RENESAS Tool News

A Note on Using the C/C++ Compiler Package V.4 through V.6 for the H8SX, H8S, and H8 Families of MCUs

Please take note of the following problem in using the C/C++ compiler package V.4 through V.6 for the H8, H8S, and H8SX families:

- On assignment expressions before and after an iteration or conditional branch block (H8C-0027)

1. **Versions Concerned**

   C/C++ Compiler Package for the H8, H8S and H8SX family:
   V.4.0 through V.4.0.09,
   V.5.0 through V.5.0.06, and
   V.6.00 Release 00 through V.6.01 Release 01

   Product Types:

	V.4	**V.5**	**V.6**
For Windows	PS008CAS4-MWR	PS008CAS5-MWR	R0C40008XSW06R
For Solaris	PS008CAS4-SLR	-	R0C40008XSS06R
For HP-UX	PS008CAS4-H7R	-	R0C40008XSH06R

2. **Description**

   If a function call is made within an iteration statement or the TRUE/ FALSE statement in a conditional statement or an iteration statement, and before and after the statement or the block exist two assignment expressions that assign the same constant or variable to other variables, incorrect values may be assigned.

   2.1   Conditions
          This problem occurs if the following conditions are all

satisfied:

(1)   The optimize=1 option is selected in the compiler.

(2)   The goptimize option is also selected in the compiler.

(3)   A function call is made within an iteration statement or the TRUE/FALSE statement in a conditional branch block.

(4)   Before and after the statement or the branch block in (3) exist two assignment expressions that assign the same constant or variable to other variables.

(5)   The optimize=register option is made effective in the optimizing linkage editor.

2.2   Example

```
------------------------------------------------------------
void foo(void){

a = 1;              // Condition (4): Same constant assigned
   for( i = 0; i < 10; i++){
     func();        // Condition (3): Function call made in
                              iteration statement
   }
   b = 1;           // Condition (4): Same constant assigned
}
------------------------------------------------------------
```

3. Workaround

This problem can be circumvented any of the following ways:

(1)   Don't select the goptimize option in the compiler.

(2)   Don't make the optimize=register option effective in the optimizing linkage editor.

(3) Select the optimize=0 option in the compiler. Or to the function satisfying Conditions (3) and (4) (for instance, foo in the above example), apply the #pragma option nooptimize directive.

(4) Select the code=asm option in the compiler.

(5) Place a dummy function call immediately before the iteration statement or the conditional branch block.

Example:
```
------------------------------------------------------------
void dmy(){}

void foo(void){
  a = 1;
  dmy();          // Dummy function call placed before
                             iteration statement

  for( i = 0; i < 10; i++){
    func();
  }
  b = 1;
}
------------------------------------------------------------
```

4. Schedule of Fixing the Problem

We plan to fix this problem in the next release of the product. (Due to be released in the first quarter of 2006.)