

RENESAS TOOL NEWS on January 16, 2009: 090116/tn2

Notes on Using the C/C++ Compiler Package for the H8SX, H8S, and H8 MCU Families

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

1. With overlaying sections with different alignment numbers each other (LNK-0004)
 2. With using the `data_stuff` and `noptimize` options in linking (LNK-0005)
-

1. Products and Versions Concerned

The C/C++ compiler packages for the H8SX, H8S, and H8 MCU families
V.6.00 Release 03 through V.6.02 Release 01

2. Two Problems in Optimizing Linkage Editor

2.1 With Overlaying Sections with Different Alignment Numbers (LNK-0004)

Version concerned:

V.6.01 Release 02 through V.6.02 Release01

Symptom:

If sections with alignment numbers different from each other are overlaid, aligning the section first to be overlaid will place other sections at incorrect addresses.

Conditions:

This symptom may arise if the following condition (1) or (2) is satisfied:

(1) The following conditions are all satisfied:

- (1-1) Overlays are specified by using parentheses '()', and there is a section before or behind parentheses '().'

(1-2) One or more overlays are specified for groups of sections (that is, one overlay for one group), and, the alignment numbers of the sections in a group are different from each other.

(1-3) The start address specified by the start option is not a common multiple of the alignment number of sections.

Example: `-start=(P1:P2),P3,(P4:P5),P6/100`

Here, one or more of the alignment numbers of sections P1, P3, P4, or P6 is different from others.

The start address of assignment is 100, and it is not a common multiple of numbers of the alignment number of sections P1, P3, P4, and P6.

(2) The following conditions are all satisfied.

(2-1) Overlays are not specified by using parentheses '()'.
The alignment number of the front section of the group first to be overlaid is greater than that of any front section of the other groups (overlaid later).

(2-2) The address specified by the start option is not a multiple of the alignment number of the front section of the group first to be overlaid.

Example: `-start=A1,A2:B1,B2:C1,C2/152`

Here, the alignment number of A1 is greater than those of B1 and C1.

The start address of assignment is 152, and it is not a multiple of the alignment number of section A1.

Workarounds:

Avoid the problem in any of the following ways:

(1) If the condition (1) is satisfied: Specify same value for all alignment number of the section.

(2) If the condition (2) is satisfied: Overlay the section group whose front section has the least alignment number first.

(3) Specify the common multiple of the alignment number of sections to the start address by start option.

2.2 With Using the `data_stuff` and `nooptimize` Options in Linking (LNK-0005)

Versions concerned:

Symptom:

If options `data_stuff` and `nooptimize` are selected to stuff data, odd addresses may be assigned to variables of even bytes in length.

Conditions:

This symptom may arise if the following conditions are all satisfied:

- (1) The `data_stuff` option is selected in linking.
- (2) The `nooptimize` option is selected at the same time.
- (3) Two or more object files are linked.
- (4) At least two object files contain a data section with the same name.
- (5) An object file specified second or later by the input file contains a data section with the same name in (4), and the last data item of the data section is 1 byte long.
- (6) In the data section in (5) does not exist the symbol with same alignment number as the alignment number of the section.

Example:

```
-----  
//a.c // Condition (4)  
char a;  
//b.c // Condition (4)  
short b; // Condition (6)  
char c; // Conditions (5) and (6)  
-----
```

Linker command

```
-----  
optlnk a.obj b.obj ?data_stuff -nooptimize  
-----
```

Workarounds:

Avoid the problem in any of the following ways:

- (1) Do not select the `data_stuff` option in linking.

(2) Select the source file where this symptom arises; then, in the section in which exist symbols that odd addresses are assigned to, define dummy variables whose length are the same as the alignment number of the section.

Example:

```
-----  
//b.c  
long dummy;  
short b;  
char c;  
-----
```

(3) Select the source file where this symptom arises; then, define dummy variables of 2 bytes long such a way that those are placed at the end of the section in which exist symbols that odd addresses are assigned to.

Example:

```
-----  
//b.c  
short b;  
char c;  
short dummy;  
-----
```

3. Schedule of Fixing the Problems

We plan to fix these problems in the V.6.02 Release 02 compiler package.

[Disclaimer]

The past news contents have been based on information at the time of publication. Now changed or invalid information may be included. The URLs in the Tool News also may be subject to change or become invalid without prior notice.