RENESAS Tool News

A Note on Using C Compiler M3T-NC30WA

Please take note of the following problem in using C compiler (with an assembler and integrated development environment) M3T-NC30WA for the M16C/60, M16C/30, M16C/20, and M16C/10 series of MCUs:

- On describing an AND operation of a variable and an immediate value in the conditional expression of the if statement

1. **Versions Concerned**
   M3T-NC30WA V.5.00 Release 1 and Release 2

2. **Description**
   In the conditional expression of the if statement, describing an AND operation of a variable and an immediate value may bring an incorrect result.

3. **Conditions**
   This problem occurs if either the conditions (1) through (5) and (6) or those (1) through (5) and (7) described below are satisfied.

   (1) In the conditional expression of the if statement, an AND operation of a variable and an immediate value is performed; then its result is tested for equality or inequality with a zero.

   (2) The variable's type is any of the following:
   signed long, unsigned long, signed long long, and unsigned long long.

   (3) The immediate value is not a number of powers of 2.

   (4) The uppermost 2 bytes of the immediate value represent a number of powers of 2.

   (5) The each number divided the immediate value except the uppermost 2 bytes into 2 bytes at a time has numerical value other than a number of powers of 2.

   (6) The variable meets any of the following conditions if it is an external variable:
   - The -fbit option is used.

- The variable is declared using the preprocessor directive #pragma SBDATA, #pragma BIT, or #pragma ADDRESS.

(7) The variable meets the following condition if it is an internal variable:
- It is assigned to a register or to an area from -16[FB] to 15[FB].

4. **Example**

```
------------------------------------------------
  void smp(void)
  {
   long al;    /* Conditions (2) and (4); or(2) and (5) */
      :
    if((al & 0x10006000L)!=0){   /* Conditions (1), (3), and (4);
                                    or (1), (3), and (5) */
      :
   }
  }
------------------------------------------------
```

5. **Workaround**

This problem can be circumvented by performing the following procedure:
   (1) Assign the immediate value to a temporary variable.
   (2) Then, place a dummy asm function just after the assignment in (1).

```
------------------------------------------------
  void smp(void)
  {
    long  al;
    long  tmp;
       :
   tmp = 0x10006000L;   /* (1) */
   asm();           /* (2) */
    if(al & tmp){
       :
    }
   }
------------------------------------------------
```

6. **Schedule of Fixing the Problem**
We plan to fix this problem in our next release of the product.