Hitachi America, Ltd.

Application Note

H8/300 GNU

Jennifer Ediyanto

GNU Command Line Switches

INTRODUCTION

Hitachi America provides the GNU H8/300 Software Tools to program the Hitachi H8/300 series micoprocessors. These software tools consist of C compiler, assembler, linker, and debugger.

This paper will provide some explanations on the most common-used command line switches of the GNU software tools.

This paper uses the following software tools:

- GCC C Cross Compiler
- AS Cross Assembler
- LD Linker

C CROSS COMPILER

When we invoke the GNU C Compiler, it normally does preprocessing, compilation, assembly and linking. The default command to invoke the compiler is:

GCC *source_filename*

Example: GCC Test.c

Result:

- Produce absolute file in COFF format.
- The absolute file is named: a.out.
- No optimizations.
- The starting address of the program is H'8000.

The command line switches allow us to stop this process at an intermediate stage. By default, all switches are off. The following command invokes the compiler with switches:

GCC [-*switch*] *source_filename*

where:

GCC	The name of the compiler.
-switch	Any of the command line switches. Each switch must be preceded by a dash (-) and is case sensitive. The switches can be specified in any order.
source_filename	The name of the input file containing one of these: - a C source file with .c extension - H8 family assembly files with .s extension - object files with .o extension

The most common-used switches are as follows:

1. **-c**

The -c switch causes the compiler to produce an object file. By producing object files, we can later link several programs together. The output object file will have **.o** extension. By default, the compiler will invoke the linker to produce an absolute file.

Example: **GCC -c Test.c**

2. **-S**

The -S switch causes the compiler to generate an assembly source file to be assembled by the Assembler. Producing the assembly source file will help the assembly programmer to read the C program and the C programmer to learn the syntax of Hitachi Assembler. The default output assembler file is the input file name with **.s** extension.

Example: **GCC -S Test.c**

Please see *Listing 1* and *Listing 2* for Test.c and Test.s files. Test.c is a simple C program that prints 'Hitachi America' ten times to the screen.

3. **-g**

With the -g switch, the compiler generates debugging information. It is essential to compile our C program with this switch for a high level debugging.

Example: **GCC -g Test.c**

4. **-o** *filename*

With the -o switch, the compiler places output in file *filename*. This applies regardless of whatever sort of output is being produced, whether it be an executable file, an object file, an assembler file or preprocessed C code. If -o switch is not specified, the default is to put an executable file in **a.out**, the object file for **source.suffix** in **source.o**, its assembler file in **source.s**, and all preprocessed C source on standard output.

Example: **GCC -o Test.x Test.c**

The above command will name the absolute file of Test.c as Test.x. By default the absolute file for Test.c is a.out.

5. **-O**

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.

Example: **GCC -O Test.c**

The -O switch has five level of optimizations. The -O switch can be followed by the number of level optimization, such as -O1, -O2, -O3, -O4, -O5. By default, the -O switch is level one optimization. The -O5 switch forces the compiler to perform the most aggressive optimization.

Different levels of optimizations perform different optimizations, which are:

-O and -O1:

- Omit the frame pointer in a register for functions that do not need one. This avoids the instructions to save, set up, and restore frame pointer. It also makes an extra register available in many functions.
- Accumulate function arguments on the stack for several function calls and pops them all at once.
- Optimizes the jump instructions to check if a jump branches to a location where another comparison subsumed by the first is found. If so, the first branch is redirected to either the destination of the second branch or a point immediately following it, depending on whether the condition is known to be true or false.

-O2, -O3, -O4, and -O5:

- Scan through jump instructions when the target of the jump is not reached by any other path.
- Follow jumps which conditionally skip over blocks.
- Performs a number of minor optimizations that are relatively expensive.
- Performs the optimizations of loop strength reduction and elimination of iteration variables.
- Re-run common subexpression elimination after loop optimizations has been performed.
- Enables values to be allocated in registers that will be clobbered by function calls, by emitting extra instructions to save and restore the registers around such calls.

Example**: GCC -o Test2.s -O5 -S Test.c**

The above command will produce an assembly file named Test2.s with optimization. Please see *Listing 3* for the Test2.s file and compare it with *Listing 2* (Test.s). We can see the compiler generated assembly file with optimization (Listing 3) has smaller code size as compare to assembly file without optimization (Listing 2).

6. **-Xlinker**

With the -Xlinker switch, the compiler will pass any switches following to the linker. This switch allows us to use linker switches when compiling the program.

Example: **GCC -o Test.x -Xlinker -M Test.c**

The above command will produce the absolute file named Test.x and output the linker map file to the screen. This -M switch causes the linker to produce the

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

linker map file. Please refer to the *Linker Section* for information about the -M switch.

7. **-Wa**

With the -Wa switch, the compiler will pass any switches following to the assembler. This allows us to use assembler switches when compiling the program. This switch must be followed by a coma (,) when used.

Example: **GCC -o Test.x -Wa,-al Test.c**

The above command will compile Test.c program to produce the absolute file Test.x and the assembly listing file by using the -al assembler switch. We can redirect the output assembly listing to a user-defined file. Please refer to *Assembler Section* for more information on the -al switch.

8. **-v**

With the -v switch, the compiler will display the commands executed to run the stages of compilation. This switch will also print the compiler's version. This switch is very useful to help us writing our own batch file, since this switch will display all the necessary command lines needed to produce the absolute file.

Example: **GCC -v Test.c**

9. **-V** *version*

Older and newer versions of GNU Compiler can be installed side by side. One of them (probably the newest) will be the default, but we may sometimes wish to use another. This switch specifies which version of GNU Compiler to run. The default version is controlled by the way GNU Compiler is installed. Normally, it will be the version that is recommended for general use.

Example: **GCC -V 2.1**

The above command means to run the GNU Compiler version 2.1.

CROSS ASSEMBLER

The default command to invoke the assembler is:

AS *source_filename*

Example: AS Test.s

Result:

- Produce an object file
- The object file name is a.out.

The following command invokes the assembler with switches:

AS [-*switch*] *source_filename*

where:

AS	The name of the assembler.
-*switch*	Any of the assembler command line switches. Each switch must be preceded by a dash (-) and is case sensitive.The switches can be specified in any order and there is no conflicting switches.
source_filename	The name of the assembly source file with .s extension.

The most common-used switches are as follows:

1. **-as**

With the -as switch, the assembler produces the symbol table listing on standard output (i.e., the screen). We can redirect the output listing to a user-defined file by using the DOS redirect sign (>).

Example: **AS -as Test2.s > Test.sym**

The above command will produce a symbol table listing called Test.sym. Please see *Listing 4* for Test.sym file.

2. **-al**

The -al switch causes the assembler to produce the assembly listing on standard output. We can redirect the output listing to a user-defined file by using the DOS redirect sign (>).

Example: **AS -al Test2.s > Test.lst**

The above command will produce the assembly listing called Test.lst. Please see *Listing 5* for Test.lst file.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

3. **-als**

The -als switch is a combination of -as and -al switches. With the -als switch, the assembler produces both the assembly and symbol table listing on standard output. We can redirect the output listing to a user-defined file by using the DOS redirect sign (>).

Example: **AS -als Test2.s > Test.lis**

The above command will produce the assembly and symbol table listing called Test.lis. Please see *Listing 6* for Test.lis file.

4. **-ahld**

The -ahld switch is similar to the -al switch, except it will intermix the assembly source listing with the C source code as comments. The input file must be the assembly file that is produced by the compiler with the -g switch. The -g switch is a compiler switch that generates debugging information. For more information on -g switch, please refer to the *Compiler Section*. We can redirect the output listing to a user-defined file by using the DOS redirect sign (>).

Example: **GCC -o Test2.s -g -O5 -S Test.c**
AS -ahld Test2.s > Test2.lst

The first command will produce the Test2.s file with debugging information. The second command will produce the assembly listing called Test2.lst. Please see *Listing 7* for Test2.lst and compare it with *Listing 5* (Test.lst). We can see that Listing 7 has intermixed C source code with assembly code and Listing 5 has only assembly code.

5. **-a**

The -a switch is a combination of -as and -ahld switches. With the -a switch, the assembler produces both the intermixed assembly and symbol table listing on standard output. We can redirect the output listing to a user-defined file by using the DOS redirect sign (>).

Example: **AS -a Test2.s > Test2.lis**

The above command will produce the assembly and symbol table listing called Test2.lis. Please see *Listing 8* for Test2.lis file and compare it with *Listing 6* (Test.lis). We can see that Listing 8 has more information on symbols and intermixed C source code with assembly code and Listing 6 has a limited information on symbols and list assembly code only.

6. **-o *filename***

With the -o switch, the assembler places output object file in *filename*. By default the assembler will put the output object file in **a.out**. In order to avoid confusion, it is essential to name the output object file to an appropriate file name.

Example: **AS -o Test.o Test.s**

The above command will produce an object file named Test.o.

7. **-v**

We can find out what version of assembler is running by using the -v switch.

Example: **AS -v**

8. **-W**

This switch suppresses the warning messages. This switch only effects the warning messages, any errors that stop the assembler are still reported.

Example: **AS -W -o Test.o Test.s**

The above command will produce the object file called Test.o while eliminating any warning messages.

LINKER

The default command to invoke the linker is:

LD *object_files standard_library*

Example: LD Test.o ..\lib\libc.a

Result:

- Produce absolute file in COFF format.
- The absolute file is named: a.out.
- The starting address of the program is H'8000.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

The following command invokes the linker with switches:

LD [-*switch*] *object_files standard_library*

where:

LD	The name of the linker.
-*switch*	Any of the linker command line switches. Each switch must be preceded by a dash (-) and is case sensitive. The switches can be specified in any order and may be repeated at will.
object_files	The list of the object files with .o extension.

The most common-used switches are as follows:

1. **-o *filename***

With the -o switch, the linker places output absolute file in *filename*. By default the linker will put the output absolute file in **a.out**. In order to avoid confusion, it is essential to name the output absolute file to an appropriate file name. By default the assembler and compiler also places their output file in a.out.

Example: **LD -o Test.x Test.o ..\lib\libc.a**

The above command line tells the linker to produce an absolute file called **Test.x** as the result of linking the file **Test.o** and the library **libc.a** that will come from the standard search directories.

2. **-M**

With the -M switch, the linker will produce a linker map, diagnostic information about where symbols are mapped by the linker, and information on global common storage allocation. The linker will print the linker map to the standard output, but we can redirect the output to the user-defined file (Test.map) by using the DOS redirect sign (>).

Example:

LD -o Test.x -M Test.o ..\lib\libc.a > Test.map

The above command will produce the absolute file named Test.x and the map file named Test.map. Please see *Listing 9* for Test.map file.

3. **-i**

With the -i switch, the linker will perform the incremental linking, generating an output object file that can in turn serve as input to the linker.

Example:

LD -o Main.o -i Test1.o Test2.o ..\lib\libc.a

The above command line will link Test1.o and Test2.o files and produce an output object file called Main.o.

4. **-relax**

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. By default, the compiler uses jsr and jmp instructions instead of bsr and bra instructions.
- 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 turns
  `mov.b @aa:16 ==> mov.b @aa:8`
  whenever the address **aa** is in the top page of memory.

Example:

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

The above command line will produce the absolute file called **Test.x** with global optimizations.

5. **-T*linker_commandfile***

The -T switch directs the linker to read linker commands from the file *linker_commandfile*. The linker command file consists of all necessary linker commands to produce the absolute file.

Example: **LD -TTest.ld Test.o ..\lib\libc.a**

The above command will produce an absolute file in Motorola S-record format because there is a command to specify the format file, output_format(srec), in the Test.ld linker command file. Please see *Listing 10* for Test.ld file. *For more information on the GNU Linker Command File, please refer to TN-0093*.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

6. **-e *function_name***

With the -e switch, we can specify the starting point of a program at certain function. The specified function name is a predefined function in the program. The function name must be preceded by an underscore because the linker will only recognize the compiler generated symbols for function names. The compiler always puts an underscore in front of all function names.

Example:

LD -o Test.x -e _main Test.o ..\lib\libc.a

The above command will produce an absolute file named Test.x with the starting point at a function called main. The function main is a predefined function in Test.c program.

7. **-v**

We can find out what version of linker is running by using the -v switch.

Example: **LD -v**

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

Listing 1. **Test.c** (begin)

```
#include <stdio.h>


main()
{
        int i;
        for (i = 0; i < 10; i++)
                iprintf("Hitachi America %d\n",i);
}
```

Listing 1. **Test.c** (end)

Listing 2. **Test.s** (begin)

```
;       GCC For the Hitachi H8/300
;       By Hitachi America Ltd and Cygnus Support
;       release 2.0


        .file   "Test.c"
        .section .text
        .def    _size_t
        .scl    13
        .type   016
        .endef
        .def    ___gnuc_va_list
        .scl    13
        .type   021
        .endef
        .def    _fpos_t
        .scl    13
        .type   05
        .endef
        .def    ___sbuf
        .scl    10
        .type   010
        .size   4
        .endef
        .def    __base
        .val    0
        .scl    8
        .type   034
        .endef
        .def    __size
        .val    2
        .scl    8
        .type   03
        .endef
        .def    .eos
        .val    4
        .scl    102
        .tag    ___sbuf
        .size   4
        .endef
        .def    ___sFILE
        .scl    10
        .type   010
        .size   46
        .endef
        .def    __p
```


HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

```
        .val    0
        .scl    8
        .type   034
        .endef
        .def    __r
        .val    2
        .scl    8
        .type   03
        .endef
        .def    __w
        .val    4
        .scl    8
        .type   03
        .endef
        .def    __flags
        .val    6
        .scl    8
        .type   03
        .endef
        .def    __file
        .val    8
        .scl    8
        .type   03
        .endef
        .def    __bf
        .val    10
        .scl    8
        .tag    ___sbuf
        .size   4
        .type   010
        .endef
        .def    __lbfsize
        .val    14
        .scl    8
        .type   03
        .endef
        .def    __cookie
        .val    16
        .scl    8
        .type   021
        .endef
        .def    __read
        .val    18
        .scl    8
        .type   0223
        .endef
        .def    __write
        .val    20
        .scl    8
        .type   0223
        .endef
        .def    __seek
        .val    22
        .scl    8
        .type   0225
        .endef
        .def    __close
        .val    24
        .scl    8
        .type   0223
        .endef
        .def    __ub
        .val    26
```


HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

```
        .scl    8
        .tag    ___sbuf
        .size   4
        .type   010
        .endef
        .def    __up
        .val    30
        .scl    8
        .type   034
        .endef
        .def    __ur
        .val    32
        .scl    8
        .type   03
        .endef
        .def    __ubuf
        .val    34
        .scl    8
        .dim    3
        .size   3
        .type   074
        .endef
        .def    __nbuf
        .val    37
        .scl    8
        .dim    1
        .size   1
        .type   074
        .endef
        .def    __lb
        .val    38
        .scl    8
        .tag    ___sbuf
        .size   4
        .type   010
        .endef
        .def    __blksize
        .val    42
        .scl    8
        .type   03
        .endef
        .def    __offset
        .val    44
        .scl    8
        .type   03
        .endef
        .def    .eos
        .val    46
        .scl    102
        .tag    ___sFILE
        .size   46
        .endef
        .def    _FILE
        .scl    13
        .tag    ___sFILE
        .size   46
        .type   010
        .endef
        .global ___swbuf
        .global _iprintf
.LC0:
        .ascii "Hitachi America %d\12\0"
        .align 2
```


HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

```
        .def    _main
        .val    _main
        .scl    2
        .type   043
        .endef
        .global _main
_main:
        push    r6
        mov.w   r7,r6
        subs    #2,sp
        .def    .bf
        .val    .
        .scl    101
        .line   5
        .endef
        jsr     @___main
        .ln     2
        .def    .bb
        .val    .
        .scl    100
        .line   2
        .endef
        .def    _i
        .val    -2
        .scl    1
        .type   03
        .endef
        .ln     3
        sub.w   r0,r0
        mov.w   r0,@(-2,r6)
.L6:
        mov.w   @(-2,r6),r0
        mov.w   #9,r1
        cmp.w   r1,r0
        ble     tl500
        jmp     @.L7
        tl500:
        mov.w   @(-2,r6),r0
        mov.w   r0,@-r7
        mov.w   #.LC0,r0
        mov.w   r0,@-r7
        jsr     @_iprintf
        adds    #2,r7
        adds    #2,r7
.L8:
        mov.w   @(-2,r6),r1
        mov.w   r1,r0
        mov.w   r1,r0
        adds    #1,r0
        mov.w   r0,@(-2,r6)
        jmp     @.L6
.L7:
        .ln     5
        .def    .eb
        .val    .
        .scl    100
        .line   5
        .endef
.L5:
        .def    .ef
        .val    .
        .scl    101
        .line   5
```


HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

```
        .endef
        adds    #2,sp
        pop     r6
        rts
        .def    _main
        .val    .
        .scl    -1
        .endef
        .end
```

Listing 2. **Test.s** (end)

Listing 3. **Test2.s** (begin)

```
;       GCC For the Hitachi H8/300
;       By Hitachi America Ltd and Cygnus Support
;       release 2.0
; -O5


        .file   "Test.c"
        .global ___swbuf
        .global _iprintf
        .section .text
.LC0:
        .ascii "Hitachi America %d\12\0"
        .align 2
        .global _main
_main:
        push    r2
        jsr     @___main
        sub.w   r2,r2
.L9:
        mov.w   r2,@-r7
        mov.w   #.LC0,r0
        mov.w   r0,@-r7
        jsr     @_iprintf
        adds    #2,r7
        adds    #2,r7
        adds    #1,r2
        mov.w   #9,r0
        cmp.w   r0,r2
        ble     .L9
        pop     r2
        rts
        .end
```

Listing 3. **Test2.s** (end)

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

Listing 4. **Test.sym** (begin)

```
Hitachi H8/300 GAS  test2.s                     page 1


DEFINED SYMBOLS
             test2.s:11      1:00000000 .LC0
             test2.s:15      1:00000014 _main
             test2.s:19      1:0000001c .L9

UNDEFINED SYMBOLS
.file
.text
.data
.bss
___swbuf
_iprintf
___main
```

Listing 4. **Test.sym** (end)

Listing 5. **Test.lst** (begin)

```
Hitachi H8/300 GAS  test2.s                     page 1

   1                    ;       GCC For the Hitachi H8/300
   2                    ;       By Hitachi America Ltd and Cygnus Support
   3                    ;       release 2.0
   4                    ; -O5
   5
   6
   7                            .file   "Test.c"
   8                            .global ___swbuf
   9                            .global _iprintf
  10                            .section .text
  11                    .LC0:
  12 0000 48697461              .ascii "Hitachi America %d\12\0"
  12      63686920
  12      416D6572
  12      69636120
  12      25640A00
  13 0014 0000                  .align 2
  14                            .global _main
  15                    _main:
  16 0014 6DF2                  push    r2
  17 0016 5E000000              jsr     @___main
  18 001a 1922                  sub.w   r2,r2
  19                    .L9:
  20 001c 6DF2                  mov.w   r2,@-r7
  21 001e 79000000              mov.w   #.LC0,r0
  22 0022 6DF0                  mov.w   r0,@-r7
  23 0024 5E000000              jsr     @_iprintf
  24 0028 0B87                  adds    #2,r7
  25 002a 0B87                  adds    #2,r7
  26 002c 0B02                  adds    #1,r2
  27 002e 79000009              mov.w   #9,r0
  28 0032 1D02                  cmp.w   r0,r2
  29 0034 4F00                  ble     .L9
  30 0036 6D72                  pop     r2
  31 0038 5470                  rts
  32 003a 00                    .end
```

Listing 5. **Test.lst** (end)

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

Listing 6. **Test.lis** (begin)

```
Hitachi H8/300 GAS  test2.s                     page 1


   1                    ;       GCC For the Hitachi H8/300
   2                    ;       By Hitachi America Ltd and Cygnus Support
   3                    ;       release 2.0
   4                    ; -O5
   5
   6
   7                            .file   "Test.c"
   8                            .global ___swbuf
   9                            .global _iprintf
  10                            .section .text
  11                    .LC0:
  12 0000 48697461              .ascii "Hitachi America %d\12\0"
  12      63686920
  12      416D6572
  12      69636120
  12      25640A00
  13 0014 0000                  .align 2
  14                            .global _main
  15                    _main:
  16 0014 6DF2                  push    r2
  17 0016 5E000000              jsr     @___main
  18 001a 1922                  sub.w   r2,r2
  19                    .L9:
  20 001c 6DF2                  mov.w   r2,@-r7
  21 001e 79000000              mov.w   #.LC0,r0
  22 0022 6DF0                  mov.w   r0,@-r7
  23 0024 5E000000              jsr     @_iprintf
  24 0028 0B87                  adds    #2,r7
  25 002a 0B87                  adds    #2,r7
  26 002c 0B02                  adds    #1,r2
  27 002e 79000009              mov.w   #9,r0
  28 0032 1D02                  cmp.w   r0,r2
  29 0034 4F00                  ble     .L9
  30 0036 6D72                  pop     r2
  31 0038 5470                  rts
  32 003a 00                    .end


Hitachi H8/300 GAS  test2.s                     page 2


DEFINED SYMBOLS
             test2.s:11      1:00000000 .LC0
             test2.s:15      1:00000014 _main
             test2.s:19      1:0000001c .L9

UNDEFINED SYMBOLS
.file
.text
.data
.bss
___swbuf
_iprintf
___main
```

Listing 6. **Test.lis** (end)

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

Listing 7. **Test2.lst** (begin)

```
Hitachi H8/300 GAS  test2.s                     page 1


   1                    ;       GCC For the Hitachi H8/300
   2                    ;       By Hitachi America Ltd and Cygnus Support
   3                    ;       release 2.0
   4                    ; -O5
   5
   6
   7                            .file   "test.c"
   8                            .section .text
 169                            .global ___swbuf
 170                            .global _iprintf
 171                    .LC0:
 172 0000 48697461              .ascii "Hitachi America %d\12\0"
 172      63686920
 172      416D6572
 172      69636120
 172      25640A00
 173 0014 0000                  .align 2
 179                            .global _main
 180                    _main:
 181 0014 6DF2                  push    r2
 187 0016 5E000000              jsr     @___main
   1:test.c         **** #include <stdio.h>
   2:test.c         ****
   3:test.c         ****
   4:test.c         **** main()
   5:test.c         **** {
   6:test.c         ****           int i;
   7:test.c         ****           for (i = 0; i < 10; i++)
 200 001a 1922                  sub.w   r2,r2
 201                    .L9:
 202 001c 6DF2                  mov.w   r2,@-r7
 203 001e 79000000              mov.w   #.LC0,r0
 204 0022 6DF0                  mov.w   r0,@-r7
 205 0024 5E000000              jsr     @_iprintf
 206 0028 0B87                  adds    #2,r7
 207 002a 0B87                  adds    #2,r7
 208 002c 0B02                  adds    #1,r2
 209 002e 79000009              mov.w   #9,r0
 210 0032 1D02                  cmp.w   r0,r2
 211 0034 4F00                  ble     .L9
   8:test.c         ****                   iprintf("Hitachi America %d\n",i);
   9:test.c         **** }
 223 0036 6D72                  pop     r2
 224 0038 5470                  rts
 229 003a 00                    .end
```

Listing 7. **Test2.lst** (end)

Listing 8. **Test2.lis** (begin)

```
Hitachi H8/300 GAS  test2.s                     page 1


   1                    ;       GCC For the Hitachi H8/300
   2                    ;       By Hitachi America Ltd and Cygnus Support
   3                    ;       release 2.0
   4                    ; -O5
   5
   6
   7                            .file   "test.c"
   8                            .section .text
   9                            .def    _size_t
  10                            .scl    13
  11                            .type   016
  12                            .endef
  13                            .def    ___gnuc_va_list
  14                            .scl    13
  15                            .type   021
  16                            .endef
  17                            .def    _fpos_t
  18                            .scl    13
  19                            .type   05
  20                            .endef
  21                            .def    ___sbuf
  22                            .scl    10
  23                            .type   010
  24                            .size   4
  25                            .endef
  26                            .def    __base
  27                            .val    0
  28                            .scl    8
  29                            .type   034
  30                            .endef
  31                            .def    __size
  32                            .val    2
  33                            .scl    8
  34                            .type   03
  35                            .endef
  36                            .def    .eos
  37                            .val    4
  38                            .scl    102
  39                            .tag    ___sbuf
  40                            .size   4
  41                            .endef
  42                            .def    ___sFILE
  43                            .scl    10
  44                            .type   010
  45                            .size   46
  46                            .endef
  47                            .def    __p
  48                            .val    0
  49                            .scl    8
  50                            .type   034
  51                            .endef
  52                            .def    __r
  53                            .val    2
  54                            .scl    8
  55                            .type   03
  56                            .endef
  57                            .def    __w
```

```
Hitachi H8/300 GAS  test2.s                     page 2
  58                            .val    4
  59                            .scl    8
  60                            .type   03
  61                            .endef
  62                            .def    __flags
  63                            .val    6
  64                            .scl    8
  65                            .type   03
  66                            .endef
  67                            .def    __file
  68                            .val    8
  69                            .scl    8
  70                            .type   03
  71                            .endef
  72                            .def    __bf
  73                            .val    10
  74                            .scl    8
  75                            .tag    ___sbuf
  76                            .size   4
  77                            .type   010
  78                            .endef
  79                            .def    __lbfsize
  80                            .val    14
  81                            .scl    8
  82                            .type   03
  83                            .endef
  84                            .def    __cookie
  85                            .val    16
  86                            .scl    8
  87                            .type   021
  88                            .endef
  89                            .def    __read
  90                            .val    18
  91                            .scl    8
  92                            .type   0223
  93                            .endef
  94                            .def    __write
  95                            .val    20
  96                            .scl    8
  97                            .type   0223
  98                            .endef
  99                            .def    __seek
 100                            .val    22
 101                            .scl    8
 102                            .type   0225
 103                            .endef
 104                            .def    __close
 105                            .val    24
 106                            .scl    8
 107                            .type   0223
 108                            .endef
 109                            .def    __ub
 110                            .val    26
 111                            .scl    8
 112                            .tag    ___sbuf
 113                            .size   4
 114                            .type   010
 115                            .endef
 116                            .def    __up
 117                            .val    30
 118                            .scl    8
 119                            .type   034
```


HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

```
Hitachi H8/300 GAS  test2.s                     page 3
 120                            .endef
 121                            .def    __ur
 122                            .val    32
 123                            .scl    8
 124                            .type   03
 125                            .endef
 126                            .def    __ubuf
 127                            .val    34
 128                            .scl    8
 129                            .dim    3
 130                            .size   3
 131                            .type   074
 132                            .endef
 133                            .def    __nbuf
 134                            .val    37
 135                            .scl    8
 136                            .dim    1
 137                            .size   1
 138                            .type   074
 139                            .endef
 140                            .def    __lb
 141                            .val    38
 142                            .scl    8
 143                            .tag    ___sbuf
 144                            .size   4
 145                            .type   010
 146                            .endef
 147                            .def    __blksize
 148                            .val    42
 149                            .scl    8
 150                            .type   03
 151                            .endef
 152                            .def    __offset
 153                            .val    44
 154                            .scl    8
 155                            .type   03
 156                            .endef
 157                            .def    .eos
 158                            .val    46
 159                            .scl    102
 160                            .tag    ___sFILE
 161                            .size   46
 162                            .endef
 163                            .def    _FILE
 164                            .scl    13
 165                            .tag    ___sFILE
 166                            .size   46
 167                            .type   010
 168                            .endef
 169                            .global ___swbuf
 170                            .global _iprintf
 171                    .LC0:
 172 0000 48697461              .ascii "Hitachi America %d\12\0"
 172      63686920
 172      416D6572
 172      69636120
 172      25640A00
 173 0014 0000                  .align 2
 174                            .def    _main
 175                            .val    _main
 176                            .scl    2
 177                            .type   043
```


HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

```
Hitachi H8/300 GAS  test2.s                       page 4
 178                            .endef
 179                            .global _main
 180                        _main:
 181 0014 6DF2                  push    r2
 182                            .def    .bf
 183                            .val    .
 184                            .scl    101
 185                            .line   5
 186                            .endef
 187 0016 5E000000              jsr     @___main
   1:test.c        **** #include <stdio.h>
   2:test.c        ****
   3:test.c        ****
   4:test.c        **** main()
   5:test.c        **** {
   6:test.c        ****           int i;
 188                            .ln     2
 189                            .def    .bb
 190                            .val    .
 191                            .scl    100
 192                            .line   2
 193                            .endef
 194                            .def    _i
 195                            .val    2
 196                            .scl    4
 197                            .type   03
 198                            .endef
   7:test.c        ****           for (i = 0; i < 10; i++)
 199                            .ln     3
 200 001a 1922                  sub.w   r2,r2
 201                        .L9:
 202 001c 6DF2                  mov.w   r2,@-r7
 203 001e 79000000              mov.w   #.LC0,r0
 204 0022 6DF0                  mov.w   r0,@-r7
 205 0024 5E000000              jsr     @_iprintf
 206 0028 0B87                  adds    #2,r7
 207 002a 0B87                  adds    #2,r7
 208 002c 0B02                  adds    #1,r2
 209 002e 79000009              mov.w   #9,r0
 210 0032 1D02                  cmp.w   r0,r2
 211 0034 4F00                  ble     .L9
   8:test.c        ****                   iprintf("Hitachi America %d\n",i);
   9:test.c        **** }
 212                            .ln     5
 213                            .def    .eb
 214                            .val    .
 215                            .scl    100
 216                            .line   5
 217                            .endef
 218                            .def    .ef
 219                            .val    .
 220                            .scl    101
 221                            .line   5
 222                            .endef
 223 0036 6D72                  pop     r2
 224 0038 5470                  rts
 225                            .def    _main
 226                            .val    .
 227                            .scl    -1
 228                            .endef
 229 003a 00                    .end
```


HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

```
Hitachi H8/300 GAS  test2.s                     page 5
DEFINED SYMBOLS
             test2.s:171     1:00000000 .LC0
             test2.s:180     1:00000014 _main
             test2.s:183     1:00000016 .bf
             test2.s:190     1:0000001a .bb
             test2.s:201     1:0000001c .L9
             test2.s:214     1:00000036 .eb
             test2.s:219     1:00000036 .ef

UNDEFINED SYMBOLS
.file
_size_t
___gnuc_va_list
_fpos_t
___sbuf
__base
__size
.eos
___sFILE
__p
__r
__w
__flags
__file
__bf
__lbfsize
__cookie
__read
__write
__seek
__close
__ub
__up
__ur
__ubuf
__nbuf
__lb
__blksize
__offset
.eos
_FILE
_i
.text
.data
.bss
___swbuf
_iprintf
___main
```

Listing 8. **Test2.lis** (end)

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

Listing 9. **Test.map** (begin)

```
**FILES**

  test.o
00000000 0000003a 2** 1d .text
00000000 00000000 2** 1d .data
00000000 00000000 2** 1d .bss
0000033e 00000000 2** 1d COMMON

  iprintf.o
0000003a 00000020 2** 1d .text
00000000 00000000 2** 1d .data
00000000 00000000 2** 1d .bss
0000033e 00000000 2** 1d COMMON

  vfiprintf.o
0000005a 00000950 2** 1d .text
00000000 00000000 2** 1d .data
00000000 00000000 2** 1d .bss
0000033e 00000000 2** 1d COMMON

  wbuf.o
000009aa 000000a4 2** 1d .text
00000000 00000000 2** 1d .data
00000000 00000000 2** 1d .bss
0000033e 00000000 2** 1d COMMON

  wsetup.o
00000a4e 000000c0 2** 1d .text
00000000 00000000 2** 1d .data
00000000 00000000 2** 1d .bss
0000033e 00000000 2** 1d COMMON

  malloc.o
00000b0e 0000033a 2** 1d .text
00000000 00000078 2** 1d .data
00000000 00000000 2** 1d .bss
0000033e 00000000 2** 1d COMMON

  __main.o
00000e48 00000002 2** 1d .text
00000078 00000000 2** 1d .data
00000000 00000000 2** 1d .bss
0000033e 00000000 2** 1d COMMON

  mulhi3.o
00000e4a 00000014 2** 1d .text
00000078 00000000 2** 1d .data
00000000 00000000 2** 1d .bss
0000033e 00000000 2** 1d COMMON

  syscalls.o
00000e5e 000000a0 2** 1d .text
00000078 00000000 2** 1d .data
00000000 00000002 2** 1d .bss
0000033e 00000000 2** 1d COMMON

  memcpy.o
00000efe 00000030 2** 1d .text
00000078 00000000 2** 1d .data
00000002 00000000 2** 1d .bss
```


HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

```
0000033e 00000000 2** 1d COMMON


strlen.o
00000f2e 00000014 2** 1d .text
00000078 00000000 2** 1d .data
00000002 00000000 2** 1d .bss
0000033e 00000000 2** 1d COMMON

  cvt.o
00000f42 000005ee 2** 1d .text
00000078 00000006 2** 1d .data
00000002 00000000 2** 1d .bss
0000033e 00000000 2** 1d COMMON

  fflush.o
00001530 000000ba 2** 1d .text
0000007e 00000000 2** 1d .data
00000002 00000000 2** 1d .bss
0000033e 00000000 2** 1d COMMON

  findfp.o
000015ea 0000013e 2** 1d .text
0000007e 0000009a 2** 1d .data
00000002 0000033c 2** 1d .bss
0000033e 00000006 2** 1d COMMON

  fiprintf.o
00001728 00000020 2** 1d .text
00000118 00000000 2** 1d .data
0000033e 00000000 2** 1d .bss
00000344 00000000 2** 1d COMMON

  fputc.o
00001748 00000046 2** 1d .text
00000118 00000000 2** 1d .data
0000033e 00000000 2** 1d .bss
00000344 00000000 2** 1d COMMON

  fputs.o
0000178e 00000050 2** 1d .text
00000118 00000000 2** 1d .data
0000033e 00000000 2** 1d .bss
00000344 00000000 2** 1d COMMON

  fvwrite.o
000017de 00000406 2** 1d .text
00000118 00000000 2** 1d .data
0000033e 00000000 2** 1d .bss
00000344 00000000 2** 1d COMMON

  fwalk.o
00001be4 00000084 2** 1d .text
00000118 00000000 2** 1d .data
0000033e 00000000 2** 1d .bss
00000344 00000000 2** 1d COMMON

  makebuf.o
00001c68 00000134 2** 1d .text
00000118 00000000 2** 1d .data
0000033e 00000000 2** 1d .bss
00000344 00000000 2** 1d COMMON
```


HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

```
  stdio.o
00001d9c 00000122 2** 1d .text
00000118 00000000 2** 1d .data
0000033e 00000000 2** 1d .bss
00000344 00000000 2** 1d COMMON
  divhi3.o
00001ebe 00000076 2** 1d .text
00000118 00000000 2** 1d .data
0000033e 00000000 2** 1d .bss
00000344 00000000 2** 1d COMMON

  divsi3.o
00001f34 000000f4 2** 1d .text
00000118 00000000 2** 1d .data
0000033e 00000000 2** 1d .bss
00000344 00000000 2** 1d COMMON

  memchr.o
00002028 00000034 2** 1d .text
00000118 00000000 2** 1d .data
0000033e 00000000 2** 1d .bss
00000344 00000000 2** 1d COMMON

  memmove.o
0000205c 0000005a 2** 1d .text
00000118 00000000 2** 1d .data
0000033e 00000000 2** 1d .bss
00000344 00000000 2** 1d COMMON

**GLOBAL SYMBOLS**

offset    section    offset   symbol
00000014  .text      00000014 _main
00000000  .text      00000000 ___swbuf
00000000  .text      00000000 _iprintf
00000000  .text      00000000 ___main
undefined                     ___builtin_next_arg
000001c8  .text      000001c8 _vfiprintf
00000008  .data      00000146 ___sF
00000000  .text      00000000 _fputs
00000000  .text      00000000 _fputc
00000000  .text      00000000 _strlen
00000460  .text      00000460 __sicvt
00000024  .text      00000024 __licvt
000002a4  .text      000002a4 __icvt
00000000  .text      00000000 ___mulhi3
00000000  .text      00000000 ___swsetup
00000000  .text      00000000 _fflush
00000000  COMMON     00000000 ___sdidinit
0000012c  .text      0000012c ___sinit
00000160  .text      00000160 _free
00000000  .text      00000000 ___smakebuf
00000000  .text      00000000 _morecore
000000d8  .text      000000d8 _malloc
00000198  .text      00000198 _realloc
00000270  .text      00000270 _mstats
00000060  .text      00000060 _sbrk
00000000  .text      00000000 _memcpy
00000000  .text      00000000 _fiprintf
00000000  .text      00000000 _read
00000004  .text      00000004 _lseek
00000018  .text      00000018 _write
00000058  .text      00000058 _close
```


HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

```
00000084  .text      00000084 _isatty
0000008c  .text      0000008c _fstat
0000009c  .text      0000009c _open
00000344  .bss       0000a512 _end
0000005c  .text      0000005c ___divsi3
00000030  .text      00000030 ___modsi3
00000034  .text      00000034 ___umodhi3
0000003a  .text      0000003a ___udivhi3
0000001c  .text      0000001c ___divhi3
0000002c  .text      0000002c ___modhi3
00000000  .text      00000000 __fwalk
00000094  .data      000001d2 ___sglue
00000000  .text      00000000 ___sfmoreglue
0000006c  .text      0000006c ___sfp
0000011c  .text      0000011c __cleanup
00000002  COMMON     00000002 ___cleanup
00000000  .text      00000000 ___sread
00000048  .text      00000048 ___swrite
000000ac  .text      000000ac ___sseek
00000110  .text      00000110 ___sclose
00000004  COMMON     00000004 _errno
00000000  .text      00000000 ___sfvwrite
00000000  .text      00000000 _memmove
00000000  .text      00000000 _memchr
00000040  .text      00000040 ___udivsi3
0000004c  .text      0000004c ___umodsi3
000020b6  .text      0000a0b6 _etext
00000118  .data      0000a1ce _edata
00000000  .bss       0000a1ce _bss_start
00000000  .stack     0000f000 _stack
**MEMORY CONFIGURATION**

name            origin   length   r_size   c_size    is    attributes
rom             00000000 00007fe0 00000000 00000000 ()
duart           00007fe0 00000010 00000000 00000000 ()
ram             00008000 00007000 00000000 00002512 ()
topram          0000f000 00000400 00000000 00000000 ()
hmsram          0000fb80 00000200 00000000 00000000 ()


**LINK EDITOR MEMORY MAP**

output   input     virtual
section  section   address    tsize


*ABS*             00000000     0     0 2**0
                                   forced alignment section 2**0 subsection
2**0
LOAD test.o
LOAD ..\lib\libc.a
OUTPUT(a.out coff-h8300)

.text             00008000  20b6     0 2**1 load alloc reloc contents
 from *(.text)
         .text    00008000    3a    3a 2**1  coff-h8300 test.o(overhead 9128 bytes)
                  00008014              _main
         .text    0000803a    20    20 2**1  coff-h8300
[..\lib\libc.a]iprintf.o(overhead 9300 bytes)
                  0000803a              _iprintf
         .text    0000805a   950   950 2**1  coff-h8300
[..\lib\libc.a]vfiprintf.o(overhead 34522 bytes)
                  00008222              _vfiprintf
```


HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

```
         .text    000089aa    a4    a4 2**1  coff-h8300
[..\lib\libc.a]wbuf.o(overhead 11010 bytes)
                  000089aa              ___swbuf
         .text    00008a4e    c0    c0 2**1  coff-h8300
[..\lib\libc.a]wsetup.o(overhead 12700 bytes)
                  00008a4e              ___swsetup
         .text    00008b0e   33a   33a 2**1  coff-h8300
[..\lib\libc.a]malloc.o(overhead 26266 bytes)
                  00008b0e               _morecore
                  00008be6               _malloc
                  00008c6e               _free
                  00008ca6               _realloc
                  00008d7e               _mstats
         .text    00008e48     2     2 2**1  coff-h8300
[..\lib\libc.a]__main.o(overhead 3894 bytes)
                  00008e48              ___main
         .text    00008e4a    14    14 2**1  coff-h8300
[..\lib\libc.a]mulhi3.o(overhead 3894 bytes)
                  00008e4a              ___mulhi3
         .text    00008e5e    a0    a0 2**1  coff-h8300
[..\lib\libc.a]syscalls.o(overhead 19350 bytes)
                  00008e5e               _read
                  00008e62               _lseek
                  00008e76               _write
                  00008eb6               _close
                  00008ebe               _sbrk
                  00008ee2               _isatty
                  00008eea               _fstat
                  00008efa               _open
         .text    00008efe    30    30 2**1  coff-h8300
[..\lib\libc.a]memcpy.o(overhead 6084 bytes)
                  00008efe               _memcpy
         .text    00008f2e    14    14 2**1  coff-h8300
[..\lib\libc.a]strlen.o(overhead 5700 bytes)
                  00008f2e               _strlen
         .text    00008f42   5ee   5ee 2**1  coff-h8300
[..\lib\libc.a]cvt.o(overhead 36960 bytes)
                  00008f66               __licvt
                  000091e6               __icvt
                  000093a2               __sicvt
         .text    00009530    ba    ba 2**1  coff-h8300
[..\lib\libc.a]fflush.o(overhead 11148 bytes)
                  00009530               _fflush
         .text    000095ea   13e   13e 2**1  coff-h8300
[..\lib\libc.a]findfp.o(overhead 19654 bytes)
                  000095ea              ___sfmoreglue
                  00009656              ___sfp
                  00009706              __cleanup
                  00009716              ___sinit
         .text    00009728    20    20 2**1  coff-h8300
[..\lib\libc.a]fiprintf.o(overhead 9570 bytes)
                  00009728               _fiprintf
         .text    00009748    46    46 2**1  coff-h8300
[..\lib\libc.a]fputc.o(overhead 10134 bytes)
                  00009748               _fputc
         .text    0000978e    50    50 2**1  coff-h8300
[..\lib\libc.a]fputs.o(overhead 11206 bytes)
                  0000978e               _fputs
         .text    000097de   406   406 2**1  coff-h8300
[..\lib\libc.a]fvwrite.o(overhead 19086 bytes)
                  000097de              ___sfvwrite
         .text    00009be4    84    84 2**1  coff-h8300
[..\lib\libc.a]fwalk.o(overhead 12768 bytes)
```

```
                00009be4                __fwalk
        .text   00009c68   134   134 2**1  coff-h8300
[..\lib\libc.a]makebuf.o(overhead 19346 bytes)
                00009c68                ___smakebuf
        .text   00009d9c   122   122 2**1  coff-h8300
[..\lib\libc.a]stdio.o(overhead 22642 bytes)
                00009d9c                ___sread
                00009de4                ___swrite
                00009e48                ___sseek
                00009eac                ___sclose
        .text   00009ebe    76    76 2**1  coff-h8300
[..\lib\libc.a]divhi3.o(overhead 5678 bytes)
                00009eda                ___divhi3
                00009ef8                ___udivhi3
                00009eea                ___modhi3
                00009ef2                ___umodhi3
        .text   00009f34    f4    f4 2**1  coff-h8300
[..\lib\libc.a]divsi3.o(overhead 6286 bytes)
                00009f64                ___modsi3
                00009f74                ___udivsi3
                00009f80                ___umodsi3
                00009f90                ___divsi3
        .text   0000a028    34    34 2**1  coff-h8300
[..\lib\libc.a]memchr.o(overhead 6330 bytes)
                0000a028                _memchr
        .text   0000a05c    5a    5a 2**1  coff-h8300
[..\lib\libc.a]memmove.o(overhead 6932 bytes)
                0000a05c                _memmove
 from *(.strings)
                0000a0b6 000020b6 _etext =.

.data           0000a0b6   118     0 2**1 load alloc reloc contents
 from *(.data)
        .data   0000a0b6    78    78 2**1  coff-h8300
[..\lib\libc.a]malloc.o(overhead 26266 bytes)
        .data   0000a12e     6     6 2**1  coff-h8300
[..\lib\libc.a]cvt.o(overhead 36960 bytes)
        .data   0000a134    9a    9a 2**1  coff-h8300
[..\lib\libc.a]findfp.o(overhead 19654 bytes)
                0000a13c                ___sF
                0000a1c8                ___sglue
                0000a1ce 00000118 _edata =.

.bss            0000a1ce   344     0 2**1 alloc
                0000a1ce 00000000 _bss_start =.
 from *(.bss)
        .bss    0000a1ce     2     2 2**1  coff-h8300
[..\lib\libc.a]syscalls.o(overhead 19350 bytes)
        .bss    0000a1d0   33c   33c 2**1  coff-h8300
[..\lib\libc.a]findfp.o(overhead 19654 bytes)
 from *(COMMON)
        COMMON  0000a50c     6     6 2**1  coff-h8300
[..\lib\libc.a]findfp.o(overhead 19654 bytes)
                0000a50c                ___sdidinit
                0000a50e                ___cleanup
                0000a510                _errno
                0000a512 00000344 _end =.

.stack          0000f000     0     0 2**1
                0000f000 00000000 _stack =.
 from *(.stack)
LOAD command line
```

Listing 9. **Test.map** (end)

Listing 10. **Test.ld** (begin)

```
OUTPUT_ARCH(h8300)
OUTPUT_FORMAT(srec)

SECTIONS
{
        .text 0x8000 : { *(.text) _etext = . ; }
        .data  .     : { *(.data) _edata = . ;}
        .bss   .     : { *(.bss)  *(COMMON) _end = . ; }
        .stack 0xf800: { *(.stack) }
}
```

Listing 10. **Test.ld** (end)

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300