[Notes]

C Compiler Package for RL78 Family

R20TS0105EJ0100
Rev.1.00
Dec. 16, 2016

Outline

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

1. Updating of values of array elements, structure members, or union members not being reflected (CCRL#013)

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

1. Updating of Values of Array Elements, Structure Members, or Union Members Not being Reflected (CCRL#013)

1.1 Applicable Products

CC-RL V1.00.00 to V1.03.00

1.2 Details

There may be cases where updating of values of array elements, structure members, or union members is not reflected.

1.3 Condition

When all of the following conditions (1) to (5) are satisfied, values of array elements, structure members, or union members which are to be referenced in (3) might be the values before updating in (4).

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

(2) Array elements, structure members, or union members are referenced. When structure members or union members are referenced, the following conditions (2-a) and (2-b) are also satisfied:

(2-a) When structure members are referenced

The first member is an array.

(2-b) When union members are referenced

An array member is contained.

(3) The same array elements, structure members, or union members as (2) are referenced after reference in (2). If the reference processing in (2) is in the loop, the reference in (2) that is repeatedly executed in the loop is also contained. Additionally, reference in the called functions is also contained.

(4) Values of the same array elements, structure members, or union members as those referenced in (2) are updated between references in (2) and (3).(*1)

*1: The condition applies only to updating under the state where all of the following conditions (4-1) to (4-4) are satisfied.

(4-1) Updating is done by any of the following items.

(4-1-a) For array elements

- Array address

or

- Pointer that indicates the address of any element contained in the array

(4-1-b) For structure members

- Structure address

or

- Pointer that indicates the address of any member contained in the structure

(4-1-c) For union members

- Union address

or

- Pointer that indicates the address of any member contained in the union

(4-2) Updating is started for array elements, structure members, or union members that are different from those referenced in (2) and (3).

(4-3) Units for updating are the size(*2) of array elements, structure members, or union members for which updating is to be started.
*2: If the structure members or union members are the array type, the size of their element types is also contained.

(4-4) Either of the following relationships (4-4-a) and (4-4-b) is established between array elements, structure members, or union members that are referenced in (2) and (3), and those for which updating is started in (4).

(4-4-a) "First address of elements or members located at the larger address" - "End address of elements or members located at the smaller address" > Size referenced in (3)

(4-4-b) "First address of elements or members located at the larger address" - "End address of elements or members located at the smaller address" > Size of the area to be updated at one time in (4)

(5) There is no updating of values of the same array elements, structure members, or union members as those referenced in (2) between references in (2) and (3) except for updating in (4).

1.4 Example

```
1:#include <string.h>
2:typedef struct test {
3:unsigned char  param01[8];   // Condition (4-2): Structure member for which
updating is to be started
4:unsigned char  param02;
5:unsigned short param03;
6:unsigned int   param04;            // Conditions (2) and (3): Structure member
to be referenced
7:} test_t;
8:
9:int test_func(void) {
10:test_t        t1;
11:test_t        t2;
12:int          ret = 1;
13:unsigned char  *src;
14:unsigned char  *dst;
15:int          size;
16:memset(&t1, 0, sizeof(test_t));
17:if (t1.param04 == 0) {                  // Condition (2)
18:t2.param04 = 10;
19:src = (unsigned char *)&t2;
20:dst = (unsigned char *)&t1;
21:size = sizeof(test_t);
22:while (size-- > 0){
23:*dst++ = *src++;     // Condition (4): Updating of structure variable t1
24:}
25:}
26:if (t1.param04 < 5) {            // Condition (3)
27:ret = -1;
28:}
29:return(ret);
30:}
```

Line 23 corresponds to Condition (4-1) because updating is done by a structure address. Additionally, a difference between the first address of t1.param04 referenced in Conditions (2) and (3) and the end address of t1.param01 for which updating is started in Condition (4-2) is 3 bytes (when structure packing is performed) or 4 bytes (when structure packing is not performed). Therefore, the size of the difference is larger than the size (1 byte in the unsigned character type) to be updated in Condition (4-3), and consequently Condition (4-4) is applied. This is why line 23 applies to updating in Condition (4).

Condition (5) is also applied because there is only updating in Condition (4) from reference in line 17 to reference in line 26 for t1.param04.

As a result, updating from line 22 to line 24 is not reflected in the t1.param04 value to be referenced in line 26.

1.5 Workaround

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

(1) Specify the optimization level as -Onothing.

(2) Change the updating in Condition (4) to either of the following ways.

- For a structure, change the way to structure assignment as follows.

[Before modification]

```
19:src = (unsigned char *)&t2;
20:dst = (unsigned char *)&t1;
21:size = sizeof(test_t);
22:while (size-- > 0){
23:*dst++ = *src++;
24:}
```

[After modification]

```
19:t1 = t2;
```

- Change the way to calling of the memcpy library function as follows.

[Before modification]

```
22:while (size-- > 0){
23:*dst++ = *src++;
24:}
```

[After modification]

```
22:memcpy(dst, src, size);
```

1.6 Schedule for Fixing the Problem

This problem will be fixed in the next version.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Dec. 16, 2016	-	First edition issued

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

■Inquiry
http://www.renesas.com/contact/