

A Note on Using C compilers M3T-NC308WA and M3T-NC30WA --On Function Calls Using the Indirection Operator--

Please take note of the following problem in using the M3T-NC308WA and M3T-NC30WA C compilers (with an assembler and integrated development environment):

- On function calls using the indirection operator
-

1. Products and Versions Concerned

M3T-NC308WA V.5.00 Release 1 and V.5.10 Release 1
for the M32C/80 and M16C/80 series

M3T-NC30WA V.5.20 Release 1

for the M16C/60, M16C/30, M16C/Tiny, M16C/20, M16C/10, and R8C/Tiny series

2. Description

Function calls using the indirection operator may cause incorrect code to be generated.

3. Conditions

This problem occurs if the following five conditions are satisfied:

- (1) The -OS optimizing option is selected.
- (2) At the processing immediately before ending a function, A, another one, B, is called using the indirection operator.
- (3) A pointer variable pointing to function B is declared outside of function A.
- (4) In the arguments of function A is included an argument passed to the stack.
- (5) The code for "enter #00H" is created at the beginning of function A.

4. Example

```
void (*sub)(void); /* Condition (3) */
:
void func(long l) /* Condition (4) */
{
:
(*sub)(); /* Condition (2) */
}
-----
```

5. Workaround

Place a dummy asm function immediately after a function call using the indirection operator.

```
-----
void func(long l)
{
:
(*sub)();
asm(); /* A dummy asm function placed */
}
-----
```

6. Schedule of Fixing the Problem

We plan to fix this problem in our 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.