RENESAS Tool News

RENESAS TOOL NEWS on February 1, 2005: RSO-M3T-NC30WA_2-050201D

A Note on Using C-Compiler Package M3T-NC30WA

--On Right Shifting 4-Byte Variables of Type int--

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

- On right shifting 4-byte variables of type int

1. Versions Concerned

M3T-NC30WA V.4.00 Release 1 through V.5.30 Release 1

2. Description

Performing a right shift of a 4-byte variable of type int and then referencing the value previous to the shift may result in incorrect code being generated.

2.1 Conditions

This problem occurs if the following conditions are all satisfied:

(1) Any one or more of the optimizing options -O3, -O4, -O5, -OR, and -OS are used at compilation.

(2) A 4-byte variable of type int or the result of an operation performed on such a variable is right-shifted, and its result is stored as a 2-byte variable of type int.

(3) The shift count in (2) is greater than 10 and less than 17.

(4) After storing the shift result in (2) as a 2-byte variable of type int, a reference is made to the 4-byte variable before the shift.

2.2 Examples

(1) The case where a 4-byte variable of type int right-shifted:

```
----------------------------------------------------------------
long    m, n;
int     i;

void func(void)
{
    register long   a = m;
    i = a >> 11;      /* Conditions (2) and (3) */
     n = a;           /* Condition (4) */
}
----------------------------------------------------------------
```

(2) The case where the result of an operation performed on a 4-byte variable of type int right-shifted:

```
----------------------------------------------------------------
long  o, p;
int  ii;

void func2(void)
{
  ii = (o - 1) >> 11;    /* Conditions (2) and (3) */
  p = (o - 1);           /* Condition (4) */
}
----------------------------------------------------------------
```

3. **Workaround**

(1) If a 4-byte variable of type int to be shifted is qualified as "register" and compile option "-fenable_register[-fER]" is used, be sure to keep the following:

(a) Don't qualify a 4-byte variable of type int in Condition (2) as "register".

(b) Place a dummy asm function between the line containing the expression for right-shifting a 4-byte variable of type int (Condition (2)) and the line containing the expression for referencing the above variable (Condition (4)).

```
----------------------------------------------------------------------
long    m, n;
int     i;
```

```
void func(void)
{
    long   a = m;    /* (a); Not qualified as "register" */
    i = a >> 11;
  asm();            /* (b); Dummy asm function placed */
     n = a;
}
------------------------------------------------------------------
```

(2) Otherwise, place a dummy asm function between the line containing the expression for right-shifting a 4-byte variable of type int (Condition (2)) and the line containing the expression for referencing the above variable (Condition (4)).

Example (1) The case where a 4-byte variable of type int right-shifted:

```
---------------------------------------------------------------
long    m, n;
int     i;

void func(void)
{
    register long   a = m;
    i = a >> 11;
  asm();                   /* Dummy asm function placed */
    n = a;
}
---------------------------------------------------------------
```

Example (2) The case where the result of an operation performed
            on a 4-byte variable of type int right-shifted:

```
---------------------------------------------------------------
long  o, p;
int  ii;

void func2(void)
{
  ii = (o - 1) >> 11;
  asm();               /* Dummy asm function placed */
  p = (o - 1);
}
---------------------------------------------------------------
```

4. Schedule of Fixing the Problem

This problem has already been fixed in the M3T-NC30WA V.5.30 Release 02 and later.

[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.