[Notes]

C/C++ Compiler Package for RX Family

Outline

When using the CC-RX C/C++ compiler package for the RX family, note the following point.

1. Loop statements with loop-control variables in which constants are used as the condition for ending the loop (CCRX#048)

   Note: The number which follows the description of a precautionary note is an identifying number for the precaution.

1. Loop Statements with Loop-Control Variables in Which Constants are Used as the Condition for Ending the Loop (CCRX#048)

1.1   Applicable Products

CC-RX V2.00.00 to V2.07.00

1.2   Details

If any of the following items exists in a loop with a loop-control variable in which an integer constant is used as the condition for ending the loop, the loop repeat count may be incorrect.

- Multiplication expression with a loop-control variable and integer constant

- Loop induction variable whose incremental value is an integer multiple of the incremental value of a loop-control variable

1.3   Conditions

This problem may arise if all of the conditions from (1) to (5) are met.

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

(2) A loop with a loop-control variable[Note 1] exists, and the condition for ending the loop is a comparison between the loop-control variable and an integer constant[Note 2].

(3) Either (3-1) or (3-2) exists in the loop of (2).

   (3-1) Multiplication expression that contains the loop-control variable and integer constant[Note 2] of (2) as the operands

   (3-2) Loop induction variable[Note 1] whose incremental value is an integer multiple[Note 3] of the incremental value of the loop-control variable of (2)

(4) The loop of (2) does not contain a loop induction variable whose incremental value is the same as the incremental value of the loop-control variable of (2).

(5) The loop-control variable of (2) satisfies either (5-1) or (5-2).

   (5-1) The type of the loop-control variable is signed long long, unsigned long long, or pointer.

   (5-2) The loop-control variable is used in the loop of (2) as an operand of an operation expression that contains 15 or more operands.

   Note 1: Loop-control variable: Loop induction variable used in loop end determination

         Loop induction variable: Variable that is incremented by a fixed value each time in a loop

   Note 2: Includes cases in which such a value is statically known to be a constant.

   Note 3: Zero-fold and one-fold are not applicable.

1.4 Example

The following is an example of the problem. Characters in red are the parts that correspond to the conditions.

[C source]

```
 1    #pragma inline_asm sub
 2    static void sub(long long arg) {}
 3    long long func( void )
 4    {
 5      long long count;                        // Condition (5-1)
 6      long long result = 0;
 7
 8      for ( count = 0; count < 6; count++ ) {  // Condition (2)
 9        result = result * 2 + count * 10;      // Condition (3-1)
10        sub(result);
11      }
12      return result;
13    }
```

- Line 8: Corresponds to Condition (2) and Condition (5) because this loop uses "count", a loop-control variable in the long long type, and uses comparison with "6", an integer constant, as the condition for ending the loop.

- Line 9: Corresponds to Condition (3) because a multiplication expression with the loop-control variable "count" and integer constant "10" exists.

- Lines 8 to 11: Corresponds to Condition (4) because the loop does not contain a loop induction variable whose incremental value is the same as the incremental value "+1" of the loop-control variable "count".

Supplement: In the above example, although the loop of lines 8 to 11 should be executed 6 times for the correct operation, the loop is actually executed only 4 times.

1.5 Workaround

To avoid this problem, take either of the following (1) and (2) steps.

(1) Specify either of the "-optimize=0" and the "-optimize=1" options.

(2) Modify the loop-control variable of Condition (2) or the loop induction variable of Condition (3-2) by declaring it as volatile.

1.6 Schedule for Fixing the Problem

This problem will be fixed in CC-RX V2.08.00.
This revision will be available from January 22.

Revision History

Rev.	Date	Description	
		Page	**Summary**
1.00	Jan. 16, 2018	-	First edition issued

TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061 Japan

Renesas Electronics Corporation

■Inquiry

https://www.renesas.com/contact/

All trademarks and registered trademarks are the property of their respective owners.