HITACHI SEMICONDUCTOR TECHNICAL UPDATE

Classification of Production	Development Environment		No	TN-CSX-046A/E	Rev	1
THEME	SuperH RISC engine C/C++ Compiler Ver.6 Failure Found in This Release (When CPU=SH4 is Specified.)	Classification of Information	1. Spec change 2. Supplement of Documents (3.) Limitation of Use 4. Change of Mask 5. Change of Production Line			
PRODUCT NAME	P0700CAS6-MWR P0700CAS6-SLR P0700CAS6-H7R	Lot No. All	Reference Documents	SuperH RISC engine C/C++ Compiler Assembler Optimizing Linkage Editor User's Manual ADE-702-246A Rev.1.0	Effective Date Eternity	

Attached is the description of the known bugs in Ver. 6 series of the SuperH RISC engine C/C++ compiler. Inform the customers who have the package version in the table below of the bugs.

	Package version	Compiler version
P0700CAS6-MWR	6.0	6.0
	6.0R1	6.0
	6.0A	6.0A
	6.0AR1	6.0A
	6.0AR2	6.0A
	6.0B	6.0B
	6.0C	6.0C
P0700CAS6-SLR	6.0	6.0
	6.0A	6.0A
	6.0AR1	6.0AR1
	6.0B	6.0B
	6.0C	6.0C
P0700CAS6-H7R	6.0	6.0
	6.0A	6.0A
	6.0AR1	6.0A
	6.0B	6.0B
	6.0C	6.0C

Attached: P0700CAS7-021118E
SuperH RISC engine C/C++ Compiler Ver. 6
Failure Found in This Release (When CPU=SH4 is Specified)

SuperH RISC engine C/C++ Compiler ver. 6
Failure Found in This Release (When CPU = SH4 is Specified)

The failure found in the ver. 6 series of the SuperH RISC engine C/C++ compiler is as follows:

1. Incorrect floating-point operations

[Description]
If an operation includes a compound assignment expression of (unsigned variable) op = (double-type variable) (op: an operator for addition, subtraction, multiplication, or division) when CPU = SH4 is specified, the operation may not be performed correctly.

[Example]
<C source program>

```
#include <stdio.h>
unsigned int a=2, c;
double b=3;

void main()
{
    c=b;
    a*=b;
    printf("a=%d\n", a); /* a is not 6 */
}
```

<Assembly source program>

```
      STS         FPSCR,R3
      MOV.L       L282+4,R2
      OR          R2,R3
      LDS         R3,FPSCR   ; H'00080000 The precision of an operation is changed to the double type
                             ; (FPSCR.PR = 1).
       :
      MOV.L       R3,@R0
       :                     ; Coding expansion of a* = b
      MOV.L       @R2,R0
      MOV.L       L282+20,R1 ; __u2d
      JSR         @R1
      NOP
      FMUL        DR2,DR0    ; A double-type multiplication cannot be performed correctly because
                             ; the precision of an operation has been set as the single type by __u2d.
      MOV.L       L282+24,R1 ; __d2u
      JSR         @R1
      NOP
      MOV.L       R0,@R2
```

[Conditions]

This problem may occur when all of the following conditions are satisfied.

(1) cpu =sh4 and fpscr = aggressive (default) are specified, not fpu = {single | double}.

(2) An operation includes a compound assignment expression of data1 op = data2 (op: an operator for addition, subtraction, multiplication, or division).
Type of data1: unsigned char, unsigned short, unsigned int, or unsigned long
Type of data2: double

(3) A double-type operation (including a type conversion) is included in the same function where the expression of condition (2) is placed.

[Method of Checking]

Check if a relevant failure exists in the program by the following method.

(1) Output the assembly source program or a listing file to check if the function __u2d or __d2u is called.

[Solution]

If a relevant part is found, prevent the problem by the following method.

(1) Convert a compound assignment expression to a simple assignment expression.
<Example> `a* = b; -> a = a*b;`