

Notes on Using CX Compiler together with CubeSuite+

When using the CX compiler (for the V850E2M CPU core) together with CubeSuite+, take note of the following problems:

- With if statements containing two or more comparison expressions (No. 10)
 - With functions containing infinite loops with no instructions (No. 11)
-

1. Problem with if Statements Containing Two or More Comparison Expressions (No. 10)

1.1 Version Concerned

CubeSuite+ CX compiler V1.20

1.2 Description

Suppose that the conditional expression of an if statement consists of two or more comparison expressions successively joined by the Logical OR (||) operator. If some of these expressions have consecutive values of integer constants, all the comparison expressions except the one that compares the smallest constant may be deleted.

However, this problem may not arise depending on optimization and others.

1.3 Conditions

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

- (1) Optimizing option -O, -Osize, or -Ospeed is used.
- (2) In the program exists an if statement that satisfies the following:
 - (2-1) Its conditional expression contains two or more comparison expressions in which integer constants including 0 are compared with any one of the following variables:
 - simple variable, pointer variable, element of an array, and member of a structure or union
 - (2-2) The comparison expressions in (2-1) are successively joined by the Logical OR (||) operator.
- (3) Some of the comparison expressions in (2-1) have consecutive

values of integer constants including 0.

(4) The variable in (2-1) is declared to be of the type associated with a new name that is defined by a typedef statement; the type is any of the following

- (signed) short
- unsigned short
- (signed) char
- unsigned char
- _Bool
- enum

Example:

```
-----  
typedef short SS;  
SS ary[10];  
int sss;  
  
.....  
if( ary[0] == 0 || ary[0] == 1 || ary[0] == 5){  
    sss = 0;  
}  
else {  
    sss = 1;  
}  
-----
```

In this example, the comparison expression `ary[0] == 1` is deleted in error.

1.4 Workaround

To avoid this problem, use either of the methods described bellow. By using a checking tool, you can locate the line of the comparison expression or expressions deleted in error. To obtain the check tool, contact your local Renesas Electronics marketing office or distributor.

(1) Use no typedef statement for the variable in Condition (2-1).

```
-----  
short ary[10];  
  
.....  
if( ary[0] == 0 || ary[0] == 1 || ary[0] == 5){  
-----
```

(2) Store the variable in Condition (2-1) on a temporary variable; then use it.

```
-----  
if( ary[0] == 0 || ary[0] == 1 || ary[0] == 5){
```

```

typedef short SS;
SS ary[10];
SS tmp = ary[0];
.....
if( tmp == 0 || tmp == 1 || tmp == 5){
-----

```

1.5 Schedule of Fixing the Problem

We plan to fix this problem in a future version.

2. Problem with Functions Containing Infinite Loops with No Instructions (No. 11)

2.1 Version Concerned

CubeSuite+ CX compiler V1.00 through V1.11

2.2 Description

If a function containing an infinite loop with no instructions is compiled, compilation may not be complete.

2.3 Conditions

This problem arises if the following conditions are all satisfied:

- (1) Optimizing option -O, -Osize, or -Ospeed is used.
- (2) A call is made to an interrupt function or a function that is no interrupt function and has parameters.
- (3) The beginning statement of the function in (2) is an infinite loop with no instructions.

Example 1:

```

-----
#pragma interrupt INTSW0 func1
void func1(void)
{
  for(;;);
.....
-----

```

Example 2:

```

-----
void func2(int a)
{
  for(;;);
.....
-----

```

2.4 Workaround

To avoid this problem, place "__asm("¥n");" in the beginning line of the function.

Example 1 modified:

```
-----  
#pragma interrupt INTSW0 func1  
void func1(void)  
{  
__asm("¥n"); // Placed  
for(;;);  
.....  
-----
```

2.5 Schedule of Fixing the Problem

This problem has already been fixed in the CubeSuite+ CX compiler V1.20, which is included in the CubeSuite+ package.

[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.