RENESAS Tool News

RENESAS TOOL NEWS on February 16, 2006: RSO-M3T-NC30WA-060216D

A Note on Using the C Compiler Package M3T-NC30WA

Please take note of the following problem in using the C compiler package M3T-NC30WA, which is used for the M16C/60, M16C/30, M16C/20, M16C/10, M16C/Tiny, and R8C/Tiny series of MCUs:

- On comparing a constant with the return value from a function in an if statement

---

1. **Versions Concerned**
   M3T-NC30WA V.5.10 Release 1--V.5.40 Release 00

2. **Description**
   When the controlling expression of an if statement consists of a comparison of a constant with the return value from a function, and the substatement is an expression for assigning a constant to a variable, incorrect code may be generated for the assignment.

   2.1 Conditions
   This problem occurs if the following conditions are all satisfied:

   (1) Any one of the following optimizing options is selected:
   -O1, -O2, -O3, -O4, -O5, -OR, and -OS

   (2) In the program exists an if statement that has no else statement.

   (3) The controlling expression of the if statement in (2) consists of a comparison of a constant with the return value from a function.

   (4) An equality (==) or inequality (!=) operator is used for the comparison in (3).

(5) The substatement of the if statement in (2) is only an expression for assigning a constant to a variable.

(6) The variable in (5) is any one of the following types:
unsigned char, signed char, unsigned int, and signed int

(7) The variable in (5) is the return value from another function.

(8) Another assignment expression same as the one described in (5) exists before the if statement in (2).

2.2 Example

```
------------------------------------------------------------
---------
unsigned char  sub(int);

inline unsigned char func( unsigned char c )
{
 switch(c){
  case 0:
   c = 1;                /* Condition (8) */
   break;
  case 1:
   if ( sub(0) == 1 ){      /* Conditions (2), (3), and (4)
*/
     c = 1;              /* Conditions (5), (6), and (7) */
   }
   break;
  default:
   break;
 }
 return( c );              /* Condition (7) */
}
------------------------------------------------------------
---------
```

3. **Workaround**

   Place a dummy asm function immediately before the substatement of the if statement:

```
-----------------------------------------------------------------
 . . . . . .

    if ( sub(0) == 1 ){
      asm("");    /* Dummy asm function placed */
      c = 1;
    }

 . . . . . .

-----------------------------------------------------------------
```

4. **Schedule of Fixing the Problem**

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