RENESAS Tool News

RENESAS TOOL NEWS on June 16, 2008: 080616/tn2

A Note on Using the C Compiler Packages for the M16C MCU Family --With Comparing a Bit Field Variable with 1 or 0--

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

- With comparing a bit field variable with 1 or 0

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.5.10 Release 1, and
V.5.40 Release 00 through V.5.41 Release 01
(3) C compiler package for the M16C series (M3T-NC30WA) (See NOTE 2)
V.2.00 Release 1 through V.5.20 Release 1, and
V.5.40 Release 00 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 an if-else statement, incorrect code may be generated if a bit field variable of 1 bit long is compared with constant 1 or 0 using the "==" or "!=" operator, the constant is assigned to the variable in the true or false statement after the comparison, and the variable is used in another expression.

2.1 Conditions

This problem occurs if the following conditions are all satisfied:

(1) The Generated code modification option -fenable_register (-fER) is selected.
(2) A bit field of 1 bit long is compared with constant 1 or 0 in an if-else statement.
(3) The "==" or "!=" operator is used for the comparison in (2).
(4) The same constant as in (2) is assigned to the variable in the true or false statement.
(5) The variable in (4) is used only once in another expression.

2.2 Example

```
-----------------------------------------------------------------------
  #pragma BIT s
  #pragma BIT gi
  extern struct {
     unsigned int   b0:1;
  } s;
  extern unsigned int    gi;
  void func(void)
  {
     register unsigned int   t;
     if (s.b0 == 1) {              // Conditions (2) and (3)
        t = 1;                  // Condition (4)
     } else {
        t = 0;
     }
     t = gi;   // Condition (5); code for assignment expression
            // s.b0 = gi; generated.
  }
-----------------------------------------------------------------------
```

3. Workaround

Place an asm() function in the if or else statement in the if-else construct.

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.