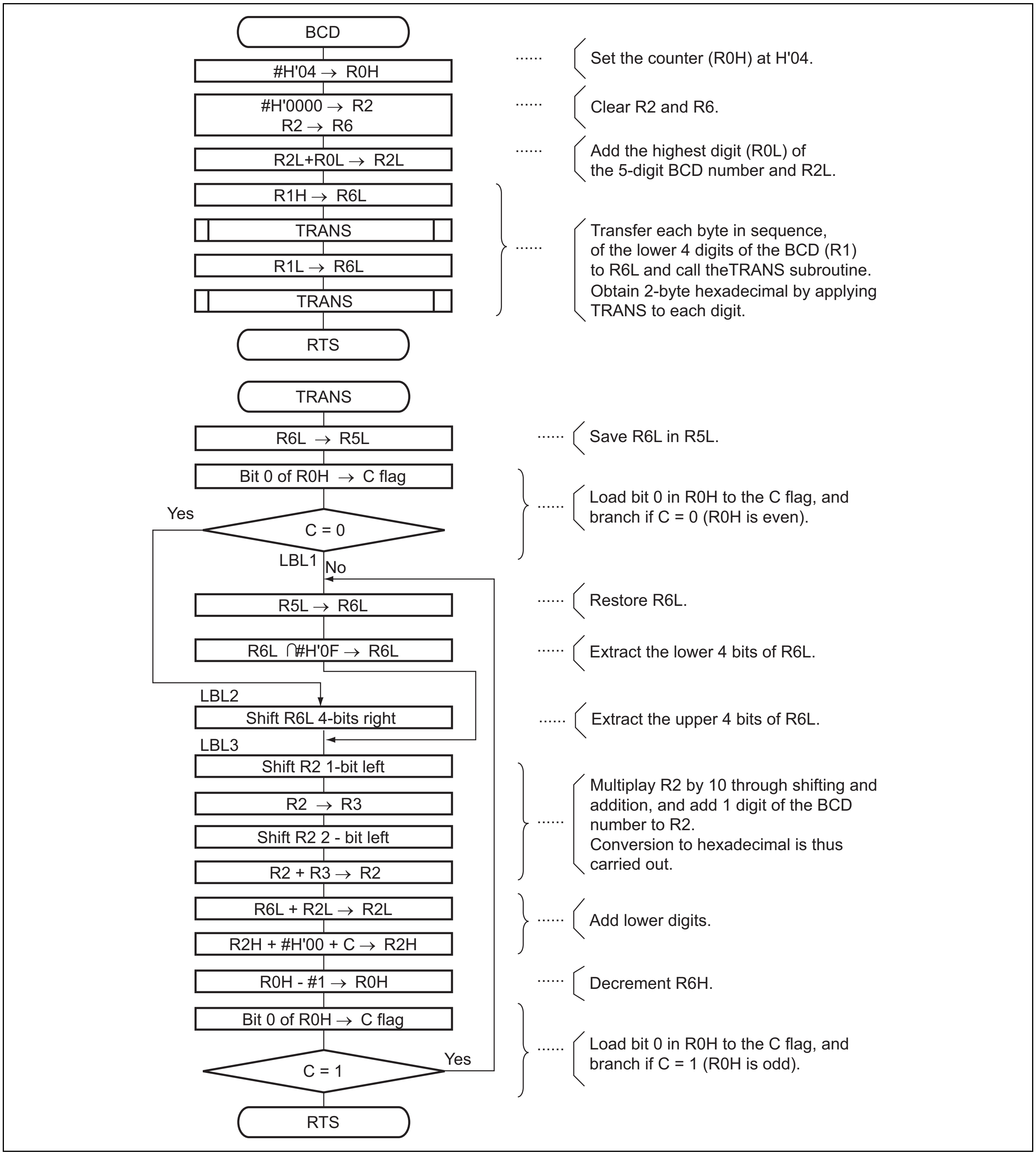
$$= \underbrace{(\underbrace{(\underbrace{D_3 \times 10 + D_2}_{\alpha}) \times 10 + D_1}_{\beta}) \times 10 + D_0}_{\gamma} \quad \text{......Formula (2)}$$

Figure 4 Concept of the Four-Digit BCD Number $D_3D_2D_1D_0$

   2) Formula (2) in the above figure tells us that a four-digit BCD number can be converted to hexadecimal by finding
      $\alpha = D_3 \times 10 + D_2$; and then calculating $\beta = \alpha \times 10 + D_1$ and $\gamma = \beta \times 10 + D_0$.
   3) $D_3 \times 10$ can be calculated by the following formulae (3) and (4).
      $D_3 \times 10 = D_3 \times (2 + 8)$ Formula (3)
      $= D_3 \times 2 \times (1 + 2^2)$ Formula (4)
   4) In the BCD subroutine, formula (4) is implemented by using R2 and R3 in the following steps.
      a. D_3 is set in R2 and then shifted 1-bit left.
      b. R2 is transferred to R3 and then shifted two-bits left.
      c. R3 is added to R2.

4. The two-byte hexadecimal number is obtained by repeating the above steps, 2 and 3, five times.

6. Flowchart


BCD
#H'04 → R0H
#H'0000 → R2
R2 → R6
R2L+R0L → R2L
R1H → R6L
TRANS
R1L → R6L
TRANS
RTS
Set the counter (R0H) at H'04.
Clear R2 and R6.
Add the highest digit (R0L) of the 5-digit BCD number and R2L.
Transfer each byte in sequence, of the lower 4 digits of the BCD (R1) to R6L and call theTRANS subroutine. Obtain 2-byte hexadecimal by applying TRANS to each digit.
TRANS
R6L → R5L
Bit 0 of R0H → C flag
Yes
C = 0
LBL1
No
R5L → R6L
R6L ∩#H'0F → R6L
LBL2
Shift R6L 4-bits right
LBL3
Shift R2 1-bit left
R2 → R3
Shift R2 2 - bit left
R2 + R3 → R2
R6L + R2L → R2L
R2H + #H'00 + C → R2H
R0H - #1 → R0H
Bit 0 of R0H → C flag
C = 1
Yes
RTS
Save R6L in R5L.
Load bit 0 in R0H to the C flag, and branch if C = 0 (R0H is even).
Restore R6L.
Extract the lower 4 bits of R6L.
Extract the upper 4 bits of R6L.
Multiplay R2 by 10 through shifting and addition, and add 1 digit of the BCD number to R2. Conversion to hexadecimal is thus carried out.
Add lower digits.
Decrement R6H.
Load bit 0 in R0H to the C flag, and branch if C = 1 (R0H is odd).

7. Program Listing

```
                         ;************************************************************
                         ;*                                                          *
                         ;*          NAME : CHANGE 5 DIGIT BCD                       *
                         ;*                 TO 2 BYTE HEXADECIMAL                    *
                         ;*                                                          *
                         ;************************************************************
                         ;*                                                          *
                         ;*         ENTRY:  R0L      (HIGHEST DIGIT (BCD))           *
                         ;*                 R1       (LOWER 4 DIGITS (BCD))          *
                         ;*                                                          *
                         ;*        RETURN: R2       (2 BYTE HEXADECIMAL)             *
                         ;*                                                          *
                         ;************************************************************
                         ;
                                 .CPU        300HN
0000                             .SECTION    BCD_code,CODE,ALIGN=2
                                 .EXPORT     BCD
                         ;
     00000000            BCD     .EQU        $           ;Entry point
0000 F004                        MOV.B       #H'04,R0H   ;Set bit counter
0002 79020000                    MOV.W       #H'0000,R2  ;Clear R2
0006 0D26                        MOV.W       R2,R6       ;Clear R6
                         ;
0008 088A                        ADD.B       R0L,R2L     ;R2L + R0L -> R2L
000A 0C1E                        MOV.B       R1H,R6L     ;R1H -> R6L
000C 5506                        BSR         TRANS
000E 0C9E                        MOV.B       R1L,R6L     ;R1L -> R6L
0010 5502                        BSR         TRANS
0012 5470                        RTS
                         ;
                         ;-----------------------------------------------------------
                         ;
0014                     TRANS
0014 0CED                        MOV.B       R6L,R5L     ;R6L -> R5L
0016 7700                        BLD         #0,R0H      ;Load bit 0 of R0H
0018 4406                        BCC         LBL2        ;Branch if C=0
001A                     LBL1
001A 0CDE                        MOV.B       R5L,R6L     ;R51 -> R6L
001C EE0F                        AND.B       #H'0F,R6L   ;Clear bit 7-4 of R6L
001E 4008                        BRA         LBL3        ;Branch always
0020                     LBL2
0020 110E                        SHLR.B      R6L         ;Shift R6L 4 bits left
0022 110E                        SHLR.B      R6L
0024 110E                        SHLR.B      R6L
0026 110E                        SHLR.B      R6L
0028                     LBL3
0028 100A                        SHLL.B      R2L         ;Shift hexadecimal 1 bit left
002A 1202                        ROTXL.B     R2H
002C 0D23                        MOV.W       R2,R3       ;R2 -> R3
002E 100A                        SHLL.B      R2L         ;Shift hexadecimal 2 bit left
0030 1202                        ROTXL.B     R2H
0032 100A                        SHLL.B      R2L
```

```
 53   0034 1202                 53              ROTXL.B     R2H
 54   0036 0932                 54              ADD.W       R3,R2       ;R3 + R2 -> R2
 55   0038 08EA                 55              ADD.B       R6L,R2L
 56   003A 9200                 56              ADDX.B      #0,R2H
 57   003C 1A00                 57              DEC.B       R0H         ;Decrement bit counter
 58   003E 7700                 58              BLD         #0,R0H      ;Load bit 0 of R0H
 59   0040 45D8                 59              BCS         LBL1        ;Branch if C=1
 60   0042 5470                 60              RTS
 61                             61      ;
 62                             62              .END
*****TOTAL ERRORS       0
*****TOTAL WARNINGS     0
```

Revision Record

Rev.	Date	Description	
		Page	Summary
2.00	Feb.28.06	—	Format has been changed from Hitachi version to Renesas version.