RENESAS Tool News

RENESAS TOOL NEWS on April 16, 2008: 080416/tn1

A Note on Using the C Compiler Packages for the R32C/100 and the M32C Series of MCUs

Please take note of the following problem in using the C compiler package for the R32C/100 series, M16C family, and the one for the M32C series* of MCUs, M3T-NC308WA:

- With issuing preprocessing directive #pragma DMAC to extend the C compiler's functionality
  *The M32C series is the generic name of the M32C/80 and M16C/80 series.

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)
V.5.00 Release 1 through V.5.41 Release 01

2. Description

If a variable specified with #pragma DMAC is used in a controlling expression, assemble errors may arise.

3. Conditions

This problem occurs if the following conditions are all satisfied:

(1) Any of the optimizing options -O1, -O2, -O3, -O4, -O5, -OR, -OS, -OR_MAX and -OS_MAX are selected.

(2) An external variable is specified with #pragma DMAC.

(3) The external variable in (2) and an integer constant are bitwise-ANDed in the controlling expression in an if, while, or for statement.

(4) The integer constant in (3) is 2 to the nth power (1, 2, 4, 8, etc).

Example in C compiler package for R32C/100 series:

```
------------------------------------------------------------------
  unsigned char val;
  unsigned long dmd0;
```

```
 #pragma DMAC dmd0 DMD0        /* Condition (2) */

 void func(void)
 {
    if(dmd0 & 0x00000001L) {   /* Conditions (3) and (4) */
       val++;
    }
 }
-------------------------------------------------------------------
```

4. Workaround

Assign the variable specified with #pragma DMAC to a temporary variable, and then evaluate it in the controlling expression.
Example:

```
-------------------------------------------------------------------
 unsigned char val;
 unsigned long dmd0;
 #pragma DMAC dmd0 DMD0

 void func(void)
 {
    unsigned long tmp;

    tmp = dmd0;                 /* Assigned to tmp */
    if(tmp & 0x00000001L) {
       val = 0x1;
    }
 }
-------------------------------------------------------------------
```

5. Schedule of Fixing the Problem

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

[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.