[Notes]

C/C++ Compiler Package for SuperH RISC engine Family

R20TS0306EJ0100
Rev.1.00
Sep. 01, 2018

Outline

When using the C/C++ compiler package for the SuperH RISC engine family, note the following point.

1. A value that is not a constant expression is written as the initial value for static variables or aggregate-type/union-type automatic variables
(SHC-0099)

Note: The above number following the description is an identification number for the note.

1. A Value That is not a Constant Expression is written as the Initial Value for Static Variables or Aggregate-Type/Union-Type Automatic Variables (SHC-0099)

1.1 Applicable Products

C/C++ Compiler Package for SuperH RISC engine Family

(We have confirmed that the defect is present in V5.1A and later versions.)

1.2 Details

If the initializer of a static variable or aggregate-type/union-type automatic variable is not a constant expression, a compilation error may not be generated, and an incorrect initial value may be set, even if the expression violates the ANSI Standard (C89 standard).

1.3 Conditions

This problem may arise if all of the following conditions are met.

(1) The program is compiled as a C program.

(2) Any of the following definitions is present:

(2-1) A static variable having an initial value

(2-2) An aggregate-type automatic variable having an initial value

(2-3) A union-type automatic variable having an initial value and its first member to be initialized being of an aggregate type

(3) The initial value of (2) above has been specified as the address of an array member of a structure- or union-type variable.

(4) The definition of the array member of the structure- or union-type variable of (3) above is qualified as static.

(5) The array of (4) above satisfies either of the following conditions, (5-1) or (5-2).

(5-1) A static variable rather than a constant expression is used as the subscript expression of the array of (4) above.

(5-2) The code of the array of (4) above includes an expression that is the result of a conditional operator in which a static variable is used.

1.4 Example

Red texts indicate the parts corresponding to the above conditions.

The -lang=c option is specified /* Condition (1) */

```
1:  struct S {
2:    int m;
3:  } sa[3], sa2[3];
4:  int i;
5:  int *p1 = &sa[i].m;                    /*  Conditions (2-1) (3) (4) (5-1)  */
6:  int *p2 = &(i?sa:sa2)[0].m;            /*  Conditions (2-1) (3)  (4)  (5-2)  */
7:  void func1(void)
8:  {
9:    int *p3[2] = {&sa[i+1].m,            /*  Conditions (2-2) (3)  (4)  (5-1)  */
10:             &(i>3?sa:sa2)[1].m};       /*  Conditions (2-2) (3)  (4)  (5-2)  */
11: }
12: void func2(void)
13: {
14:    union {
15:      struct {
16:        int *mem1_1;
17:      } mem1;
18:     int *mem2;
19:   } uni1 = {&sa[i].m};                  /*  Conditions (2-3) (3)  (4)  (5-2)  */
20:
21:   p1 = uni1.mem1.mem1_1;
22: }
```

1.5 Workaround

If there is a variable having an initial value to which a condition is applied, do not set the initial value for the variable through initialization but set it during execution. Then, this problem can be avoided.

```
1:  struct S {
2:      int m;
3:  } sa[3], sa2[2];
4:  int i;
5:  int *p1;          /* Workaround */
6:  int *p2;          /* Workaround */
7:
8:  void func1(void)
9:  {
10:     int *p3[2];      /* Workaround */
11:     p1 = &sa[i].m;      /* Workaround */
12:     p2 = &(i?sa:sa2)[0].m;      /* Workaround */
13:     p3[0] = &sa[i+1].m;       /* Workaround */
14:     p3[1] = &(i>3?sa:sa2)[1].m;      /* Workaround */
15: }
16: void func2(void)
17: {
18:    union {
19:      struct {
20:        int *mem1_1;
21:      } mem1;
22:     int *mem2;
23:   } uni1;      /* Workaround */
24:   uni1.mem1.mem1_1 = &sa[i].m;  /* Workaround */
25:
26:   p1 = uni1.mem1.mem1_1;
27: }
```

1.6 Schedule for Fixing the Problem

There is no update scheduled to fix this issue.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Sep. 01, 2018	-	First edition issued

TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061 Japan
Renesas Electronics Corporation

■Inquiry
https://www.renesas.com/contact/