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.

H8/300L Super Low Power Series

Addition of Multiple-Precision BCD Numbers (ADDD2)

Introduction

The software ADDD2 adds a multiple-precision binary-coded decimal (BCD) number to another multiple-precision BCD number and places the result in the data memory where the augend was placed.

Target Device

H8/38024

Contents

1. Arguments

Description		Memory area	Data length (bytes)
Input	Augend and addend byte count	R0L	1
	Start address of augend	R3	2
	Start address of addend	R4	2
Output	Start address of the result of addition	R3	2
	Error	Z flag (CCR)	—
	Carry	C flag (CCR)	—

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)
44
Data memory (bytes)
0
Stack (bytes)
0
Clock cycle count
7680
Reentrant
Possible
Relocation
Possible
Interrupt
Possible

4. Notes

The clock cycle count (7680) in the specifications is for addition of 255 bytes to 255 bytes.

5. Description

5.1 Details of functions

1. The following arguments are used with the software ADDD2:
   - R0L: Sets, as an input argument, the byte count of an augend and an addend in 2-digit hexadecimals.
   - R3: Sets the start address of the augend in the data memory area. The start address of the result of addition is placed in this register after execution of the software ADDD2.
   - R4: Sets, as an input argument, the start address of the addend in the data memory area.
   - Z flag (CCR): Indicates an error in data length as an output argument.
     - Z flag = 0: The data byte count (R0L) was not 0.
     - Z flag = 1: The data byte count (R0L) was 0 (indicating an error).
   - C flag (CCR): Indicates whether there is or isn't a carry, as an output argument, after execution of the software ADDD2.
     - C flag = 0: There is no carry in the result of addition.
     - C flag = 1: There is a carry in the result of addition (see figure 2).

2. The following figure illustrates the execution of the software ADDD2. When the input arguments are set as shown in (1), the result of addition is placed in the data memory area as shown in (2).

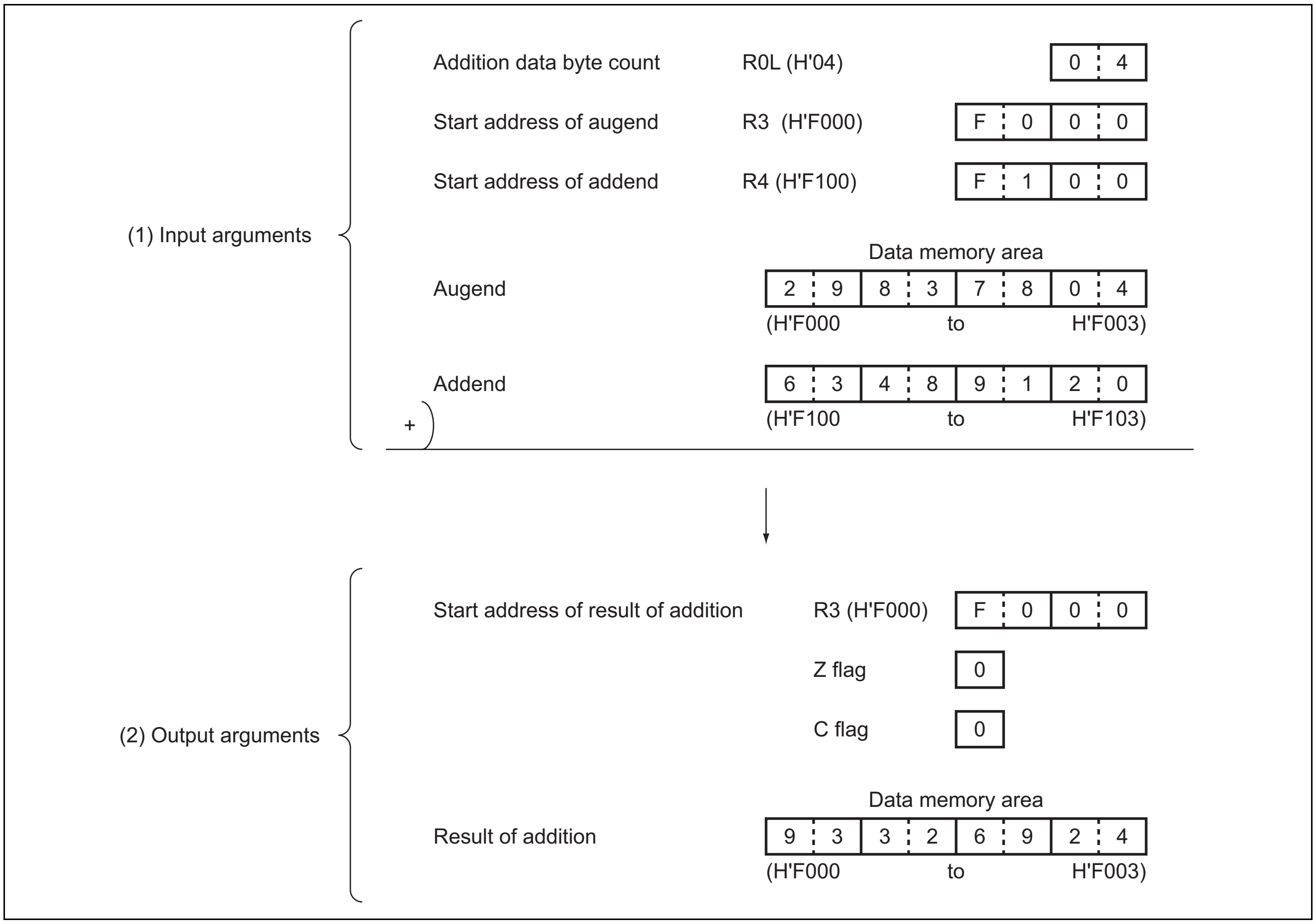


Figure 1 Example of Software ADDD2 Execution

Figure 2 shows an example of addition where a carry has been produced.

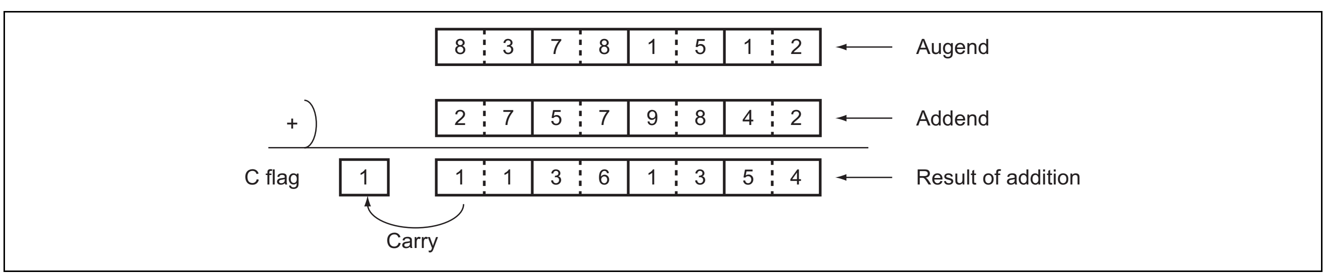


Figure 2 Example of Addition with a Carry

5.2 Notes on usage

1. When the upper bits are not used (see figure3), set them to 0. The software ADDD2 performs byte-based addition; when 0 are not set in the unused upper bits, a correct result cannot be obtained because the addition is done on the numbers including indeterminate data.

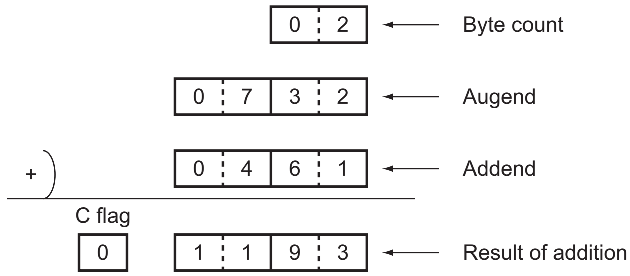


Figure 3 Example of Addition with Upper Bits Unused

2. After execution of the software ADDD2, the augend will be lost because the result is placed in the data memory area where the augend was set. When the augend is still needed after software ADDD2 execution, save it in memory.

5.3 Data memory

The software ADDD2 uses no data memory.

5.4 Example of usage

This is an example of adding 8 bytes of data. Set the start addresses of a byte count, an augend, and an addend in the registers and call the software ADDD2 as a subroutine.

```
WORK1    .RES.B    1              ---------- Reserve a data memory area in which the user program places
                                             a byte count.

WORK2    .RES.B    8              ---------- Reserve a data memory area in which the user program places
                                             an 8-byte (16-digit BCD) augend.

WORK3    .RES.B    8              ---------- Reserve a data mermoy area in which the user program places
                                             an 8-byte (16-digit BCD) addend.
           .
           .
           .
         MOV.B     @WORK1, R0L    ------- Place in the input argument (R0L) the byte count
                                          set by the user program.

         MOV.W     #WORK2, R3     ------- Place in the input argument (R3) the start address
                                          of the augend set by the user program.

         MOV.W     #WORK3, R4     ------- Place in the input argument (R4) the start address
                                          of the addend set by the user program.

         JSR       @ADDD2         ------- Call the software ADDD2 as a subroutine.

         BCS       OVER           ------- Branch to the carry processing routine
                                          when a carry has occurred in the result of addition.
           .
           .
           .
OVER     Carry processing routine
```

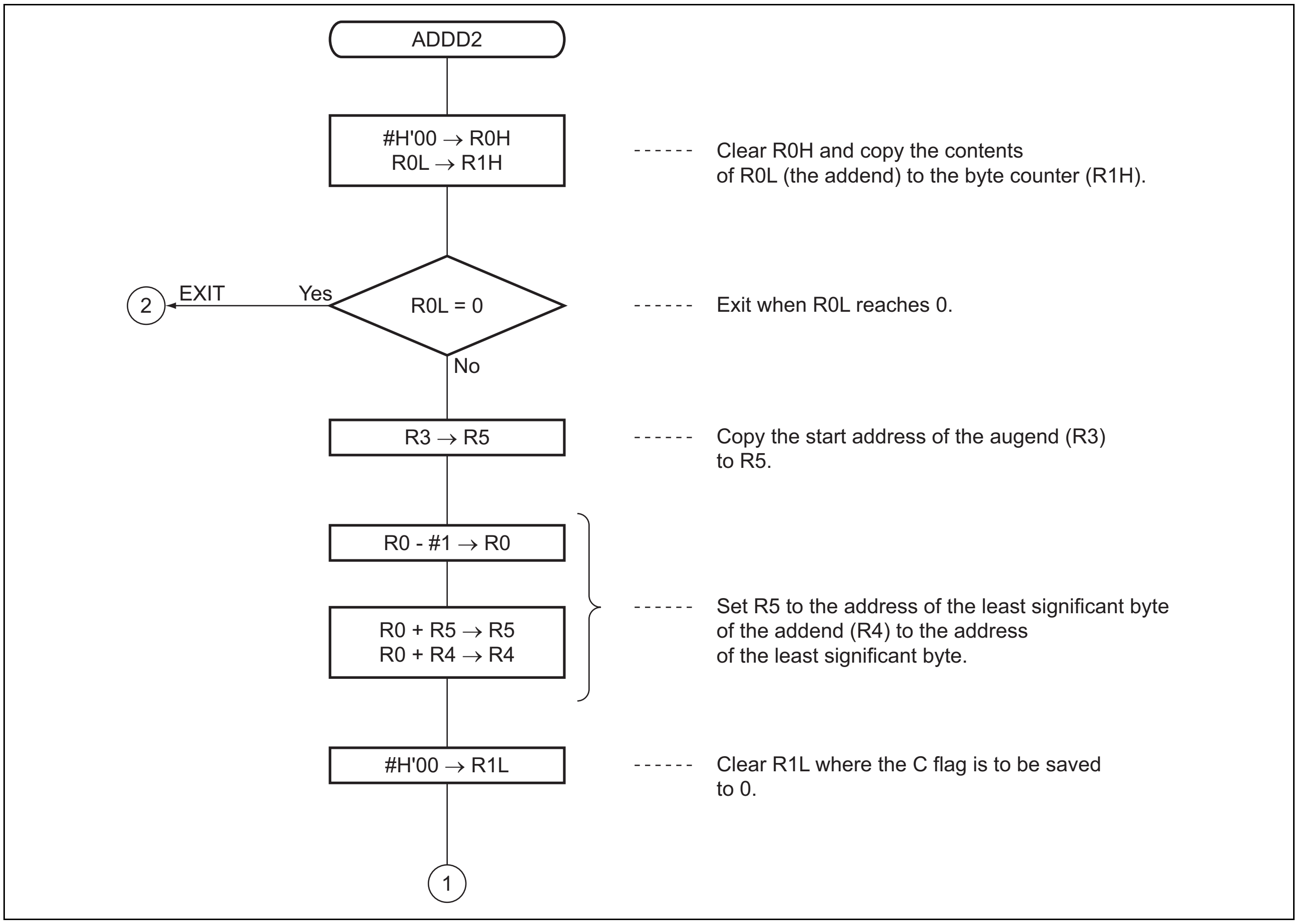
5.5 Operation

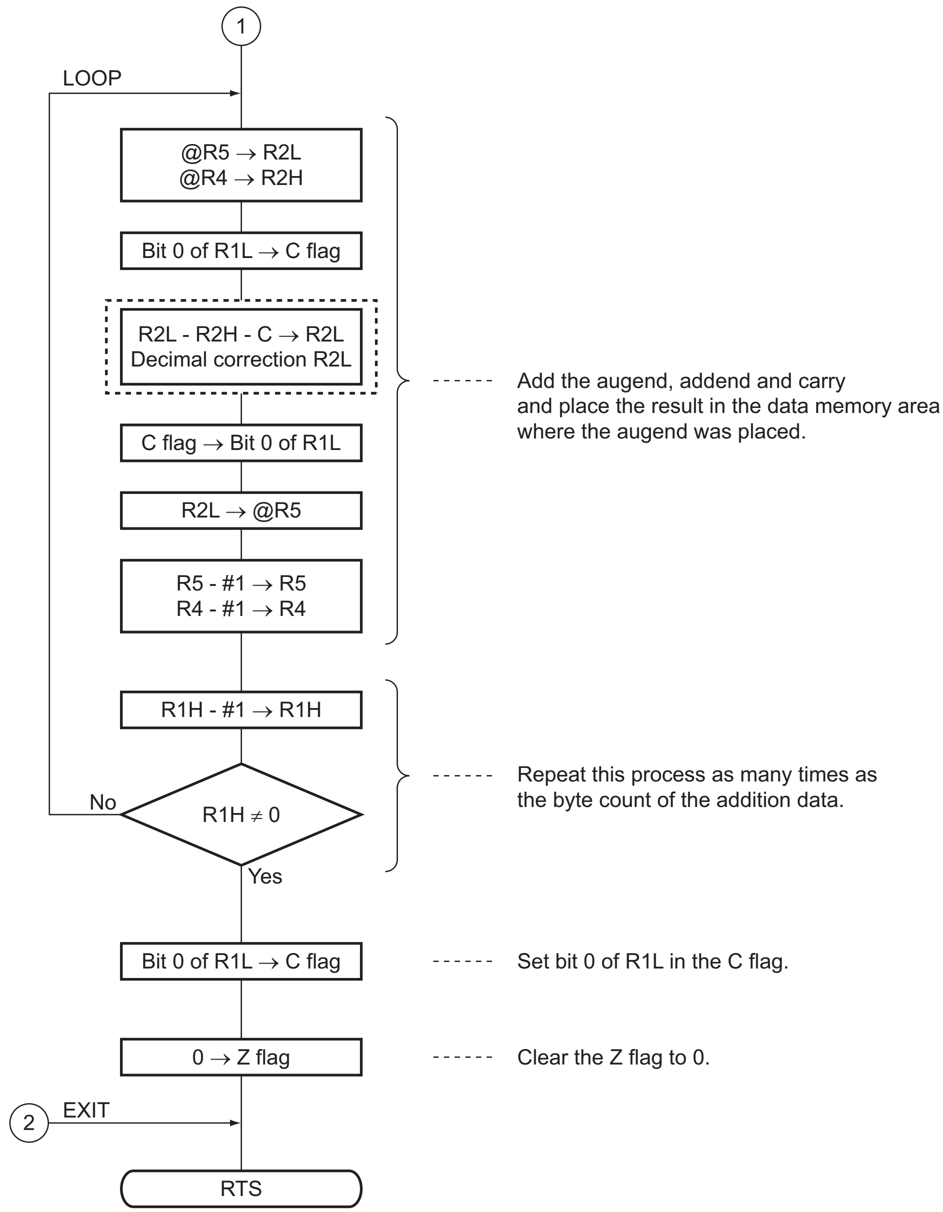
1. Since the augend and addend data are placed in registers, 2 digits in 1 byte, addition of multiple-precision BCD numbers can be done by performing a series of 1-byte add instructions (ADDX.B) with decimal-correct instructions (DAA).
2. The address of the lowest byte of the data memory area for the augend is placed in R3, and the address of the lowest byte of the data memory area for the addend in R4.
3. R1L that is used for saving the C flag is cleared.
4. The augend and addend are loaded to R2L and R2H respectively, byte by byte, starting at their lowest byte and then the operation given by equation 1 is executed:

   R2L (augend) + R2H (addend) + C → R2L
   Decimal correction of R2L → R2L ------- equation 1
   R2L → @R3

   where the C flag indicates a carry that may be produced in the result of addition of the lower bytes.
5. The result of step (4) is placed in the data memory area for the augend.
6. R3, R4, and R0L are decremented each time the operation of steps 4 and 5 has been finished. This processing is repeated until R0L reaches 0.

6. Flowchart


ADDD2
#H'00 → R0H
R0L → R1H
Clear R0H and copy the contents of R0L (the addend) to the byte counter (R1H).
EXIT
Yes
R0L = 0
Exit when R0L reaches 0.
No
2
R3 → R5
Copy the start address of the augend (R3) to R5.
R0 - #1 → R0
R0 + R5 → R5
R0 + R4 → R4
Set R5 to the address of the least significant byte of the addend (R4) to the address of the least significant byte.
#H'00 → R1L
Clear R1L where the C flag is to be saved to 0.
1



Note: ADDD2 is the same as ADD2, SUB2 and SUBD2 except for the stop surrounded by dotted lines.

7. Program List

```
*** H8/300 ASSEMBLER VER 1.0B ** 08/18/92 10:02:42
PROGRAM NAME =
                               ;*****************************************************************
                               ;*
                               ;*      00 - NAME  :MULTIPLE-PRECISION DECIMAL ADDITION
                               ;*                  (ADDD2)
                               ;*
                               ;*****************************************************************
                               ;*
                               ;*      ENTRY      :R0L (BYTE COUNTER OF ADDTION DATA)
                               ;*                  R3 (START ADDRESS OF AUGEND)
                               ;*                  R4 (START ADDRESS OF ADDEND)
                               ;*
                               ;*      RETURNS    :R3 (START ADDRESS OF RESULT)
                               ;*                  Z flag OF CCR (Z=0;TRUE,Z=1;FALSE)
                               ;*                  C flag OF CCR (C = 0;TRUE,C = 1;OVERFLOW)
                               ;*
                               ;*****************************************************************
                               ;
ADDD2_co C   0000                      .SECTION            ADDD2_code,CODE,ALIGN=2
                     .EXPORT   ADDD2
                               ;
ADDD2_co C         00000000    ADDD2   .EQU $              ;Entry point
ADDD2_co C   0000  F000                MOV.B    #H'00,R0H  ;Clear R0H
ADDD2_co C   0002  0C81                MOV.B    R0L,R1H    ;Clear R1H
ADDD2_co C   0004  4724                BEQ      EXIT       ;Branch if Z=1 then exit
ADDD2_co C   0006  0D35                MOV.W    R3,R5
ADDD2_co C   0008              MAIN
ADDD2_co C   0008  1B00                SUBS.W   #1,R0      ;Decrement R0
ADDD2_co C   000A  0905                ADD.W    R0,R5      ;Set end address to summand pointer
ADDD2_co C   000C  0904                ADD.W    R0,R4      ;Set end address to addend pointer
ADDD2_co C   000E  F900                MOV.B    #H'00,R1L  ;Clear R1L
ADDD2_co C   0010              LOOP
ADDD2_co C   0010  685A                MOV.B    @R5,R2L    ;Load summand data
ADDD2_co C   0012  6842                MOV.B    @R4,R2H    ;Load addend data
ADDD2_co C   0014  7709                BLD      #0,R1L     ;Bit load bit 0 of R1L
ADDD2_co C   0016  0E2A                ADDX.B   R2H,R2L    ;R2H + R2L + C -> R2L
ADDD2_co C   0018  0F0A                DAA      R2L        ;Decimal adjust R1L
ADDD2_co C   001A  6709                BST      #0,R1L     ;Store C falg to bit 0 of R1L
ADDD2_co C   001C  68DA                MOV.B    R2L,@R5    ;Store struct
ADDD2_co C   001E  1B05                SUBS.W   #1,R5      ;Decrement summand pointer
ADDD2_co C   0020  1B04                SUBS.W   #1,R4      ;Decrement addend pointer
ADDD2_co C   0022  1A01                DEC.B    R1H        ;Decrement R1H
ADDD2_co C   0024  46EA                BNE      LOOP       ;Branch if Z=0
                               ;
ADDD2_co C   0026  7709                BLD      #0,R1L     ;Load bit 0 of R1L to C flag
ADDD2_co C   0028  06FB                ANDC.B   #H'FB,CCR  ;Clear Z flag of CCR
ADDD2_co C   002A              EXIT
ADDD2_co C   002A  5470                RTS
                               ;
                                       .END
*****TOTAL ERRORS 0
*****TOTAL WARNINGS 0
```

Website and Support

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

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

Revision Record

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