RENESAS Tool News

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

Please take note of the following problem in using the C/C++ compiler package for the H8SX, H8S, and H8 families of MCUs:

- On referencing incorrect memory area (H8C-0025)

1. **Product Concerned**
   C/C++ Compiler Package V.6.00 Release 00 through V.6.00 Release 03, and V.6.01 Release 00

2. **Description**
   Addresses within the memory area may not be assigned to registers, resulting in incorrect memory area being referenced.

   2.1 Conditions
   This problem may occur if the following conditions are all satisfied:
   1. Any of the options cpu=h8sxn, h8sxm, h8sxa, and h8sxx is selected.
   2. Also the optimize=1 option is selected.
   3. In the program exists a selection statement that satisfies both of the following conditions:
      (1) The controlling expression of the selection statement contains an array-type or pointer-type variable.
      (2) Within or immediately after the selection statement exists an expression containing the variable in (1). (If the variable is of type array, here it can refers to any other element than the one the variable in (1) refers to.)

Example 1:
```
--------------------------------------------------------
if( a != expression){        /* Condition (1)
              a: a variable of type array or pointer
*/
  An expression containing a    /* Condition (2) */
}
--------------------------------------------------------
```
Example 2:
```
--------------------------------------------------------
if( a != expression){      /* Condition (1)
              a: a variable of type array or pointer
*/
  . . . . . . . . . . . . . . . . . . . . . . .
}
An expression containing a  /* Condition (2) */
--------------------------------------------------------
```

3. Workaround

This problem can be circumvented in any of the following ways:

(1) Select the optimize=0 option.

(2) Place "#pragma option nooptimize" and "#pragma option" before and after the function containing the processing of the selection statement in Condition 3, respectively.

(3) Place nop() immediately before the expression described in Condition 3-(2).

Circumvention of Example 1:
```
-----------------------------------------------------------
  if( a != expression){
    nop();                  /* nop() placed */
    An expression containing a
  }
-----------------------------------------------------------
```
Circumvention of Example 2:
```
-----------------------------------------------------------
```

```
-
    if( a != expression){
     . . . . . . . . . . . . . . . . . . . . . . . .
    }
    nop();                 /* nop()placed */
    An expression containing a
 ----------------------------------------------------------
    -
```

4. **Schedule of Fixing the Problem**

We plan to fix this problem in this C/C++ compiler package V.6.01 Release 01.