RENESAS Tool News

A Note on Using the C Compiler Package --M3T-NC77WA-- for the 77xx Series of MCUs

Please take note of the following problem on using the M3T-NC77WA C-compiler package, which is used for the 77xx series, 7700 family:

- On overflown bits used in operations

---

1. **Versions Concerned**
   The M3T-NC77WA V.3.00 Release 1 through V.5.20 Release 4

2. **Description**
   Even when overflow occurs during an operation, overflown bits are not used usually in the operation. However, they may be used in some cases.

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

   (1)   Three or more integer constants are calculated successively in a portion of an expression; for instance, "0x4000 * 5 / 2" in the example below.
   Example: x = i + 0x4000 * 5 / 2 + j;

   (2)   The operation of the portion of an expression in (1) is performed as of type signed int.

   (3)   The portion of an expression in (1) contains a part that once overflows and then goes within the range allowed to the signed int type. In the example shown below, the

result of an operation of 0x4000 * 5 is 0x14000, and overflow occurs. However, the next operation of 0x14000 / 2 results in 0xA000, which goes back within the range allowed to the signed int type.
Example: 0x4000 * 5 / 2

2.2 Examples

C Source Program:

```
--------------------------------------------------------------------------
void func(void)
{
   int i = 0x4000 * 5 / 2;        /* Conditions (1), (2), and (3) */
}
--------------------------------------------------------------------------
```

Code Generated by the M3T-NC77WA:

```
--------------------------------------------------------------------------
;## # C_SRC :          int i = 0x4000U * 5 / 2;
     ldm.W   #0a000H,DP:1    ; Must be #2000H,DP:1
--------------------------------------------------------------------------
```

3. Workarounds

This problem can be circumvented either of the following ways:

(1)   Modify the expression involved so that its operation can be performed as of type unsigned int.

Example:

```
-----------------------------------------------------------
-------------
void func(void)
{
   int i = 0x4000U * 5 / 2;    /* Add the type suffix U to 0x4000

                      to make it of type unsigned int */
}
-----------------------------------------------------------
-------------
```

(2)   Split the expression involved into two or more using a temporary variable.

Example:

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

```
-------------
void func(void)
{
    int tmp = 0x4000 * 5;      /* Assign "0x4000 * 5" to
                             tmp to split expression */
    int i = tmp / 2;
}
-----------------------------------------------------------
-------------
```