To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Date: Oct.27.2004

RENESAS TECHNICAL UPD

Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan
RenesasTechnology Corp.

Product Category	User Development Environment		Document No.	TN-CSX-A084A/E	Rev.	1.0
Title	SuperH RISC engine C/C++ Compiler Ver.8 bug information(4)		Information Category	Technical Notification		
Applicable Product	R0C40700XSW08R (P0700CAS8-MWR) R0C40700XSS08R (P0700CAS8-SLR) R0C40700XSH08R (P0700CAS8-H7R)	Lot No. Ver.8.0	Reference Document	SuperH RISC engine C/C++ Compiler Assembler Optimizing Linkage Editor User's Manual (REJ10B0047-0100H Rev.1.00)		

Problem with the SuperH RISC engine C/C++ compiler V.8 is listed below.

Please be careful to use this version of the compiler.

This bug is fixed on V.8.00.03.

1) Incorrect replacement of loop induction variable (SHC-0003)

[Description]

When loop induction variables existed and their type differs others in a loop, they might be commonized incorrectly.

[Example]

```
extern void g();
void func(unsigned int x) {
    unsigned long i=3;
    signed long k=3;

    while (i<x) {
        if (k<-3) {   /* variable k was replaced illegally by variable i. */
            break;
        }
        g();
        --i;
        --k;
    }
}
```

[Conditions]

This problem might occur when all of the following conditions were fulfilled.

(1) The optimize=1 option was specified.

(2) A loop existed.

(3) The loop of (2) had a signed int type or signed long type loop induction variable and an unsigned int type or unsigned long type one.

(4) Initial values of the loop induction variables of (3) were constant value.

(5) Updating values of the loop induction variables of (3) were the same value.

[Solution]

This problem can be prevented by either of the following methods.

(1) Specify optimize=0.

(2) Declare either of the loop induction variables of (3) as volatile.

(3) Declare either of the loop induction variables of (3) as char/unsigned char/short/ unsigned short type variable.

(4) Declare the loop induction variables of (3) as the same type variables.