

RENESAS TOOL NEWS on August 1, 2004: RSO-M3T-CC32R-040801D

## A Note on Using C-Compiler Package M3T-CC32R

Please take note of the following problem in using the M3T-CC32R C-Compiler Package for the M32R family MCUs:

- On reading out data in memory using pointers
- 

### 1. Versions Concerned

M3T-CC32R V.1.00 Release 1 through V.4.20 Release 1A

### 2. Description

When several items of data in different addresses in memory are read using a pointer, one or more of the addresses may be accessed incorrectly.

#### 2.1 Conditions

This problem occurs if the following four conditions are satisfied. However, depending on how the registers are used for the processing in the function that contains the statements meeting all the conditions, this problem may not arise.

- (1) Any of the optimizing options -O7, -O5, -O3, and -O1, or -Ospace only or -Otime only is selected.
- (2) In the program exists a pointer pointing to an integer not qualified as volatile.
- (3) In the program exists a series of processing including both of the following, (a) and (b):
  - (a) At least one practice of increasing or decreasing the integer pointed to by the pointer in (2) by a constant value
  - (b) At least two practices of indirect references using the pointer in (2); or indirect references using the

pointer and a constant; or references to subscripts of an array using the pointer

(4) The series of processing in (3) includes none of the following:

- (a) branches
- (b) writes to memory using pointers
- (c) function calls

## 2.2 Example

Note that in this example the problem may not arise depending on the contents of the omitted portions in the program.

```
-----  
void func(unsigned char *ptr_top, unsigned int width)  
{  
    unsigned int x,y;  
    unsigned int a,b,c;  
    unsigned char *ptr;    /* Condition (2) */  
  
    for(y=0; y<width; y++){  
        ptr = ptr_top + y * width;  
  
        .....  
  
        for(x=0 ; x<64 ; x+=2){  
            a = *ptr++;    /* Conditions (3)-(a), -(b), and (4) */  
            b = *ptr++;    /* Conditions (3)-(a), -(b), and (4) */  
            c = *ptr++;    /* Conditions (3)-(a), -(b), and (4) */  
  
            .....  
  
            ptr -= 3;  
            *ptr++ = (a << 4) | (b >> 4);  
  
            *ptr++ = (a << 4) | (c >> 4);  
            *ptr++ = c << 4;  
        }  
    }  
}
```

-----

### 3. Workaround

This problem can be circumvented in any of the following ways:

- (1) If you are using -O7, -O5, -O3, or -O1, change it to any of these, -O6, -O4, -O2, and -O0.

If you want to use -Ospace or -Otime, use -O6, -O4, -O2, or -O0 at the same time.

- (2) Declare the pointer in 2.1-(2) to be volatile.

Circumvention of Condition 2

Original:

-----

.....

unsigned char \*ptr;

.....

-----

Modified:

-----

---

.....

volatile unsigned char \*ptr; /\* Declared to be volatile \*/

.....

-----

---

- (3) Disconnect the practices of increasing or decreasing an integer value pointed to by a pointer and reads out memory using pointers.

Circumvention of Condition 4

Original:

```
-----  
void func( )  
  
.....  
  
for(x=0 ; x<64 ; x+=2){  
    a = *ptr++;  
    b = *ptr++;  
    c = *ptr++;  
  
.....
```

```
-----
```

ModifiedF

```
-----
```

---

```
static void dummy(void) { return; } /* Dummy function  
declared */
```

```
void func( )
```

```
.....
```

```
for(x=0 ; x<64 ; x+=2){  
    a = *ptr;      /* Value pointed to by a pointer  
                   not increased or decreased */  
    b = *(ptr+1);  
  
    c = *(ptr+2);  
    dummy();      /* Dummy function inserted */  
    ptr += 3;     /* Here values pointed to by a pointer  
                 increased or decreased at a time */
```

```
.....
```

```
-----
```

---

#### 4. 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.