[Notes]

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

R20TS0102EJ0100
Rev.1.00
Dec. 16, 2016

Outline

When using the C/C++ Compiler Package V.9 for the SuperH RISC engine family, note the following point:

1. Using the "strcmp()" standard C library function (SHC-0096)

   Note: The number which follows the description of the precautionary note is an identifying number for the precaution.

1. Using the "strcmp()" Standard C Library Function (SHC-0096)

1.1 Applicable Products

C/C++ Compiler Package for SuperH RISC engine family V.9.01 Release 00 to V.9.04 Release 03

1.2 Details

When using the "strcmp()" standard C library function in a loop (*1) or interrupt function, the register(*2) value might not be retained.

*1: The problem when "strcmp()" is used in a loop will occur in V.9.02 Release 00 to V.9.04 Release 03.

*2: In V.9.01 Release 00 to V.9.03 Release 02, the value of the R1, R2, R3, and R6 registers might not be retained.
In V.9.04 Release 00 to V9.04 Release 03, the value of the R2 and R3 registers might not be retained.

1.3 Conditions

This problem might occur if either of the following conditions (1) and (2) is met:

(1) All of the Conditions (1a) to (1d) are satisfied.

(1a) C/C++ Compiler Package for SuperH RISC engine family V9.02 Release 00 to V9.04 Release 03 is used.

(1b) Neither the "-optimize=0" nor the "-optimize=debug_only" option is used.

(1c) The "strcmp()" standard C library function is used in a loop, and the function name is not enclosed in brackets.

(1d) The user defined function is not called before "strcmp()" in a loop that corresponds to (1c).

Note that this is not applied when the user defined function that is not expanded in-line is called.

(2) Both of the Conditions (2a) and (2b) are satisfied.

(2a) "strcmp()" is used in a function which contains "#pragma interrupt", and the function name is not enclosed in brackets.

(2b) The user defined function is not called in a function that corresponds to (2a).

Note that this is not applied when the user defined function that is not expanded in-line is called.

1.4 Example

Parts that correspond to the conditions are described in red.

(1) In the case of Condition (1): When using V9.02 Release 00 to V9.03 Release 02 (Condition (1a))

- Source example: "-optimize=1" is used (Condition (1b))

```
#include <string.h>
: :
void func(char *c1, char *c2, int *ip1, int i1, int i2, int i3){
  int lc_i1, lc_i2, lc_i3, lc_i4;
: :
  for(lc_i2= 0; lc_i2< 100; lc_i2++){
  /* There is no description to call the user defined function */
  /* here. (Condition (1d)) */
    if (strcmp(tbl[lc_i2].a, cp1)==0) {/* (Condition (1c)) */
: :
    }
: :
  }
: :
}
```

- Result of compilation

```
: :
MOV.L L15+12,R1  ; H'FFE7FFFF Set R1 here.(1)
MOV.L L15+16,R10 ; __slow_strcmp1
AND R1,R2
LDS R2,FPSCR
L11:
MOV.L @R13,R4
JSR @R10 ; Call ”strcmp()”.Change R0 to R6 here.
         ; -> Change R1 that should hold the (1) value.
MOV R11,R5
STS FPSCR,R2
TST R0,R0
AND R1,R2 ; Assume that the R1 value has been set at (1) and use R1.
BF/S L13
LDS R2,FPSCR
: :
```

(2) In the case of Condition (2)

- Source example:

```
#include <string.h>
#pragma interrupt(func)
int fl;
char str1[100], str2[100];

void func(){
  /* There is no description to call the user defined function */
  /* in the function. (Condition (2b)) */
  fl = strcmp(str1,str2); /* (Condition (2a)) */
}
```

- Result of compilation

```
: :
MOV.L R0,@-R15
MOV.L R2,@-R15
MOV.L R4 @-R15
MOV.L R5,@-R15
MOV.L R6,@-R15
: :
MOV.L L11+2,R2 ; __slow_strcmp1
               ; Call ”strcmp()”.
               ; Change R0 to R6 (when using V.9.02 Release 00 to
               ; V.9.03 Release 02) or R0, R2 to R5 (when using
               ; V.9.04 Release 00 to V.9.04 Release 03) here.
: :
JSR @R2
: :
MOV.L @R15+,R6
MOV.L @R15+,R5
MOV.L @R15+,R4
MOV.L @R15+,R2
MOV.L @R15+,R0
RTE             ;Change R3 and R1 (when using V.9.02 Release 00 to V.9.03
                ; Release 02) or R3 (when using V.9.04 Release 00 to V.9.04
                ; Release 03) and terminate the interrupt processing.
```

1.5 Workarounds

To avoid this problem, take any of the following steps:

(1) Enclose the function name of "strcmp()" in brackets.

(2) Define the following function in a separate file and substitute it for "strcmp()". Also, do not specify any file that contain functions defined with the "-file_inline" option.

```
#include <string.h>
int strcmp2(const char *s1, const char *s2)
/* Function names are arbitrary. */
{
return (strcmp(s1,s2));
}
```

(3) Use either of the "-optimize=0" and the "-optimize=debug_only" options. (*)

*: This workaround is only effective for cases corresponding to Condition (1)

1.6 Schedule for Fixing the Problem

This problem will be fixed in the next version. The release date has not yet been decided.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Dec. 16, 2016	-	First edition issued

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

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

Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.

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.

All trademarks and registered trademarks are the property of their respective owners.