SH7000 Series

32 Bit × 32 Bit = 64 Bit (Signed)

Label: MULS32

Functions Used: MULU Instruction
SWAP Instruction
NEGC Instruction

Contents

1. Function 2
2. Arguments 2
3. Internal Register Changes and Flag Changes 3
4. Programming Specifications 4
5. Notes 4
6. Description 5
7. Flowchart 7
8. Program Listing 9

1. Function

Multiplies the multiplicand (signed 32 bits) by the multiplier (signed 32 bits) and determines the product (signed 64 bits).

2. Arguments

Description		Storage Location	Data Length (Bytes)
Input	Multiplicand (signed 32 bits)	R0	4
	Multiplier (signed 32 bits)	R1	4
Output	Upper 32 bits of product (signed 64 bits)	R2	4
	Lower 32 bits of product (signed 64 bits)	R3	4

3. Internal Register Changes and Flag Changes

	(Before Execution) → (After Execution)
R0	Multiplicand (unsigned 32 bits) → No change
R1	Multiplier (unsigned 32 bits) → Change
R2	Undefined → Product (upper 32 bits)
R3	Undefined → Product (lower 32 bits)
R4	Work
R5	Work
R6	Work
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)
92
Data memory (bytes)
0
Stack (bytes)
16
Number of states
48
Reentrant
Yes
Relocation
Yes
Intermediate interrupt
Yes

5. Notes

The number of states indicated in the programming specifications is the value when H'7FFFFFFF × H'80000000 is calculated.

6. Description

(1) Function

Details of the arguments are as follows.

R0: Set the multiplicand (signed 32 bits) as the input argument.
R1: Set the multiplier (signed 32 bits) as the input argument.
R2: Holds the upper 32 bits of the product (signed 64 bits) as the output argument.
R3: Holds the lower 32 bits of the product (signed 64 bits) as the output argument.

Figure 1 shows a software MULS32 execution example.

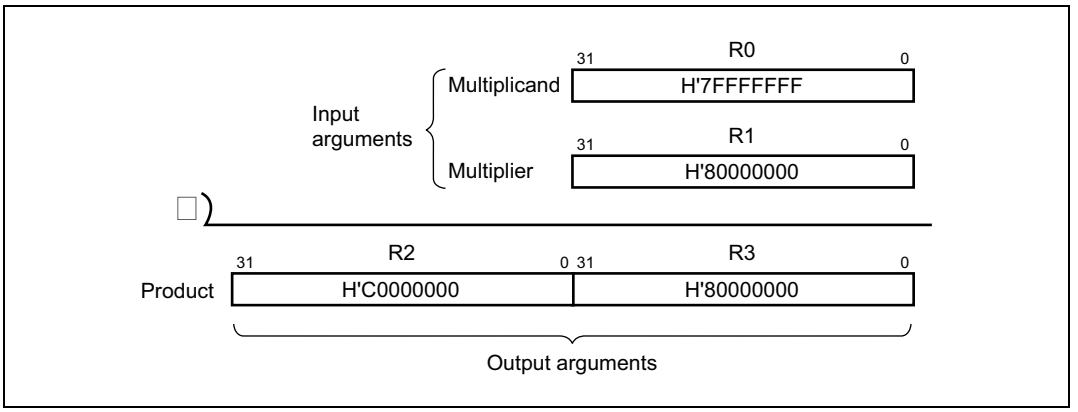


Figure 1 Software MULS32 Execution Example

(2) Usage Notes

The contents of R1, which sets the multiplier, are changed by execution of the software MULS32 instruction. If the value for the multiplier will be needed after the software MULS32 instruction is executed, it should be saved beforehand.

(3) RAM Used

No RAM is used by the software MULS32 instruction.

(4) Usage Example

After the multiplicand and multiplier are set, the software instruction MULS32 is executed by a subroutine call.

```
          MOV.L   DATA1,R0      .... Sets multiplicand in input argument (R0)
          BSR     MULU32        .... Subroutine call to software instruction MULS32
          MOV.L   DATA2,R1      .... Sets multiplier in input argument (R1)
                 .
                 .
                 .
                 .
        .align    4
DATA1   .data.l   H'7FFFFFFF
DATA2   .data.l   H'80000000
```

(5) Operating Principle

(a) As shown in figure 2, multiplication is performed in 16 bit units. Partial products (1–4) are determined, and these are added to get the final 64-bit product. The 16-bit unsigned multiplication instruction (MULU) is used in multiplication of partial products, so if the multiplicand or multiplier are negative, they are converted to positive before multiplication.

(b) The product is calculated as positive, so the determination of whether it is positive or negative is made using exclusive OR on the multiplicand and multiplier MSB, as shown in table 1.

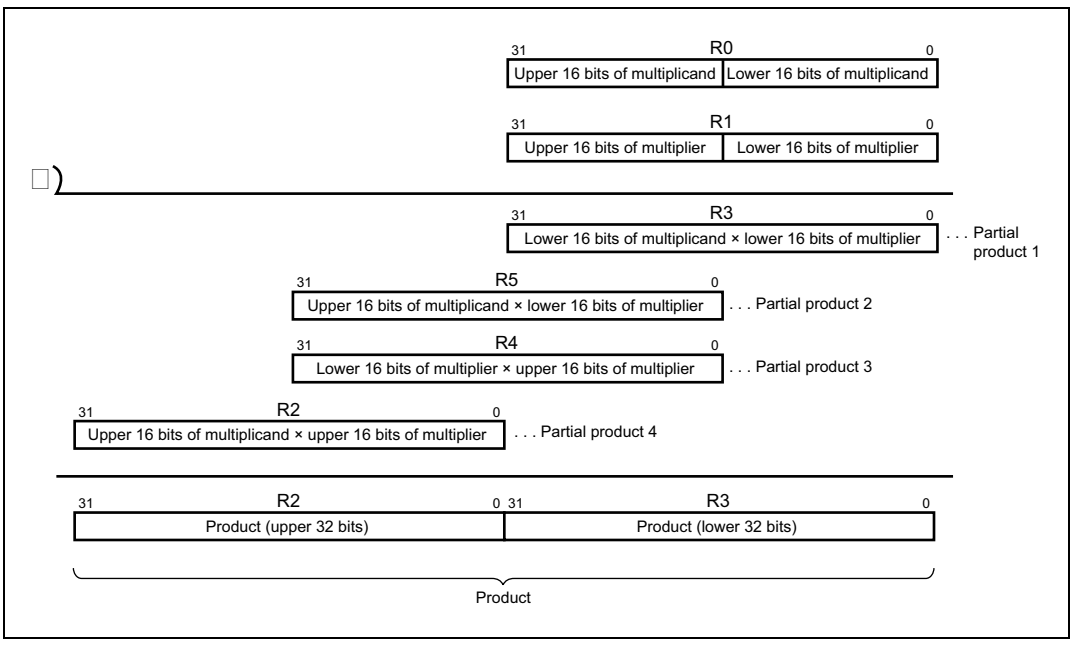
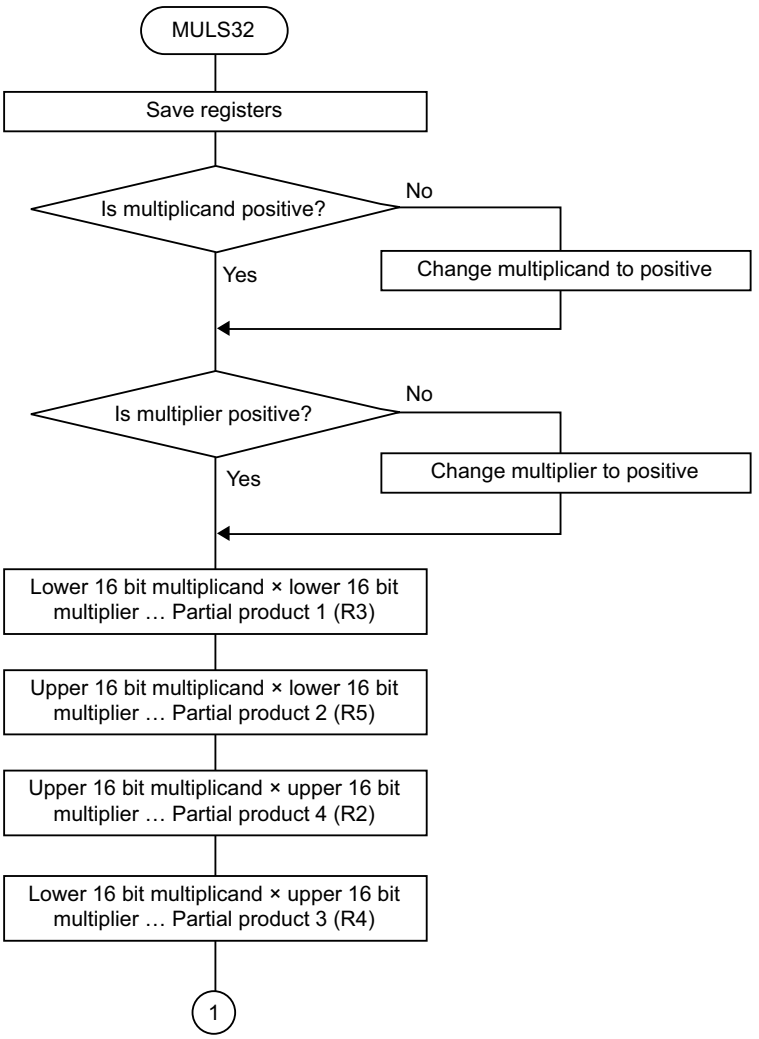


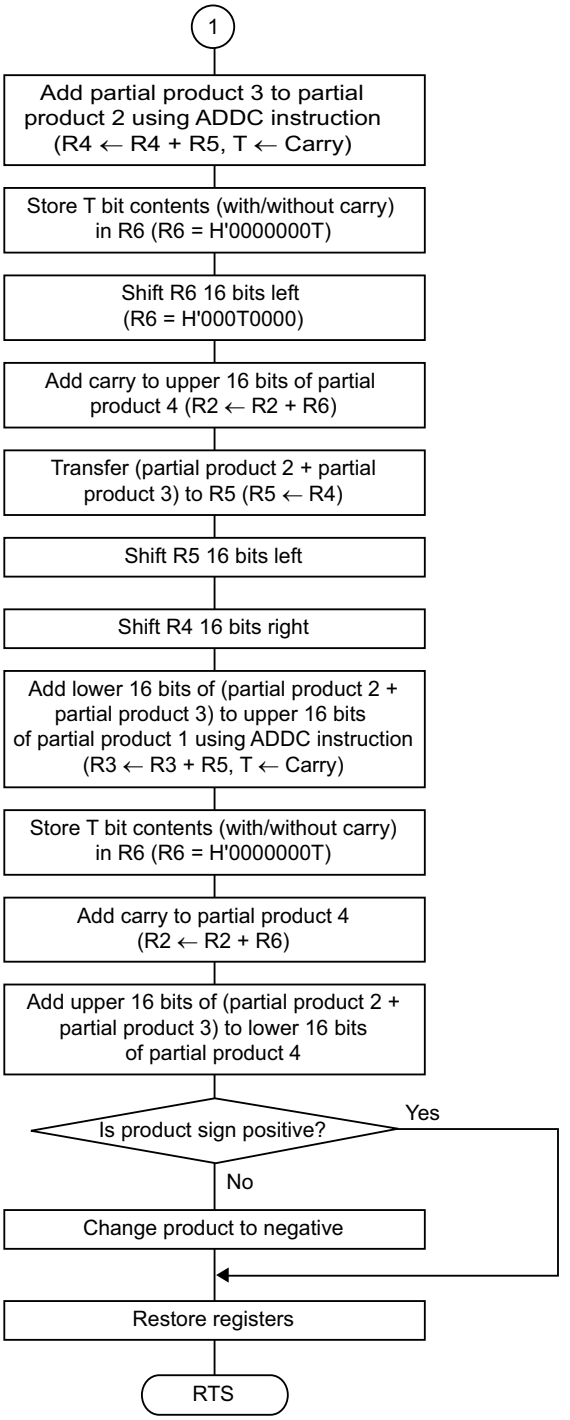
Figure 2 Multiplication

Table 1 Product Sign Changes

MSB of Multiplicand	MSB of Multiplier	Product Sign Change
Positive	Positive	Positive
Positive	Negative	Negative
Negative	Positive	Negative
Negative	Negative	Positive

7. Flowchart




1
Add partial product 3 to partial product 2 using ADDC instruction (R4 ← R4 + R5, T ← Carry)
Store T bit contents (with/without carry) in R6 (R6 = H'0000000T)
Shift R6 16 bits left (R6 = H'000T0000)
Add carry to upper 16 bits of partial product 4 (R2 ← R2 + R6)
Transfer (partial product 2 + partial product 3) to R5 (R5 ← R4)
Shift R5 16 bits left
Shift R4 16 bits right
Add lower 16 bits of (partial product 2 + partial product 3) to upper 16 bits of partial product 1 using ADDC instruction (R3 ← R3 + R5, T ← Carry)
Store T bit contents (with/without carry) in R6 (R6 = H'0000000T)
Add carry to partial product 4 (R2 ← R2 + R6)
Add upper 16 bits of (partial product 2 + partial product 3) to lower 16 bits of partial product 4
Is product sign positive?
Yes
No
Change product to negative
Restore registers
RTS

8. Program Listing

```
                                ;*************************************************************
                                ;*                                                           *
                                ;*        NAME : 32 BIT SINGNED MULTIPLATION (MULS32)        *
                                ;*                                                           *
                                ;*************************************************************
                                ;*                                                           *
                                ;*      ENTRY : R0 (MULTIPLICAND)                            *
                                ;*              R1 (MULTIPLIER)                              *
                                ;*    RETURNS : R2 (UPPER 32 BIT PRODUCT)                    *
                                ;*              R3 (LOWER 32 BIT PRODUCT)                    *
                                ;*                                                           *
                                ;*************************************************************
00001000                                .SECTION A,CODE,LOCATE=H'1000
          00001000              MULS32  .EQU    $               ; Entry point
00001000 4F12                           STS.L   MACL,@-R15      ; Escape register
00001002 2F46                           MOV.L   R4,@-R15        ;
00001004 2F56                           MOV.L   R5,@-R15        ;
00001006 2F66                           MOV.L   R6,@-R15        ;
                                                                ;
00001008 4011                           CMP/PZ  R0              ; Multiplicand >= 0 ?
0000100A 8900                           BT      MULS321         ; Yes
0000100C 600B                           NEG     R0,R0           ; Change plus
0000100E                        MULS321                         ;
0000100E 4111                           CMP/PZ  R1              ; Multiplier >= 0 ?
00001010 8900                           BT      MULS322         ; Yes
00001012 611B                           NEG     R1,R1           ; Change plus
00001014                        MULS322                         ;
00001014 201E                           MULU    R1,R0           ; Lower 16 bit + lower 16 bit -> R3
00001016 6009                           SWAP.W  R0,R0           ;
00001018 031A                           STS     MACL,R3         ;
0000101A 201E                           MULU    R1,R0           ; Upper 16 bit + lower 16 bit -> R5
0000101C 6119                           SWAP.W  R1,R1           ;
0000101E 051A                           STS     MACL,R5         ;
00001020 201E                           MULU    R1,R0           ; Upper 16 bit + upper 16 bit -> R2
00001022 6009                           SWAP.W  R0,R0           ;
00001024 021A                           STS     MACL,R2         ;
00001026 201E                           MULU    R1,R0           ; Lower 16 bit + upper 16 bit -> R4
00001028 6119                           SWAP.W  R1,R1           ;
0000102A 041A                           STS     MACL,R4         ;
                                                                ;
0000102C 0008                           CLRT                    ;
0000102E 345E                           ADDC    R5,R4           ;
00001030 0629                           MOVT    R6              ; R6 <- Carry
00001032 4628                           SHLL16  R6              ;
00001034 326C                           ADD     R6,R2           ; Carry = 1  R2 <- R2 + H'00010000
                                                                ; Carry = 0  R2 <- R2 + H'00000000
00001036 6543                           MOV     R4,R5           ;
00001038 4528                           SHLL16  R5              ;
0000103A 4429                           SHLR16  R4              ;
```

```
 50                      50                              ;
 51 0000103C 0000        51          CLRT                ;
 52 0000103E 335E        52          ADDC    R5,R3       ;
 53 00001040 0629        53          MOVT    R6          ; R6 <- Carry
 54 00001042 326C        54          ADD     R6,R2       ; Carry = 1  R2 <- R2 + H'00000001
 55                      55                              ; Carry = 0  R2 <- R2 + H'00000000
 56 00001044 324C        56          ADD     R4,R2       ;
 57                      57                              ;
 58 00001046 210A        58          XOR     R0,R1       ; Product < 0 ?
 59 00001048 4100        59          SHLL    R1          ;
 60                      60                              ;
 61 0000104A 8B02        61          BF      MULS32_END  ; No
 62 0000104C 0008        62          CLRT                ; Change minus
 63 0000104E 633A        63          NEGC    R3,R3       ;
 64 00001050 622A        64          NEGC    R2,R2       ;
 65 00001052             65  MULS32_END                  ;
 66 00001052 66F6        66          MOV.L   @R15+,R6    ; Return register
 67 00001054 65F6        67          MOV.L   @R15+,R5    ;
 68 00001056 64F6        68          MOV.L   @R15+,R4    ;
 69 00001058 000B        69          RTS                 ;
 70 0000105A 4F16        70          LDS.L   @R15+,MACL  ;
 71                      71          .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.