SH7000 Series

64 Bit + 64 Bit = 64 Bit (Signed)

Label: ADDS64

Functions Used: ADDV Instruction

Contents

1. Function

Adds the augend (signed 64 bits) and addend (signed 64 bits), and determines the sum (signed 64 bits). At this time, whether or not an overflow or underflow is present is set in the T bit. Overflow and underflow discrimination is not performed.

2. Arguments

Description		Storage Location	Data Length (Bytes)
Input	Upper 32 bits of augend (signed 64 bits)	R0	4
	Lower 32 bits of augend (signed 64 bits)	R1	4
	Upper 32 bits of addend (signed 64 bits)	R2	4
	Lower 32 bits of addend (signed 64 bits)	R3	4
Output	Upper 32 bits of sum (signed 64 bits)	R0	4
	Lower 32 bits of sum (signed 64 bits)	R1	4
	With/without overflow or underflow (with: T = 1, without: T = 0)	T bit (SR)	4

3. Internal Register Changes and Flag Changes

	(Before Execution) → (After Execution)
R0	Upper 32 bits of augend → Upper 32 bits of sum
R1	Lower 32 bits of augend → Lower 32 bits of sum
R2	Upper 32 bits of addend → No change
R3	Lower 32 bits of addend → No change
R4	Work
R5	Work
R6	
R7	
R8	
R9	
R10	
R11	
R12	
R13	
R14	
R15	(SP)

T bit | * |

— : No change
* : Change
0 : Fixed 0
1 : Fixed 1

4. Programming Specifications

Program memory (bytes)
28
Data memory (bytes)
0
Stack (bytes)
8
Number of states
15
Reentrant
Yes
Relocation
Yes
Intermediate interrupt
Yes

5. Description

(1) Function

Details of the arguments are as follows.

R0: Set the upper 32 bits of the augend (signed 64 bits) as the input argument.
Holds the upper 32 bits of the sum (signed 64 bits) as the output argument.

R1: Sets the lower 32 bits of the augend (signed 64 bits) as the input argument.
Holds the lower 32 bits of the sum (signed 64 bits) as the output argument.

R2: Set the upper 32 bits of the addend (signed 64 bits) as the input argument.

R3: Set the lower 32 bits of the addend (signed 64 bits) as the input argument.

T bit (SR): Indicates the presence or absence of an overflow or underflow after execution of the software instruction ADDS64.
T bit = 1: Indicates an overflow or underflow was generated.
T bit = 0: Indicates no overflow or underflow was generated.

Figure 1 shows a software ADDS64 execution example.

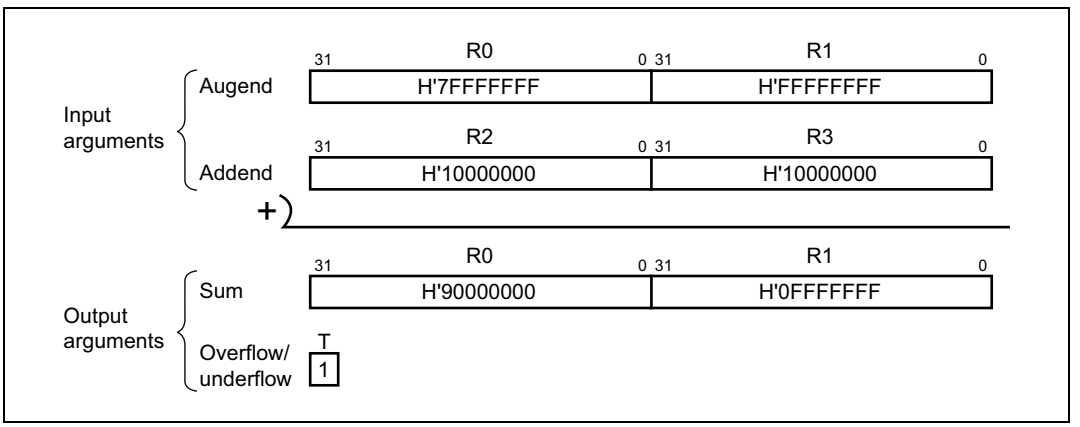


Figure 1 Software ADDS64 Execution Example

(2) Usage Notes

Since the sum is set in R1 and R2, which contained the augend settings, the augend data is destroyed. If the value for the augend will be needed after the software ADDS64 instruction is executed, it should be saved beforehand.

(3) RAM Used

No RAM is used by the software ADDS64 instruction.

(4) Usage Example

After the augend and addend are set in input arguments, the software ADDS64 instruction is executed by a subroutine call.

```
          MOV.L   DATA1,R0      ... Sets augend (upper 32 bits) in input argument
          MOV.L   DATA2,R1      ... Sets augend (lower 32 bits) in input argument
          MOV.L   DATA3,R2      ... Sets addend (upper 32 bits) in input argument
          BSR     ADDS64        ... Subroutine call to ADDS64
          MOV.L   DATA4,R3      ... Sets addend (lower 32 bits) in input argument
          BT      ERROR         ... Branches to error-processing subroutine if overflow or underflow occurs
                .
                .
                .
                .
        .align   4
DATA1   .data.l  H'7FFFFFFF
DATA2   .data.l  H'FFFFFFFF
DATA3   .data.1  H'10000000
DATA4   .data.1  H'10000000
```

(5) Operating Principle

(a) The lower 32 bits of the augend and addend are added using the add with carry instruction (ADDC). If a carry occurs after addition, it is indicated in the T bit (figure 2-(1)).

(b) If a carry occurs, it is added to the upper 32 bits of the augend. The content of the T bit from (a) is stored in R4 and then added to the upper 32 bits of the augend using the binary addition with overflow check instruction (ADDV). If a carry occurs, 1 is added to the upper 32 bits of the augend. If there is no carry, 0 is added to the upper 32 bits of the augend. Whether an overflow or underflow has occurred is indicated in the T bit after addition (figure 2-(2)).

(c) The sum from (b) is added to the upper 32 bits of the addend by binary addition using the overflow check instruction (ADDV). Whether an overflow or underflow has occurred is indicated in the T bit after addition (figure 2-(3)).

(d) The contents of the T bits from (b) and (c) are logically ORed and the result is stored in the T bit. If the value of the T bit is 1, an overflow or underflow has occurred. If the value of the T bit is 0, no overflow or underflow has occurred.

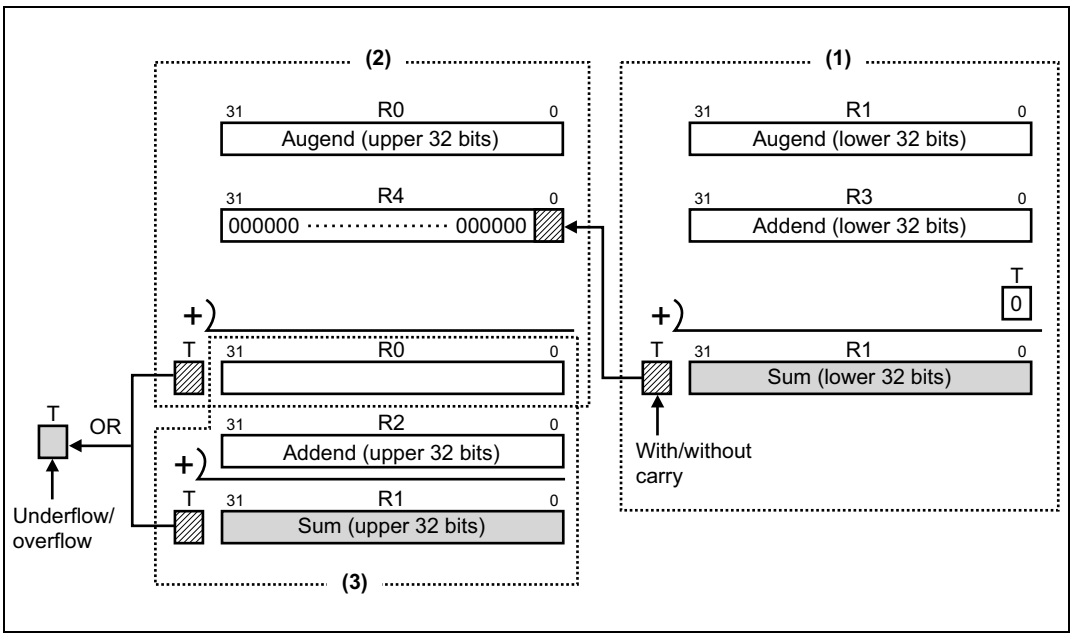
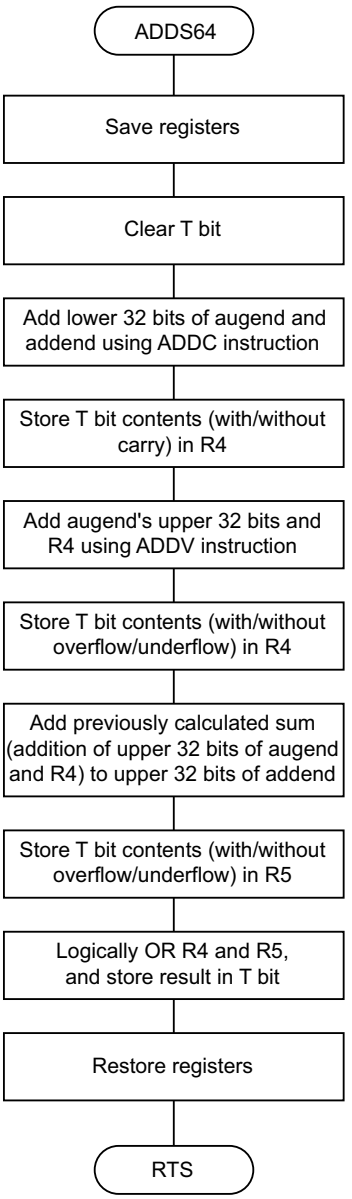


Figure 2 Signed Addition

6. Flowchart


ADDS64
Save registers
Clear T bit
Add lower 32 bits of augend and addend using ADDC instruction
Store T bit contents (with/without carry) in R4
Add augend's upper 32 bits and R4 using ADDV instruction
Store T bit contents (with/without overflow/underflow) in R4
Add previously calculated sum (addition of upper 32 bits of augend and R4) to upper 32 bits of addend
Store T bit contents (with/without overflow/underflow) in R5
Logically OR R4 and R5, and store result in T bit
Restore registers
RTS

7. Program Listing

```
  1                       1   ;*************************************************************
  2                       2   ;*                                                           *
  3                       3   ;*        NAME ; 64 BIT SIGNED BINARY ADDITION (ADDS64)      *
  4                       4   ;*                                                           *
  5                       5   ;*************************************************************
  6                       6   ;*                                                           *
  7                       7   ;*      ENTRY : R0 (UPPER 32 BIT AUGEND)                     *
  8                       8   ;*              R1 (LOWER 32 BIT AUGEND)                     *
  9                       9   ;*              R2 (UPPER 32 BIT ADDEND)                     *
 10                      10   ;*              R3 (LOWER 32 BIT ADDEND)                     *
 11                      11   ;*    RETURNS : R0 (UPPER 32 BIT SUM)                        *
 12                      12   ;*              R1 (LOWER 32 BIT SUM)                        *
 13                      13   ;*           T BIT (OVERFLOW/UNDERFLOW -> TRUE;T=1,FALSE;T=0) *
 14                      14   ;*                                                           *
 15                      15   ;*************************************************************
 16 00001000             16           .SECTION A,CODE,LOCATE=H'1000
 17          00001000    17   ADDS64  .EQU    $               ; Entry point
 18 00001000 2F46        18           MOV.L   R4,@-R15        ; Escape register
 19 00001002 2F56        19           MOV.L   R5,@-R15        ;
 20 00001004 0008        20           CLRT                    ; Clear T bit
 21 00001006 313E        21           ADDC    R3,R1           ; Lower 32 bit augend + Lower 32
                                                              ; bit addend
 22 00001008 0429        22           MOVT    R4              ; R4 <- Carry
 23 0000100A 304F        23           ADDV    R4,R0           ; Upper 32 bit augend + Carry
 24 0000100C 0429        24           MOVT    R4              ; R4 <- Overflow / Underflow
 25 0000100E 302F        25           ADDV    R2,R0           ; Upper 32 bit augend + Upper 32
                                                              ; bit addend
 26 00001010 0529        26           MOVT    R5              ; R5 <- Overflow / Underflow
 27 00001012 245B        27           OR      R5,R4           ; R4 <- R5 or R4
 28 00001014 4401        28           SHLR    R4              ; T bit <- Overflow / Underflow
 29 00001016 65F6        29           MOV.L   @R15+,R5        ; Return register
 30 00001018 000B        30           RTS                     ;
 31 0000101A 64F6        31           MOV.L   @R15+,R4        ;
 32                      32           .END
*****TOTAL ERRORS    0
*****TOTAL WARNINGS  0
```

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.