

RENESAS TOOL NEWS on June 16, 2008: 080616/tn4

A Note on Using the C Compiler Packages for the M16C MCU Family --With Using a Volatile-Qualified Variable in the Bit-Wise AND Operation--

Please take note of the following problem in using the C compiler package for M16C MCU family:

- With using a volatile-qualified variable in the bit-wise and operation
-

1. Products and Versions Concerned

- (1) C compiler package for the R32C/100 series
V.1.01 Release 00
- (2) C compiler package for the M32C series (M3T-NC308WA) (See NOTE 1)
V.1.00 Release 1 through V.5.41 Release 01
- (3) C compiler package for the M16C series (M3T-NC30WA) (See NOTE 2)
V.3.10 Release 1 through V.5.44 Release 00

NOTES:

1. The M32C series is the generic name of the M32C/80, M16C/80, and M16C/70 series.
2. The M16C series is the generic name of the M16C/60, /30, /20, /10, /Tiny, and R8C/Tiny series.

2. Description

In the controlling expression of an if statement, if a volatile-qualified variable is bit-wise ANDed with a constant of 2 to the power of n (that is, only a single bit is set to 1 among its bits), and the true or false statement is executed according to the result of the evaluation, part of the variable may not be accessed.

2.1 Conditions

This problem occurs if the following conditions are all satisfied:

- (1) Any of the optimizing options -O[1-5], -OR, and -OS is selected.

- (2) A variable is qualified as volatile.
- (3) In the controlling expression of an if statement, the variable in (2) is bit-wise ANDed with a constant.
- (4) The constant in (3) is a value of 2 to the power of n.

2.2 Example

```
-----  
#pragma BIT v  
extern volatile unsigned int v;  
extern unsigned int i, j;  
void func(void)  
{  
    if (v & 0x0004U) { // btst 02H,_v generated and byte-accessed.  
        i = j;  
    }  
    j = 0;  
}
```

```
-----
```

3. Workaround

Declare a temporary variable tmp and assign the volatile-qualified variable to it; then perform the bit-wise AND operation between the constant and tmp.

```
-----  
#pragma BIT v  
extern volatile unsigned int v;  
extern unsigned int i, j;  
void func(void)  
{  
    unsigned int tmp; // A tmp variable placed.  
    tmp = v; // tmp word-accessed.  
    if (tmp & 0x0004U) { // tmp bit-wise ANDed.  
        i = j;  
    }  
    j = 0;  
}
```

```
-----
```

4. Schedule of Fixing the Problem

We plan to fix this problem in the next release of the products concerned.

[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.