RENESAS Tool News

RENESAS TOOL NEWS on October 1, 2014: 141001/tn6

Note on Using CS+ CX Compiler

When using CS+ CX Compiler (for V850E2M core), take note of the following problem:

- With using the .dshw assembler directive (No. 15)
  NOTE: The number at the end of the above item is for indexing the problems in this compiler.

Notification: The product name of CubeSuite+, an integrated development environment from Renesas, has been changed to "CS+."

---

1. Versions Concerned
   CS+ CX Compiler V1.00 through V1.31

2. Description
   When the .dshw assembler directive (NOTE) is used to secure two bytes, the highest-order bit (the sign bit) may be truncated since its value does not fit into the two bytes. Nonetheless, assembly proceeds without the output of error or warning messages.

   NOTE: The .dshw is a directive that secures two bytes, and the value used in initialization is specified by the operand shifted rightwards by one bit.
   It is used to set up the branch tables in the table jump format with the switch instruction.

   When the value of the operand satisfies condition (a) or (b) below, the highest-order bit (the sign bit) is truncated since its value does not fit into the two bytes.
   (a) Between 10000H(65536) and 1FFFFH(131071)
   (b) Between FFFE0000H (-131072) and FFFEFFFFH (-65537)

   If the .dshw satisfies the condition above and it is used to set up the branch tables in the table jump format with the switch instruction, the table jumps are incorrect.

For example, when a C source file includes a switch statement, code in the table jump format that uses the switch instruction and .dshw directive may be generated as shown below.
If the .dshw directive satisfies the condition, the branch table that is generated will be incorrect.

Example: assembly source

```
------------------------------------------------
switch      r10
.SWITCH_LABEL.0:
.dshw       .BB.LABEL.0 - .SWITCH_LABEL.0
.dshw       .BB.LABEL.1 - .SWITCH_LABEL.0
.dshw       .BB.LABEL.2 - .SWITCH_LABEL.0
   /* Each line containing a .dshw directive secures two bytes, */
   /* and the value obtained by shifting the value of operand */
   /* to the right by one bit is used in initialization. */
   /* At this time, if the highest-order bit (the sign bit) is */
   /* truncated, the branch table that is generated will be
   /* incorrect. */

.BB.LABEL.0:
  Processing A
.BB.LABEL.1:
  Processing B
.BB.LABEL.2:
  Processing C
------------------------------------------------
```

3. Workarounds

When the .dshw directive is directly used in the assembly file, change the value of the operand so that it does not satisfy the condition for the restriction.

Specifically, where the .dshw directive in an assembly file generated by compiling a C source file satisfies the condition, take either of actions (1) or (2) below.

(1) Change the branch table from two bytes to four bytes

Specify the -Xword_case option.
In CS+, specify "4 bytes (-Xword_case)" from the [Compiler Options] tabbed page -> [Output Code] Category -> [Label size of switch table].

(2) Change the setting so that code in the table jump format is not generated for the switch statement

Specify ifelse or binary with the -Xswitch option.
In CS+, specify either "if-else[-Xswitch=ifelse]" or "Binary search (-Xswitch=binary)" from [Compile Options] tabbed page -> [Output Code] Category ->[Output code of switch statement].

4. Schedule for Fixing the Problem

This problem will be fixed in a later revision of the product.
(The next release date has not yet been determined.)
