[Notes]

C Compiler Package for RL78 Family

R20TS0184EJ0101
Rev.1.01
Jul. 1, 2017

Outline

When using the CC-RL C compiler package for the RL78 family, note the following point.

1. Relational operators in the control expressions of switch statements (CCRL#015)

Note1: The number which follows the description of a precautionary note is an identifying number for the precaution.

Note2: Relational operators include equality operators.

1. Relational Operators in the Control Expressions of switch Statements (CCRL#015)

1.1 Applicable Products

CC-RL V1.00.00 to V1.05.00

1.2 Details

An invalid code may be generated if the control expression of a switch statement is a relational operation or an equality operation.

1.3 Condition

An invalid code may be generated when all of conditions (1) to (5), described below, are met:

(1) The optimization level other than -Onothing is specified, or the optimization level is not specified.

(2) The C source code contains a loop statement and the loop statement contains a switch statement.

(3) The switch statement of (2) is a true/false judgment using >, >=, <, <=, ==, or != (Note 1).

(4) The members of the relational operator or the equality operator of (3) is a loop control variable and a constant that do not have a volatile qualifier (Note 2).

(5) The switch statement of (2) has two case labels with case values 0 and 1 (Note 3).

Note 1: This condition is true if the statement contains a relational operator or an equality operator.

For example, the condition is true for switch(i != 0) but not true for switch(i).

Note 2: The right and left members of the operator are interchangeable for the condition to be true; for example, "i > 1" and "1 < i" are applicable.

Even if a constant is not contained, the condition may be true if the compiler optimization regards a variable as a static constant.

Note 3: The condition is true even if the branch destinations of case 0: and case 1: are the same.

The presence or absence of a default label does not matter.

1.4 Example

The following is an example of the problem.

When the -Onothing option is not designated: Condition (1)

```
unsigned char a;
int main( void )
{
  int i = 0;              /* Condition (4) */
  while(1) {              /* Condition (2) */
    switch( i < 1 ){      /* Condition (2) (3) (4) */
      case 1 :            /* Condition (5) */
        a = 20;
        break ;
      case 0 :            /* Condition (5) */
        goto end_l ;
        break ;
    } ;
    i++ ;
  }
end_l:
  return( 0 ) ;
}
```

In the above example, case 1 of the switch statement is executed when i equals 0, but the output code does not set a to 20.

1.5 Workaround

To avoid this problem, take any of the following steps:

(1) Specify the optimization level option as -Onothing.

(2) Replace the switch statement of condition (2) with an if statement.

(3) Modify the loop control variable in Condition (4) by adding the volatile qualifier.

1.6 Schedule for Fixing the Problem

The problem had been fixed in CC-RL V1.06.00.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jul. 1, 2017	-	First edition issued
1.01	Jan.16, 2021	1	1.1 Applicable Products(V1.04.00 -> V1.05.00)
1.01	Jan.16, 2021	2	1.6 Schedule for Fixing the Problem(V1.05.00 -> V1.06.00)

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

Renesas Electronics Corporation

■Inquiry

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