

## Notes on Using CubeSuite+ CC-RX Compiler V2.00.00

When using CubeSuite+ CC-RX compiler V2.00.00, take note of the following problems:

- When the repeat count of the loop differs from that written in the source file (RXC#023)
- When the -optimize=0, -debug, and -schedule options are all used (RXC#024)

The number here at the end of the above items is for indexing the problems in this compiler.

---

### 1. Product and Version Concerned

The following version of the product has both of the above problems:

CubeSuite+ CC-RX Compiler V2.00.00

### 2. Problem When the Repeat Count of the Loop Differs from that Written in the Source File (RXC#023)

#### 2.1 Description

Optimization by the compiler may generate a code whose repeat count of the loop differs from that written in the C/C++ source file.

In such a case, the program cannot be executed correctly.

#### 2.2 Conditions

This problem may arise if the following conditions are all satisfied:

- (1) The optimize=2 or optimize=max option is used.
- (2) A loop control variable is a loop that uses any one of the following types.
  - signed char
  - signed short
  - unsigned char
  - unsigned short
- (3) A variable is used for the value to terminate the loop.

However, optimization of inline expansion of a function changes this value to a constant.

#### 2.3 Example

When the compile option "ccrx -cpu=rx600 -optimize=max -speed" is used: // Condition (1)

-----  
#include

```
short sub( short arg )
```

```
{  
    short i = 1, j = 0 ;    // Condition (2)  
    while( i ) i-- ;  
    while( ++i <= arg ) j++ ; // Condition (3)  
    return( j ) ;  
}
```

```
int main( void )
```

```
{  
    short result ;  
    result = sub( 1 ) ;    // Condition (3)  
    if( result != 1 ) {  
        printf( ""%-12s %04d:NG...[1]--->[%d]¥n"" , __FILE__ , __LINE__ ,  
            result ) ;  
    } else {  
        printf( ""%-12s %04d:OK¥n"" , __FILE__ , __LINE__ ) ;  
    }  
    return( 0 ) ;  
}
```

-----  
When the above example is compiled, the execution count of the while statement does not match the original count written in the C/C++ source file (the return value does not become 1) and NG is output at the printf statement.

## 2.4 Workarounds

To avoid the problem, do one of the following:

- (1) Use the optimize=0 or optimize=1 option.
- (2) Change the type of the loop control variable to signed int or unsigned int.
- (3) Qualify the variable of the value which terminates the loop to be volatile.

## 3. Problem When the -optimize=0, -debug, and -schedule Options are All Used (RXC#024)

### 3.1 Description

When compiling is performed with the -optimize=0, -debug, and -schedule options all used, an internal error may occur. (See NOTE)

Note that when an internal error occurs, no code will be generated.

**NOTE:**

Even though the above three options are all used, an internal error does not occur in some cases due to the contents of the source file.

### **3.2 Workaround**

Avoid using all the -optimize=0, -debug, and -schedule option simultaneously.

### **4. Schedule for Fixing the Problem**

Both of the problems have been fixed in CubeSuite+ CC-RX compiler V2.00.01.

For details on V2.00.01, refer to RENESAS TOOL NEWS Document No. 130616/tn2.

The details can also be referenced from the following URL.

<https://www.renesas.com/search/keyword-search.html#genre=document&q=130616tn2>

This page will be published on June 20.

---

**[Disclaimer]**

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.

© 2010-2016 Renesas Electronics Corporation. All rights reserved.