

A Note on Using C Compiler M3T-NC308WA

Please take note of the following problem in using the M3T-NC308WA C-compiler (with an assembler and integrated development environment) for the M32C/80 and M16C/80 series MCUs:

- On calling a function in an if-else construct
-

1. Versions Concerned

M3T-NC308WA V.3.10 Release 1--V.5.00 Release 1 for the M32C/80 and M16C/80 series MCUs

2. Description

In an if-else construct, when the function called by the program statement executed if the condition is satisfied is different from the one called by the program statement executed if the condition is not satisfied, incorrect code may be generated for the latter function.

3. Conditions

This problem occurs if the following seven conditions are satisfied:

- (1) An if-else construct exists.
- (2) In the if-else construct in (1), the function called by the program statement executed if the condition is satisfied (hereafter called function A) is different from the one called by the program statement executed if the condition is not satisfied (hereafter called function B).
- (3) Functions A and B return no values.
- (4) For functions A and B, function prototypes are declared.
- (5) Functions A and B take one argument each, the two arguments are the same in bit size, and their data types are any of the following:
char, signed char, unsigned char, short, unsigned short, int, unsigned int
- (6) As a result of optimizing the if-else construct in (1) above, its two program statements

are simplified only to call functions A and B and pass an argument to each.

- (7) Also as a result of optimizing the if-else construct, a constant is assigned to each of the arguments passed to functions A and B.

4. Examples

C-language source program:

```
-----  
void funcA(int);          /* Conditions (4) and (5) */  
int funcB(unsigned short); /* Conditions (4) and (5) */  
  
void exam(int cond)  
{  
    if (cond) {          /* Condition (1) */  
        int tmp = 0x1234;  
  
        funcA(tmp);     /* Conditions (2) and (3) */  
    } else {  
        funcB(0x5678); /* Conditions (2) and (3) */  
    }  
}
```

Generated code

```
-----  
.glob $exam  
$exam:  
    cmp.w #0000H,R0 ; cond  
    stzx.w #5678H,#1234H,R0  
    jsr $funcA  
    rts  
-----
```

5. Workaround

Place a dummy asm function immediately before calling either of the functions in the if-else construct.

```
-----  
void funcA(int);  
int funcB(unsigned short);  
  
void exam(int cond)  
{
```

```
if (cond) {
    int tmp = 0x1234;
    asm();      /* A dummy asm function placed */
    funcA(tmp);
} else {
    funcB(0x5678);
}
}
```

6. Schedule of Fixing the Problem

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

[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.