RENESAS Tool News

RENESAS TOOL NEWS on June 16, 2008: 080616/tn1

A Note on Using the C Compiler Packages for the M16C MCU Family
--With Using a Volatile-Qualified Variable in the for or while Statement--

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 for or while statement

---

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 the evaluation of the controlling expression can be omitted only in the first loop of a for or while statement by having used the optimization option in compiling, optimization that user used may be performed and the evaluation of the controlling expression be omitted the first time even if the controlled variable is qualified as volatile to avoid omitting the evaluation of the controlling expression.

2.1 Conditions

This problem occurs if the following conditions are all satisfied:

(1) Any of the optimizing options -O[3-5], -OR, and -OS is selected.
(2) The optimizing option -Ono_break_source_debug(-ONBSD) is not selected.
(3) In a for or while statement exist the loops on entry to which the evaluation of the controlling expression is made.
(4) The controlled variable of the for or while statement is qualified as volatile.
(5) The controlled variable is compared with the constant specifying the number of loops in the controlling expression.
(6) A constant is assigned to the variable in (5) immediately before the comparison is made.

2.2 Example

```
---------------------------------------------------------------------
  extern int x;
  void func(void)
  {
     volatile int i;
     for(i = 0; i < 10; ++i){
        ++x;
     }
  }
---------------------------------------------------------------------
```

3. Workarounds

Avoid this problem in either of the following ways:
(1) Do not compare the controlled variable with a constant but with a variable.
(2) Use the optimizing option -Ono_break_source_debug(-ONBSD).

4. Schedule of Fixing the Problem

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