HITACHI SEMICONDUCTOR TECHNICAL UPDATE

Classification of Production	Development Environment			No	TN-CSX-041A/E	Rev	14
THEME	Release of patch against SuperH RISC engine C/C++ compiler Package Ver.6.0, Ver.6.0A and Ver.6.0B		Classification of Information	1. Spec change 2. Supplement of Documents 3. Limitation of Use 4. Change of Mask 5. Change of Production Line			
PRODUCT NAME	P0700CAS6-MWR P0700CAS6-SLR P0700CAS6-H7R	Lot No.		SuperH RISC engine C/C++ Compiler, Assembler, Optimizing Linkage Editor User's Manual ADE-702-246 Rev.1.0		Effective Date	
		All	Reference Documents			Eternity	

SuperH RISC engine C/C++ compiler Package is updated to Ver.6.0C.

A user who has the following product should be informed.
SuperH RISC engine C/C++ compiler Package Ver. 6.0, Ver. 6.0R1, Ver.6.0A, Ver.6.0AR1,Ver.6.0AR2 or Ver.6.0B.

Attached:
"SuperH RISC engine C/C++ compiler Package Ver. 6.0C Updates"
(P0700CAS6-020822E) , 6 page

SuperH RISC engine C/C++ compiler Package Ver. 6.0C Updates

1. Compiler (Ver.6.0B -> Ver.6.0C)

1.1 Illegal restoring order of registers

[Description]
Solved the problem that the restoring order of the registers, which have been saved in the function prolog, may be illegal if a function ends with a function call, and if the cpu=sh2e or sh4 option is specified.

[Conditions]
The problem may occur when all of the following conditions are satisfied.
(1) The optimize=1 option is specified.
(2) The cpu=sh2e or sh4 option is specified.
(3) A function calls another function at the end of itself.
(4) One or more floating-point registers are saved and restored.
(5) The function calls a function, which can be a runtime routine, other than the function called at the end (refer to (3)).

[Version to be updated]
Ver.6.0B

1.2 Illegal result of execution when pic = 1 is specified

[Description]
Solved the problem that the result becomes illegal when executing a program that calls externally defined functions while pic = 1 is specified.

[Conditions]
This problem may occur when all of the following conditions are satisfied;
(1) A CPU other than SH1 is specified.
(2) The pic = 1 option is specified.
(3) An externally defined function is called in a program.

[Version to be updated]
Ver.6.0B

1.3 Illegal allocation of the PR register

[Description]

Solved the problem that the object becomes illegal when the instruction, which is included in a function, to save or restore the register data is only available for the PR and FPU registers while cpu = SH2E or SH4 is specified.

[Conditions]

This problem may occur when all of the following conditions are satisfied;

(1) The optimize = 1 option is specified.

(2) SH2E or SH4 is specified as the CPU.

(3) Saving and restoration of register data are only available for the PR and FPU registers.

(4) The #pragma interrupt option is not specified for the functions concerned.

[Version to be updated]

Ver.6.0B

1.4 Code error

[Description]

Solved the problem that an internal error from 6001 to 6007 occurs when compiling the source program that includes codes in a comment, in which the first byte is '0x8e' and the second byte is '0xa1' to '0xdf', while the listfile option is specified.

[Version to be updated]

Ver.6.0,Ver.6.0A or Ver.6.0B

1.5 Error with an illegal character string

[Description]
Solved the problem that error 2138 is output when compiling the following source program in the English version of the operating system by using the C++ language;

[Source program]
```
 char b[] = "\xec""\x01";
```

[Version to be updated]
Ver.6.0,Ver.6.0A or Ver.6.0B

1.6 Illegal allocation of sections

[Description]
Solved the problem that switching of sections is prevented when a definition, for which the section has been changed, comes after a declaration as shown below.

[Source program]
```
 extern int a;

 #pragma section OTHER
 int a = 1;
```

[Version to be updated]
Ver.6.0,Ver.6.0A or Ver.6.0B

1.7 Countermeasure for a failure in the FDIV chip

[Description]
The failure regarding the SH7055RF FPU FDIV has been fixed. When the cpu = SH2E and extra = a = 2000 options are specified, an object will be output, for which the failure mentioned above has been prevented.

[Version to be updated]
Ver.6.0,Ver.6.0A or Ver.6.0B

2. Optimizing Linkage Editor (Ver.7.0A -> Ver.7.1.06)

2.1 Incorrect debug information caused by the rename option
  The problem in which the debug information of the symbols in the renamed
  section is deleted with the form=relocate option is fixed.

 [Conditions]
  If both of the following, (1) and (2), are satisfied, the problem occurs.
  (1) The form=relocate option is specified.
  (2) The rename option is specified.

2.2 Invalid optimization of constant or literal data
  The problem in which symbols are incorrectly unified with the
  optimize=string_unify is fixed.

 [Conditions]
  If both of the following, (1) and (2), are satisfied, the problem occurs.
  (1) A C source file is compiled with the goptimize option.
  (2) The optimize=string_unify option is specified to the optimizing
     linkage editor.

2.3 Incorrect section attribute when a binary file is input
   Fixed is the problem in which section attribute is illegal when the following
   conditions are satisfied at the same time.

 [Conditions]
    (1) Input both an object file and a binary file.
    (2) A section whose size is zero is defined in the input object file.
    (3) The section whose size is zero is specified in the binary option.
    (4) The object file of (2) is input earlier than the binary file of (3).

2.4 Internal error caused by the form=relocate option

Fixed is the problem that occurs when all of the condition (1) to (4) are satisfied.

[Conditions]

(1) The first input object file is compiled with the goptimize option.

The second or later input file is either compiled without the goptimize option

or just assembled.

(2) The form=relocate option is specified.

(3) The profile option is specified.

(4) The optimize option is specified.

2.5 Incorrect literal referring

Fixed is the problem in which a referred literal value is incorrect when the following

conditions are satisfied at the same time.

[Conditions]

(1) The goptimize option has been specified in an input object file.

(2) More functions than one refer to the same literal in the object file.

(3) The optimize=register option is specified.

2.6 Incorrect optimization with partial suppression of the optimization

Fixed is the problem in which the partial suppression of the optimization does not work.

The problem occurs when the following conditions are satisfied at the same time.

[Conditions]

(1) The input file is compiled with the goptimize option.

(2) The nooptimize option is not specified.

(3) The absolute_forbid option is specified.

(4) More than one address ranges are specified to the absolute_forbid option.

(All but the first becomes ineffective)

2.7 Incorrect object code in generating a relocatable file

Fixed is the problem in which an incorrect object code is generated when the following conditions are satisfied at the same time.

[Conditions]
    (1) The input file is relocatable file.
    (2) The form=relocate option is specified.
    (3) The delete or rename option is specified.

2.8 Illegal object when save/restore registers optimization is specified

Fixed is the problem in which an illegal object code is output when optimization of codes for saving or restoring registers is specified and either of the following conditions is satisfied:
    (1) No literal exists in the final function in the file.
    (2) There is a branch destination immediately after the instruction for restoring registers.
    (3) The SUBC instruction is included in the function to be optimized.

2.9 Fixed internal errors

Fixed the following internal errors.
(1703),(1704),(3304),(7041),(7707)

3. Format converter (1.0B -> 1.0.04)

3.1 Unrecognized input files

Fixed is the problem in which the format converter can not recognize input files when input files exist in the folder with compress attribute.