

Note on Using C Compiler Package for R32C/100 Series

When using the C compiler package for the R32C/100 series of MCUs, take note of the following problem:

- With using array-type variable as divisor of remainder expression
-

1. Product and Versions Concerned

C Compiler Package for R32C/100 Series
V.1.01 Release 00 through V.1.02 Release 01A

2. Description

If you use a variable of type array as the divisor of a remainder expression, incorrect code is generated.

2.1 Conditions

This problem arises if the following conditions are all satisfied:

- (1) The dividend of a remainder expression is a variable. (NOTE 1.)
- (2) The divisor of a remainder expression is a variable of type array. (NOTE 1.)
- (3) The subscript of the array in (2) is an auto variable.
(Notes 1 and 2.)
- (4) The variables in (1) and (2) are not declared with `#pragma EXTMEM`.
- (5) The type of each variable in (1), (2), and (3) above is any of the following:
 - unsigned char
 - signed char
 - unsigned short
 - signed short
 - unsigned int (NOTE 3.)
 - signed int (NOTE 3.)
- (6) The bit widths of the variables in (1), (2), and (3) above are the same.

NOTES:

1. Excluded is the case where a variable is replaced with a constant as a result of the compiler's constant propagation.
2. If the array is represented by a pointer, the offset is an auto variable.
3. Compile option `-fint_16(-fI16)` is selected at the same time.

2.2 Example

Command line: `nc100 -S sample.c`

Problem-arising source code:

```
-----
int i;
unsigned char Dividend;      // Conditions (1), (5), and (6)
unsigned char Divisor[];    // Conditions (2), (5), and (6)

void func(void)
{
    unsigned char Index = i;  // Conditions (3), (5), and (6)

    i = Dividend % Divisor[Index]; // Conditions (1), (2), and (3)
}
-----
```

Symptom: The subscript takes the same value as of the dividend.

Generated code:

```
-----
_func:
    .line 7
;## # C_SRC :    unsigned char Index = i;
    mov.b  _i:16,R0L    ; Index
    .line 9
;## # C_SRC :    i = Dividend % Divisor[Index];
    extz.bw _Dividend:16,R0
    .inxlb R0L          <--- Subscript takes same value
                        as of dividend.
    edivu.b _Divisor:16,R0
    extz.bl R0H,_i:16
    .line 10
;## # C_SRC :    }
    rts
-----
```

3. Workaround

To avoid this problem, assign the divisor to a temp variable; then use the temp variable as the divisor.

Example:

```
int i;
unsigned char Dividend;
unsigned char Divisor[6];

void func(void)
{
    unsigned char Index = i;
    unsigned char temp = Divisor[Index]; // Divisor assigned to temp.

    i = Dividend % temp;                // temp used as divisor.
}
```

4. Schedule of Fixing Problem

Sorry we have no plan to fix this problem.

[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.