RENESAS Tool News

RENESAS TOOL NEWS on April 16, 2004: RSO-M3T-NC308WA-040416D

A Note on Using C-Compiler Package M3T-NC308WA

Please take note of the following problem in using the M3T-NC308WA C-compiler package for the M32C/80 and M16C/80 series MCUs:

- On a division or remainder operation including a variable or constant of type long long

---

1. **Versions Concerned**
   M3T-NC308WA V.5.00 Release 1 through V.5.20 Release 1

2. **Description**
   Performing a division or remainder operation including a variable or constant of type long long overwrites the contents of the address registers incorrectly.

3. **Conditions**
   This problem occurs if the following two conditions are satisfied:
   (1) In the program exists a division or remainder operation.
   (2) Either the dividend or the divisor in operation (1) is a variable or constant of type signed/unsigned long long.

3.1 Example

```
------------------------------------------------------------
#include <stdio.h>
const char far ch[10] = "Hello¥n";

long long  func1(long long ll1,long long ll2)
{
```

```
        long long   ll3;

        ll3 = ll1 / ll2;      /* Conditions (1) and (2) */
        return ll3;
    }

    int  main(void)
    {
        long long  ll;
        int far    *i;

        i = ch;

        ll = func1( 2000, 1000 );

        printf( "%s¥n",i );
    }
    ------------------------------------------------------------
```

4. **Workaround**

   Use the #pragma ASM--#pragma ENDASM directive or an asm() function in the inline assemble functions to save address registers A0 and A1 on the stack before a division or remainder operation begins; then recover them immediately after the operation ends.

   Example:

```
   ------------------------------------------------------------------
   #include <stdio.h>
   const char far ch[10] = "Hello¥n";

   long long  func1(long long ll1,long long ll2)
   {
       long long   ll3;

       asm( "     pushm A0,A1" );   /* Save A0 and A1 */
       ll3 = ll1 / ll2;
       asm( "     popm  A0,A1" );   /* Recover A0 and A1 */
       return ll3;
   }

   int  main(void)
   {
       long long  ll;
```

```
        int far    *i;

        i = ch;

        ll = func1( 2000, 1000 );

        printf( "%s¥n",i );
}
```

---

5. **Schedule of Fixing the Problem**
   We plan to fix this problem in our next release of the product.
