RENESAS Tool News

RENESAS TOOL NEWS on December 1, 2007: 071201/tn1

A Note on Using the C Compiler Packages for the M16C Family of MCUs

Please take note of the problem described below in using the C compiler packages for the M16C family of MCUs.

1. Products and Versions Concerned

C compiler package for the M32C series* (M3T-NC308WA)
V.5.00 Release 1 through V.5.40 Release 00

C compiler package for the M16C series** (M3T-NC30WA)
V.5.10 Release 1 through V.5.40 Release 00A

* The M32C series is the generic name of the M32C/80, and M16C/80 series.
** The M16C series is the generic name of the M16C/60, /30, /20, /10, /Tiny and R8C/Tiny series.

2. Description

If a write to and a read from a bit field are made before and after calling a function respectively, the value of the bit field may become incorrect.

3. Conditions

This problem may occur if the following conditions are all satisfied:
(1) Any of the optimizing options -O[1-5],-OR, and -OS is selected.
(2) A bit field is used whose storage class is auto.
(3) The bit field in (2) is not qualified to be volatile.
(4) A write is made to the bit field in (2).
(5) The program performs either of the following after the write in (4):
  (a) A call is made to the function that takes the address of the structure containing the bit field in (2) as its argument.
  (b) A call is made to a function immediately after the address of

the structure containing the bit field in (2) is assigned to another variable.

(6) The value of the bit field in (2) is changed within either of the functions in (5)-(a) and (5)-(b).

(7) The bit field in (2) is accessed after a call is made to either of the functions in (5)-(a) and (5)-(b)

Example:

```
struct tag{      // Conditions (2) and (3)
    unsigned char b0:1;
    unsigned char b1:1;
    unsigned char b2:1;
    unsigned char b3:1;
    unsigned char b4:1;
    unsigned char b5:1;
};

main()
{
    struct tag bit2;

    bit2.b1 = 1;     // Condition (4)
    func(&bit);      // Condition (5)-(a)

    if(bit2.b1==1)   // Condition (7)
    sub();
}
```

4. Workaround

Place a dummy asm function between a write to the bit field and a call to the function involved.

```
. . . . . .
main()
{
    struct tag bit2;

    bit2.b1 = 1;
    asm();      // Place asm function
    func(&bit);
. . . . . .
```

---

5. Schedule of Fixing the Problem

This problem has already been resolved in the following versions of the products concerned:

- C compiler package for the M32C series (M3T-NC308WA) V.5.41 Release 00
- C compiler package for the M16C series (M3T-NC30WA) V.5.42 Release 00

So use these or later versions.