[Notes]
C/C++ Compiler Package for RX Family (No. 68)

R20TS1082EJ0100
Rev.1.00
Dec. 05, 2024

Outline

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

1. Note on counting of the inner loop in a nested loop structure (No. 68)

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

1. Note on counting of the inner loop in a nested loop structure (No. 68)

1.1 Applicable Products

For Windows: CC-RX V2.00.00 to V3.06.00

For Linux: CC-RX V3.06.00 to V3.06.02

1.2 Details

The result of counting of the inner loop in a nested loop structure might differ from that actually stated in the source code.

1.3 Conditions

[Conditions]

The problem might occur when all conditions (1) to (6) listed below are met.

(1) The -optimize=2 or -optimize=max option is specified as the optimization level.
(2) The program includes a nested loop.
(3) The loop control variable of the outermost loop is of the integer type and is not declared as volatile.
(4) The loop control variable of the inner loop is of the 1- or 2-byte integer type and is not declared as volatile.
(5) In both loops (3) and (4) above, the condition for ending the loop is comparison between an expression for which the value does not change in the loop (e.g., a constant) and a loop control variable.
(6) In loop (4) above, the program only enters the loop once (non-repeated loop).

[Example]

In this example, counting of the inner loop is wrong and the program enters the loop twice although it should only enter the loop once. Thus, the return value of the test() function is 6 although it should be 2.

```
volatile int vi = 0;
int test(void) {
  int result = 0;
  int i;                    /* (3) */
  short j;                  /* (4) */

  for (i = 0; i < 2; i++) {  /* (2)(3)(5) */

    j = 1;
    do {

      result += j;
```

```
      if (vi != 0) {
        break;
      }

      j++;
    } while (j == 1);          /* (2)(4)(5)(6)*/

  }
  return result;
}
```

1.4 Workaround

Take any of the following actions.

(a) Specify -optimize=0 or -optimize=1 as the optimization level.
(b) Change the type of the control variable in the inner loop in condition (4) to any type other than that in condition (4) (e.g., the long type).
(c) Declare the control variable in the inner loop in condition (4) as volatile.
(d) Declare the control variable in the outermost loop in condition (3) as volatile.

1.5 Schedule for Fixing the Problem

This problem will be fixed in CC-RX V3.07.00. The release date has not been determined.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Dec.05.24	-	First edition issued

Corporate Headquarters

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

Trademarks

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/

TS Colophon 4.3