

## A Note on Using the M3T-NC308WA and M3T-NC30WA C-Compiler Packages

Please take note of the following problem in using the C compiler packages --M3T-NC308WA and M3T-NC30WA--for the M16C MCU family:

- On accessing an array variable in the loop of a for statement

---

### 1. Products and Versions Concerned

- The M3T-NC308WA V.5.00 Release 1 through V.5.40 Release 00 (for the M32C/90, M32C/80, and M16C/80 series)
- The M3T-NC30WA V.5.10 Release 1 through V.5.40 Release 00A (for the M16C/60, M16C/30, M16C/20, M16C/10, M16C/Tiny, and R8C/Tiny series)

### 2. Description

For the statement that accesses an element of an array variable using another variable decremented at each iteration as a subscript to the array, code referencing an incorrect address may be generated.

#### 2.1 Conditions

This problem may occur if the following conditions are all satisfied: Note that here a constant includes a value that is replaced with a constant by calculation during compilation.

- (1) In the program exists a for statement in which none of expressions 1, 2, and 3 shown below is omitted.  
for (expression 1; expression 2; expression 3) statement
- (2) The controlled variable (hereafter called c) of the for statement in (1) is not

declared to be static or extern in its storage class and is of type char or unsigned char.

- (3) Outside of all the functions, an array variable is declared, and the variable is of type short, unsigned short, int, or unsigned int.
- (4) Expression 1 in (1) assigns a constant to variable c.
- (5) Expression 2 in (1) compares variable c with a constant using any one of the operators <, >, <=, >=, and !=.
- (6) Expression 3 in (1) subtracts 1 from variable c.
- (7) In the loop exists a statement that accesses the array in (3) using variable c as a subscript.
- (8) None of the following exists in the loop of the for statement in (1):
  - a. labeled statements (except for case and default labels)
  - b. asm functions
  - c. #pragma ASM and #pragma ENDASM directives
  - d. continue statements that return to the beginning of the for statement in (1)
  - e. break statements for exiting the for statement in (1)
  - f. expressions for incrementing/decrementing the value of variable c in (2)
  - g. expressions for indirect writes to arbitrary memory locations using pointer variables
- (9) The optimizing option -OS is used at compilation.
- (10) Any of the following conditions is met:
  - The optimizing option -OLU is not used. (This option generates the code of the statements in the loop by the number of iterations instead of iteration control code.)
  - The -OLU option is used an unspecified number of times, and the number of iterations of the for loop in (1), which is determined by Conditions (4), (5), and (6), is 5 or less.
  - The -OLU option is used an specified number of times, and the number of iterations of the for loop in (1), which is determined by Conditions (4), (5), and (6), is equal to or less than the number of iterations specified by the -OLU option.
- (11) In the M3T-NC30WA, the array in (3) is placed in the far area. (The M3T-NC308WA is independent of this condition.)

## 2.2 Examples:

Source program

```
-----  
extern int far arr[10];          /* Conditions (3) and (11) */  
  
void func(void)  
{  
    char c;                      /* Condition (2) */  
  
    for (c = 7; c > 4; c--) { /* Conditions (1), (4), (5), (6) */  
        arr[c] = 0;          /* Conditions (7) and (8) */  
    }  
}
```

Code generated by the M3T-NC30WA

```
-----  
_func:  
    mov.b  #07H,R0H  
    mov.w  #000eH,A0  
    mov.b  R0H,R0L  
L1:  
    mov.w  #0000H,R1  
    ste.w  R1,_arr[A0]  
    add.w  #0002H,A0 ; This line must be sub.w #0002H,A0  
    dec.b  R0L  
    cmp.b  #04H,R0L  
    jgtu   L1  
    rts  
-----
```

### 3. Workaround

Add the label of dummy to any of the statements within the for loop.

```
-----  
extern int far arr[10];  
  
void func(void)  
{  
    char c;  
  
    for (c = 7; c > 4; c--) {  
dummy:    arr[c] = 0;          /* This line is labeled dummy */  
    }
```

}  
}

-----

#### 4. **Schedule of Fixing the Problem**

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

---

#### **[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.

© 2010-2016 Renesas Electronics Corporation. All rights reserved.