

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>.

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-074A/EA	Rev.	1.0
Title	H8S, H8/300 Series C/C++ Compiler Ver.6.0.01 bug information (2)		Information Category	Usage Limitation		
Applicable Product	PS008CAS6-MWR(R0C40008XSW06R) PS008CAS6-SLR(R0C40008XSS06R) PS008CAS6-H7R(R0C40008XSH06R)	Lot No.	Reference Document	H8S, H8/300 Series C/C++ Compiler Assembler Optimizing Linkage Editor User's Manual REJ10B0058-0100H Rev.1.00		
		Ver.6.0.00 Ver.6.0.01				

Attached is the description of the detected bug information in Ver.6.0.01 of the H8S, H8/300 Series C/C++ Compiler.

The bug will affect this package version.

Attached: R0C40008XSW06R-040514E

H8S, H8/300 Series C/C++ Compiler Ver. 6.0.01 The details of the detected bug information (2)

H8S, H8/300 Series C/C++ Compiler Ver.6.0.01

The details of the detected bug information (2)

Problems with the H8S, H8/300 series C/C++ compiler ver.6.0.01 are listed below.

1) Incorrect accessing to a member of array type in a structure

If an array type variable declared as not a first member in a structure and an expression has a reference of the address of array type member, the address value might not be correctly.

[Example]

```
struct S {
    int b;
    int a[8];
} s;
int *p;

void test() {
    p = &s.a[7];
}
```

<pre>< Incorrect > _test: mov.l #_s+h'e:32,@_p:32 ; it points &s.a[6] rts < Correct > _test: mov.l #_s+h'10:32,@_p: 32 ; it points &s.a[7] rts</pre>

[Conditions]

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

- a) H8SXN, H8SXM, H8SXA, or H8SXX was specified as a CPU type.
- b) An array type variable was declared as not a first member in a structure.
- c) An expression is described as accessing of an array type member address except first element.

[Solution]

This problem can be prevented by the following method:

- i. Add a volatile qualifier to pointer type variable definition.

First of array type member to get an address is assigned to the pointer type variable.

And then your program should access to an array element,

```
struct S {
    int b;
    int a[8];
} s;
int *volatile p;

void test() {
    p = &s.a;          /* get and address top of array type member */
    p += 7;          /* move to the element */
}
```

[Applied Product Version]

Ver.6.0.00 or later

2) Incorrect optimization for a pre/post increment/decrement instruction

If your code has either pre-increment, pre-decrement, post-increment, or post-decrement expression with speed option,
The value might not be correct as a result.

[Example]

```
void func(char *par1, char *par2) {
    *--par1 = *--par2;
    *--par1 = *--par2;
    *--par1 = *--par2;
    *--par1 = *--par2;
}
```

<Correct Code>

```
ADD.L    #10:16,ER0
MOV.B    @-ER0,R1L
MOV.L    ER0,ER4

MOV.L    SP,ER2
ADD.L    #16:16,ER2
MOV.B    R1L,@-ER2
```

<Incorrect Code>

```
ADD.L    #10:16, ER0
; (Deleted illegally) ... (a)
MOV.L    ER0, ER4    ... (a)was deleted then incorrect value is
                    substituted.

MOV.L    SP, ER2
ADD.L    #16:16, ER2
MOV.B    @-ER0,@-ER2    ... (a)is merged incorrectly.
```

[Conditions]

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

- a) H8SXN, H8SXM, H8SXA, or H8SXX was specified as a CPU type.
- b) Speed option was specified.
- c) One of the following expression was in your source file
 - i. Pre-increment
 - ii. Pre-decrement
 - iii. Post-increment
 - iv. Post-decrement

[Solution]

This problem can be prevented by the following method:

- Add a volatile qualifier to the variable definition.

[Example]

```
void func(volatile char *par1, volatile char *par2){
    *--par1 = *--par2;
    *--par1 = *--par2;
    *--par1 = *--par2;
    *--par1 = *--par2;
}
```

[Applied Product Version]

Ver.6.0.00 or later

3) Illegal register allocation of table expansion optimization for switch-case statement

If switch-case statement is expanded to the table optimization, register push/pop instruction might not be correct.

[Example]

```
void main(void) {
    val=c_1;
    ans2=gal;
    ans2 += val;          /* ans2 is pushed to stack          */
    switch(val){
    case 2: ans=0; break;
        :
    default: ans=0; break;
    }                    /* ans2 might not be popped from stack          */
    ...
}
```

[Conditions]

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

- a) H8SXN, H8SXM, H8SXA, or H8SXX was specified as a CPU type.
- b) The case = table or case = auto option was specified and switch-case statement was expanded as table expansion optimization.

[Solution]

This problem can be prevented by the following method:

- i. The case=Ifthen option should be specified.

[Applied Product Version]

Ver.6.0.00 or later

- 4) The instruction, which assigned 0, is deleted illegally.
If an instruction of assigned 0 was in each branch, it might be deleted illegally.

[Example]

```
sub.b    R0H, R0H    ;R0H was set as 0
:
:
L55:
:
bls      L36
sub.b    R0H, R0H    ; R0H was still 0 then this instruction can be deleted by optimization
add.w    R0, R0      ; R0H is possible to be updated to other value.
:
L48:
sub.b    R0H, R0H    ; deleted Illegally
add.w    R0, R0      ; If R0H isn't 0, it might be incorrect value
```

[Conditions]

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

- a) H8/300 or H8/300L was specified as a CPU type.
- b) The optimize=1 option was specified(default).
- c) An Instruction of assigned 0 was in each branch.

[Solution]

Take the following method to prevent this problem.

The optimize=0 option should be specified.

[Applied Product Version]

Ver.4.0.04 or later

- 5) A substitute expression was deleted 4 or less bytes of structure variable illegally
If 4 or less bytes structure variable was in a substitute expression, it might be deleted.

[Example]

```
typedef struct {
    char c;
    int i;
} ST;
:
void func()
{
    ST lst1,lst2,lst3;
    ST lst4,lst5,lst6;
    ST lst7,lst8,lst9;
    :
    lst9.c = 1;

    switch(x){
    case 1:
        y+=x;
        switch(y){
        case 3:
            :
            lst9.c = 3;          /* illegal deletion of substitute expression */
            break;
            :
        }
    }
}
```

[Conditions]

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

- a) H8SXN, H8SXM, H8SXA, or H8SXX was specified as a CPU type.
- b) The optimize=1 option was specified(default).
- c) Declared structure with either of following two conditions was fulfilled.
 - i. The member is accessed one byte to top of two bytes from top of structure or two bytes to first byte from top of structure.
 - ii. The STRUCTURE was 2, 3 or 4-byte size, and that have 2 or more members. And at least one of them was bit-field type
- d) c) member accessing was in your program

[Solution]

Take either of the following methods to prevent this problem.

- a) The optimize=0 option should be specified.
- b) Add a volatile qualifier to the variable declaration.

[Applied Product Version]

Ver.6.0.00 or later

6) An incorrect accessing structure member in different accessing

When a structure member set/reference expressions(Indirect reference and Direct reference) were in your program, setting value expression might be deleted.

set/reference method is following.

- i. Indirect (struct->member)
- ii. Direct (struct.member)

[Example]

```
struct tag {
    long e;
}str = {10};

int func() {
    struct tag *p;
    long i,k = 0;

    p = &str;
    for(i=0; i<2; i++){
        str.e = i;                               /* illegal deletion of this expression      */
        k += p->e;                               /* str.e and p->e were pointed to same memory area. */
    }
    return k;
}
```

[Conditions]

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

- a) H8SXN, H8SXM, H8SXA, or H8SXX was specified as a CPU type.
- b) The optimize=1 option was specified(default).
- c) Set/reference expression of the same address of memory was sequential
- d) The expressions of c) consisted of a pointer expression and not a pointer expression.

[Solution]

Take either of the following methods to prevent this problem.

- a) The optimize=0 option should be specified.
- b) Use same access expression to set/reference of value.

[Example]

```
str.e = i;
k += str.e;

or

p->e = i;
k += p->e;
```

[Applied Product Version]

Ver.6.0.00 or later

7) A polynomial in-loop variable results was incorrect.

When a loop has $m*(i*i+b*i)$ polynomial (m and b is constant variable), optimization for loop was incorrect code.

[Example]

```
long a[100];
f() {
    long i;
    for (i=0;i<100;i++){
        a[i] = 3 * (i * i + 555 * i);    /* It was treated as 3*i*i+555*i incorrectly.    */
    }
}
```

[Conditions]

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

- a) H8SXN, H8SXM, H8SXA, or H8SXX was specified as a CPU type.
- b) The optimize=1 option was specified(default)
- c) Loop expressions that fulfilled all of the following conditions
 - i. A loop variable was a long or an unsigned long type.
 - ii. It is in a part of quadratic expression.
 - iii. $m*(i*i+b*i)$ quadratic expression

[Solution]

Take either of the following methods to prevent this problem.

- a) The optimize=0 should be specified
- b) A loop variable should be defined with volatile qualifier.
- c) A loop variable should be defined except long/unsigned long type.
- d) A polynomial expression should be expanded to $((m*i*i) + m*b*i)$ in your program.

[Applied Product Version]

Ver.6.0.00 or later