

Notes on Using the C/C++ Compiler Package V.9 for the SuperH RISC Engine Family of MCUs

Please take note of the following problems in using the C/C++ compiler package V.9 for the SuperH RISC engine family:

- On calling a function that takes arguments passed via the stack (SHC-0050)
- On using a union that contains two or more members of the same structure or union type (SHC-0051)

1. Versions Concerned

The C/C++ compiler package for the SuperH RISC engine family
V.9.00 Release 00 through V.9.00 Release 03

2. Descriptions

- 2.1 On Calling a Function That Takes Arguments Passed via the Stack (SHC-0050)
If an interrupt is generated after a function that takes arguments passed via the stack is called, the value of the stack pointer pointing to the beginning of the interrupt routine may become an odd number.

Conditions:

This problem may occur if the following conditions are all satisfied:

1. The optimize=1 option is used.
2. A call is made to a function that takes arguments passed via the stack.
3. The total size of the arguments in Condition 2 is 124 bytes or more and 252 bytes or less.

Example:

```
-----  
typedef struct {  
    char a[128];  
} ST;  
extern void f(ST fc);
```

```

ST a;
void func() {
    f(a);    // Conditions 2 and 3
}

```

Result of compilation:

```

_func:
    STS.L    PR,@-R15
    ADD     #-128,R15
    MOV.L   L11,R2 ; _a
    MOV.L   L11+4,R6 ; __slow_mvn
    MOV     #-128,R0 ; H'FFFFFF80
    MOV     R15,R1
    JSR     @R6
    EXTU.B  R0,R0
    MOV.L   L11+8,R1 ; _f
    JSR     @R1
    NOP
    ADD     #127,R15 ; Value of stack pointer becomes odd
    ADD     #1,R15 ; If an interrupt generated immediately
                ; before this, value of stack pointer
                ; pointing to beginning of interrupt
                ; routine becomes odd
    LDS.L   @R15+,PR
    RTS
    NOP

```

Workaround:

Use the optimize=0 option; not optimize=1.

2.2 On Using a Union That Contains Two or More Members of the Same Structure or Union Type (SHC-0051)

If a union that contains two or more members of the same structure or union type is used, the offset values of union members from the top of the union may become incorrect.

Conditions:

This problem may occur if the following conditions are all satisfied:

1. A union is used.
2. The union in Condition 1 contains a structure- or union-type member.
3. The union in Condition 1 contains other members of the same structure or

union type as the member in Condition 2.

4. The offset values of the members in Conditions 2 and 3 from the top of the union are the same.
5. In the members in Conditions 2 and 3, members declared at the second or later are referenced or defined.

Example

```
-----  
typedef struct {  
    unsigned int x;  
} ST;  
typedef union {    // Condition 1  
    struct {  
        unsigned char a;  
        ST s2[1];    // Conditions 2 and 4  
    } st1;  
    struct {  
        unsigned char c;  
        ST s3;    // Conditions 3 and 4  
    } st2;  
} UN;  
void func() {  
    volatile int a=0;  
    UN u;  
    f(u.st2.s3.x);    // Condition 5  
}
```

Result of compilation:

```
_func:  
    STS.L    PR,@-R15  
    ADD     #-12,R15  
    MOV.L   L11+2,R2    ; _f  
    MOV.L   @(4,R15),R4 ; Must be MOV.L @(8,R15),R4  
    MOV     #0,R3  
    JSR    @R2  
    MOV.L   R3,@R15  
    ADD     #12,R15  
    LDS.L   @R15+,PR  
    RTS  
    NOP
```

Workaround:

Define structures or unions that have names different from the structure or union in Condition 2 above and contain the members of the same names as its members. Then declare the members in Condition 3 to have the type of the structures or unions having different names.

Example:

```
-----  
typedef struct {  
    unsigned int x;  
} ST;  
typedef struct {  
    unsigned int x;  
} ST1;  
typedef union {  
    struct {  
        unsigned char a;  
        ST s2[1];  
    } st1;  
    struct {  
        unsigned char c;  
        ST1 s3;  
    } st2;  
} UN;  
-----
```

3. Schedule of Fixing the Problems

We plan to fix these problems in the next release of the product (scheduled 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.