Hitachi America, Ltd.

TN-0096

Application Engineering
TechNote

Jennifer Ediyanto

HAL/GNU Code Optimization

The HAL/GNU H8/300 Software Tools provide some software switches to control the code optimization. This paper will discuss about **-O** and **-relax** optimization switches and their effect on the source code.

The -O compiler switch

With the -O switch, the compiler optimizes the source code by reducing code size and execution time. The shortcuts taken by optimized code may occasionally produce surprising results:

- Some variables you declared may not exist at all. If you declare variables that you do not use in the program, then the compiler will delete the unused variables.
- Flow of control may briefly move where you did not expect it because some statements have been deleted. If you use the debugger to single step over the program, the debugger will skip over the optimized statements.
- Some statements may not be executed because they compute constant results or their values were already at hand.
- Some statements may execute in different places because they were moved out of loops.

The following is a command line example that tells the compiler to generate an assembly file (**Test.s**) with optimization:

```
GCC  -O   -S   Test.c
```

where

GCC   name of the compiler.
-O   compiler switch to optimize the C source code.
-S   compiler switch to generate assembly file, i.e., Test.s.

The -relax linker switch

The -relax switch forces the linker to perform the global optimizations, which are:

- all **jsr** (jump to sub-routine) and **jmp** (jump) instructions whose targets are within eight bits will be turned into eight-bit program-counter relative **bsr** (branch to sub-routine) and **bra** (branch) instruction, respectively.
- all **mov.b** instructions that use the sixteen-bit absolute address form, but refer to the top page of memory will be changed to use the eight-bit address form.

  For example:
  The linker will turn

  ```
  mov.b @aa:16   ==>   mov.b @aa:8
  ```

  whenever the address **aa** is in the top page of memory.

The following is a command line example that produces the absolute file (**Test.x**) with the global optimizations:

```
LD  -o  Test.x   -relax  Test.o   ..\lib\libc.a
```

where

LD   name of the linker.
-o   the linker switch to name the output absolute file as Test.x. By default the output file is a.out.
-relax   the linker switch to perform the global optimization.

The information in this document has been carefully checked; however, the contents of this document may be changed and modified without notice. Hitachi America, Ltd. shall assume no responsibility for inaccuracies, or any problem involving a patent infringement caused when applying the descriptions in this document. This material is protected by copyright laws. © Copyright 1995, Hitachi America, Ltd. All rights reserved. Printed in U.S.A.