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-A085A/E	Rev.	1.0
Title	SuperH RISC engine C/C++ compiler package V.8.00 Release 03 Updates	Information Category	Technical Notification		
Applicable Product	R0C40700XSW08R (P0700CAS8-MWR) R0C40700XSS08R (P0700CAS8-SLR) R0C40700XSH08R (P0700CAS8-H7R)	Lot No. All	Reference Document	SuperH RISC engine C/C++ Compiler Assembler Optimizing Linkage Editor User's Manual REJ10B0047-0100H Rev.1.00	

SuperH RISC engine C/C++ compiler package is updated in V.8.00 Release 03.

See the notes below if you have the compiler package listed in the following table.

Part No.	Package version	Compiler version
P0700CAS8-MWR	8.0.00 8.0.01	8.0.00 8.0.01
R0C40700XSW08R	8.00 Release 01 8.00 Release 02	8.0.01 8.00.02
P0700CAS8-SLR	8.0.00 8.0.01	8.0.00 8.0.01
R0C40700XSS08R	8.00 Release 01	8.0.01
P0700CAS8-H7R	8.0.00 8.0.01	8.0.00 8.0.01
R0C40700XSH08R	8.00 Release 01	8.0.01

If you have the compiler package of the Windows® version, download the update program from the following URL:

http://www.renesas.com/eng/products/mpumcu/tool/index.html

If you have the compiler package of the UNIX version, request the update program to an authorized product distributor.

The contents of updates in this package are shown below.

Descriptions of section 1 only apply to the Windows® version.

1. High-performance Embedded Workshop (Windows® version)

1.1 Release of the restriction of HEW V.3.01.02

We have corrected the following problems:

(1) Restriction of MAP Display

The "Linker section setting information" on a MAP window isn't displayed.

(2) Restrictions of Virtual Desktop function

Don't open memory window on more than two window configurations of virtual desktop function.

(3) Restriction of the "Include map file:" option

A link error might occur when "Include map file:" was specified on the C/C++ tab of the Tools Options Dialog Box.

(c) 2004. Renesas Technology Corp., All rights reserved.

2. Compiler

2.1 The new options

The following shows the options added to V.8.00.03.

```
  - bss_order={declaration|definition}
  - stuff[={bss|data|const}[,...]]
    nostuff
```

2.2 Illegal Calculation of Quadratic Expression of Loop Induction Variable

We have fixed the problem of incorrectly calculating a quadratic expression when loop induction variable was in this expression.

[Example]

```
int a[100];
void f() {
    int i;
    for (i=0;i<100;i++) {
        a[i] = 3 * (i * i + 555 * i); /* illegally expanded to 3*i*i+555*i */
    }
}
```

[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 int/unsigned int/long/unsigned long-type loop induction variable.

(4) A quadradic expression with the loop induction variable of (3) existed in the loop of (2).

(5) The expression of (4) had the form of "m*(i*i+b*i)" (i:loop induction variable m,b:variable or constant value).

2.3 Incorrect Removing of Sign/Zero Extension of a Constant Division (SHC-0001)

We have fixed the problem of removing the cast when a divisor and a dividend were cast to the type of smaller size at a constant division and the result of the division was assigned to a variable with a type after the cast.

[Example]

```
char c;
int   i;
void func1() {
    c = ((char)i / (char)2);       /* a dividend was not cast to char-type */
}
void func2(){
    c = ((char)i / (char)0x102); /* a divisor was not changed into 0x2 */
}
```

[Conditions]

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

(1) The optimize=1 option was specified.

(2) A constant division existed.

(3) A divisor and a dividend were cast to the type of smaller size at a constant division of (2).

(4) Other than cpu=sh1 option and the division=cpu=inline option were specified, or the divisor was a power of 2.

(5) The result of the division was assigned to a variable with a type after the cast.

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

We have fixed the problem of incorrectly commonizing loop induction variables when these variables existed and their type differed others in a loop.

[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.

3. Optimizing Linkage Editor

3.1 Incorrect object due to specifying of delete unreferenced symbol optimization

The following problem were fixed.

When delete unreferenced symbol optimization was specified, a literal without the symbol name was deleted illegally.

[Conditions]

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

(1) Object files were compiled with the goptimize option.

(2) A const variable which was not referred was in C source program.

(3) The string literal was used in C source program of (2).

(4) Some relocatable file including the object file of (2) was generated.

(5) The delete unreferenced symbol optimization (optimize=symbol_delete) was valid when the relocatable file of (4) was input.

3.2 Fixed internal error (L4001) with the strip option

The following problem with the linker version 8.00.04 was fixed.

An internal error (L4001) occurred when the strip option specified.