[Notes]
C/C++ Compiler Package for RX Family (No. 65)

R20TS0857EJ0100
Rev.1.00
Aug. 01, 2022

Outline

When using the C/C++ Compiler Package for RX Family, note the following point.

1. Use of the -branch_chaining Option (No. 65)*

*The number in the parentheses is the identification number of the note.

1. Use of the -branch_chaining Option (No. 65)

1.1 Applicable Products

CC-RX V3.04.00

1.2 Details

When the -branch_chaining option is enabled, the generated code may vary each time the source code is compiled. Note that, even if this problem occurs, the generated code still operates as specified in the source file.

1.3 Conditions

The problem may occur when all the following conditions are met.

(1) Neither the -optimize=0 nor -optimize=1 option is specified.

(2) The -speed option is not specified.

(3) The -nobranch_chaining option is not specified.

(4) A function with the #pragma inline_asm directive is called in a function.

(5) The calling function in (4) contains a branch.

[Example] ccrx -isa=rxv2 tp.c // (1)(2)(3)

```
/* tp.c */
#pragma inline_asm asmFun    // (4)
void asmFun() {
              NOP
}

void fun(int a, int b) {
  if(a) {              // (5)
    asmFun();          // (4)
    if(b) {            // (5)
      fun2();
    }
  }
}
```

[Generated codes] Pattern 1

```
_fun:
                .STACK  _fun=8
                PUSH.L R6
                CMP #00H, R1
                MOV.L R2, R6
                BEQ L14
L12:    ; if_then_bb
                ._LINE_TOP  inline_asm
                NOP
                ._LINE_END  inline_asm
                CMP #00H, R6
                BEQ L14
L13:    ; if_then_bb7
                POP R6
                BRA _fun2
L14:    ; return
                RTSD #04H, R6-R6
```

[Generated codes] Pattern 2

```
_fun:
                .STACK  _fun=8
                PUSH.L R6
                CMP #00H, R1
                MOV.L R2, R6
                BEQ L13
L12:    ; if_then_bb
                ._LINE_TOP  inline_asm
                NOP
                ._LINE_END  inline_asm
                CMP #00H, R6
L13:    ; if_then_bb
                BEQ L15
L14:    ; if_then_bb7
                POP R6
                BRA _fun2
L15:    ; return
                RTSD #04H, R6-R6
```

In Pattern 1, a single conditional branch causes a branch up to the end of the function. In Pattern 2, a branch up to the end of the function is generated via two conditional branches. Both patterns show the codes that operate as specified in the C source file.

1.4 Workaround

You can avoid this problem by doing one of the following:

(1) Specify the -optimize=0 or -optimize=1 option.

(2) Specify the -speed option.

(3) Specify the -nobranch_chaining option.

1.5 Schedule for Fixing the Problem

This problem will be fixed in CC-RX V3.05.00. The release date has not been determined.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Aug.01.22	-	First edition issued

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.

The past news contents have been based on information at the time of publication. Now changed or invalid information may be included.

The URLs in the Tool News also may be subject to change or become invalid without prior notice.

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/

© 2022 Renesas Electronics Corporation. All rights reserved.

TS Colophon 4.3