RENESAS Tool News

RENESAS TOOL NEWS on May 16, 2005: RSO-M3T-NC30WA_1-050516D

A Note on Using the M3T-NC30WA C-Compiler Package

--On Using the #pragma ADDRESS Directive--

Please take note of the following problem in using the M3T-NC30WA C-compiler package:

- On using the #pragma ADDRESS directive

---

1. Versions Concerned

M3T-NC30WA V.1.00 Release 1 through V.5.30 Release 02
(This C compiler package is used for the M16C/60, M16C/30, M16C/20, M16C/10, M16C/Tiny, and R8C/Tiny series of MCUs.)

2. Description

Bit-accessing a variable defined by using the #pragma ADDRESS directive may result in an assemble error.

2.1 Conditions

This problem occurs if the following conditions are all satisfied:

(1) There exists a variable defined by using the #pragma ADDRESS directive, and the variable is a union of the array type.

(2) The value of the address specified by the #pragma ADDRESS directive in (1) is less than 01FFFH.

(3) The value of the address of the bit-accessed variable is greater than 01FFFH.

2.2 Example

```
-------------------------------------------------------------------
typdef struct{
        unsigned char b0:1;
        unsigned char b1:1;
        unsigned char b2:6;
}BIT;

#pragma ADDRESS bit     01FFEH
BIT     bit[10];
void func(void)
{
        bit[2].b0 = 1;
}
-----------------------------------------------------------------------
```

3. **Workaround**

Bit-access the variable using an asm function.

```
-----------------------------------------------------------------------
typdef struct{
        unsigned char b0:1;
        unsigned char b1:1;
        unsigned char b2:6;
}BIT;

#pragma ADDRESS bit     01FFEH
BIT     bit[10];
void func(void)
{
        asm("   or.b    #01H,$$",bit[2].b0);

}
-----------------------------------------------------------------------
```

4. **Schedule of Fixing the Problem**

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