RENESAS Tool News

RENESAS TOOL NEWS on October 1, 2005: RSO-H8C-051001D

A Note on Using the C/C++ Compiler Package V.6 for the H8SX, H8S, and H8 Families

Please take note of the following problem in using the C/C++ compiler package V.6 for the H8SX, H8S, and H8 families:

- On accessing an incorrect addresses if a structure nested in another has members of a structure-type array (H8C-0026)

---

1. Versions Concerned

The C/C++ compiler package for the H8SX, H8S, and H8 families
V.6.00 Release 00 through V.6.00 Release 03,
V.6.01 Release 00, and V.6.01 Release 01

2. Description

If a structure is declared to be nested in another structure, and the structure-type variables of the former are declared to be of array type, incorrect addresses may be accessed.

2.1 Conditions

This problem occurs if the following conditions are all satisfied:

(1) The cpu=h8sxn, h8sxm, h8sxa, h8sxx, or ae5 option is used. And also the cpu=2000n, 2600n, 2000a, or 2600a option is used if the legacy=v4 option is not selected in the compiler package V.6.01 Release 00 or later.

(2) A structure and its structure-type variables are declared.

(3) Structures nested in the structure in (2) in two or more levels are declared, and the structure-type variables of the structure in the deepest nesting

level are declared to be an array type.

(4) The structures nested in (3) is not declared to be the first member of the structure in (2).

(5) Dot operators are used for referencing or defining structure-type variables.

2.2 Example

```
---------------------------------------------------------
struct {
   int data;         // Condition (4)
   struct {          // Condition (3); 1st nesting
      struct {       // Condition (3); 2nd nesting
         int a;
         int b;
      }x[2];         // Condition (3); Structure-type
                     //    array nested in deepest level
   }y;
}z;                  // Condition (2)
int v;

void func(int offset){

   v = z.y.x[offset].a;    // Condition (5)
}
---------------------------------------------------------
```

3. Workaround

This problem can be circumvented either of the following ways:

(1) Declare the first structure nested in Condition (3) to be the first member of the structure in Condition (2).

Example:

```
------------------------------------------------------
---
struct {
   struct {           // As 1st member of a structure,
                                  use another
      struct {
         int a;
         int b;
```

```
        }x[2];
    }y;
    int data;
}z;
int v;

void func(int offset){
    v = z.y.x[offset].a;
}
-------------------------------------------------------
---
```

(2) Use a pointer to access a structure-type variable.

Example:

```
-------------------------------------------------------
---
struct str{
    int data;
    struct {
        struct {
            int a;
            int b;
        }x[2];
    }y;
}z;
int v;

void func(int offset){
    struct str *p = &z;   // Declare pointer-type variable

    v = p  y.x[offset].a; // Access variable using pointer
}
-------------------------------------------------------
---
```

4. Schedule of Fixing the Problem

We will fix this problem in the next release of the product (in the first quarter of 2006).

[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.