Tool News

RENESAS TOOL NEWS on June 16, 2008: 080616/tn3

A Note on Using the C Compiler Packages for the M16C MCU Family --With Using a Variable Volatile-Qualified with the Indirect Member Operator--

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

- With using a variable volatile-qualified with the indirect member operator

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.1.00 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

When in the program exists an expression in which its left term (a structure to be referenced) and/or its right term (a member of the structure) are qualified as volatile with the indirect member operator, incorrect code may be generated where the structure and member are interpreted as non-volatile.

2.1 Conditions

This problem occurs if the following conditions are all satisfied:
(1) The "->" operator is used.

(2) The left and/or right terms of the operator in (1) (a structure to be referenced and its member respectively) are qualified as volatile.

(3) The type of the left term in (1) is other than a volatile-qualified pointer type.

2.2 Example

```
----------------------------------------------------------------------
   struct S1 {
         int m;
   };
   struct S1 volatile *p;
   void func(void)
   {
         p->m;      // A volatile-qualified object referenced.
   }
----------------------------------------------------------------------
```

The code that references volatile-qualified objects is not generated.

3. Workaround

Avoid this problem by qualifying the pointer as volatile.

Example:

```
----------------------------------------------------------------------
   struct S1 {
         int m;
   };
   struct S1 volatile * volatile p;
   void func(void)
   {
         p->m;
   }
----------------------------------------------------------------------
```

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.