Document No.: D-SLD-CMN-0008-01

Command IF generator User manual

ver5.0-2

Renesas System Design
Design Integration Department

RENESAS CONFIDENTIAL

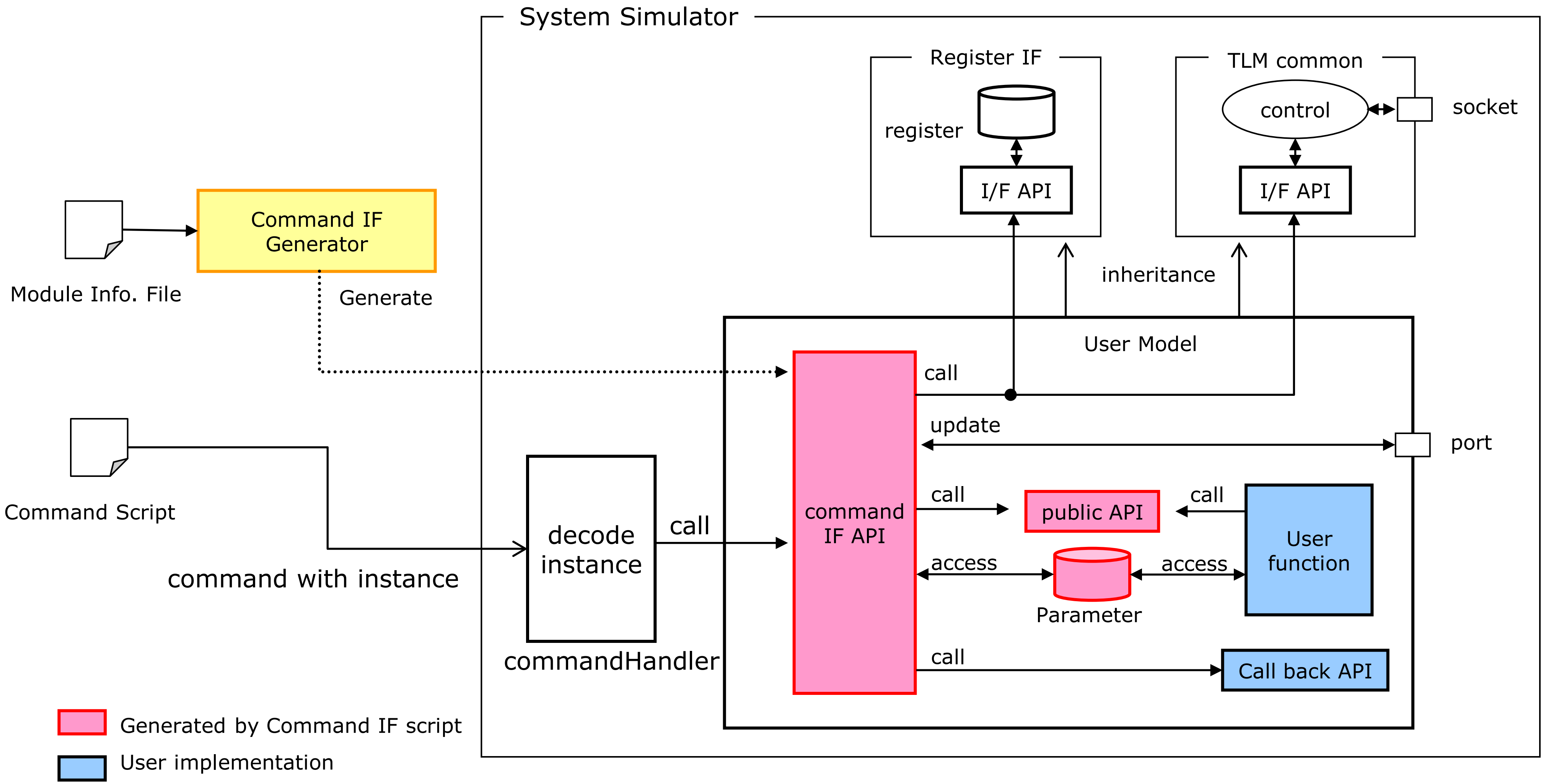
Summary/Feature

Command IF is one of model IF whose purpose is to extract a model inner data and update it for the sake of the debug, and Command IF generator makes that C++ code from a user defined command file.

- Generate C++ code to access to variable and function which are access from command IF
- Verify an access condition to user defined variable / function
- Generate not only user defined command but also some standard commands
- Enable to access to a model’s port
- Enable to access to a model’s register with Register IF generator

© 2012-2015 Renesas System Design. All rights reserved.

Summary/Block diagram

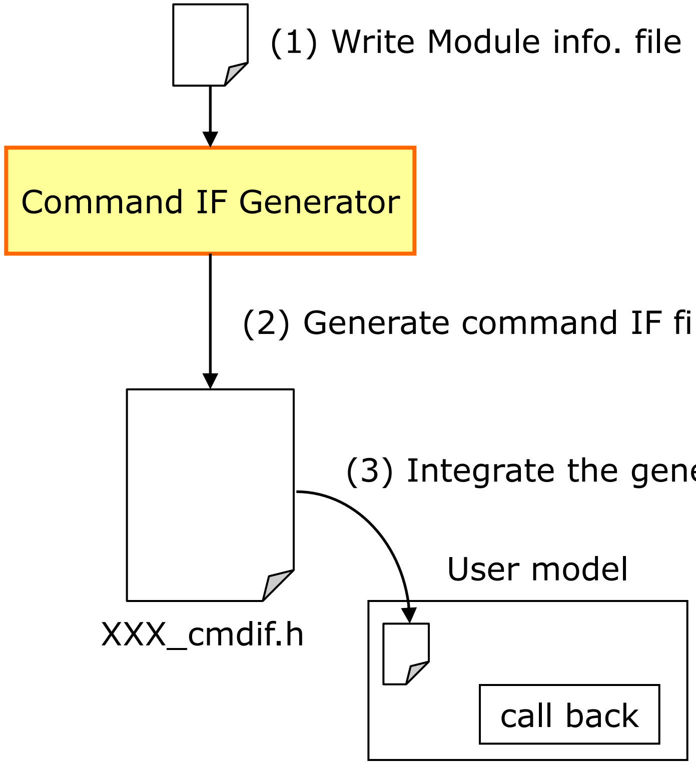


🟥 Generated by Command IF script

🟦 User implementation

© 2012-2015 Renesas System Design. All rights reserved.

Summary/Design flow


(1) Write Module info. file
Command IF Generator
(2) Generate command IF file
XXX_cmdif.h
(3) Integrate the generated file
User model
call back

(4) Implement a call back API

Example of Module info. file

```
%MODULE Cintc
%CMD_RW EnableMsg <arg:bool=false> @ Debug message on/off
```

Example of user code

```
Cintc : public sc_module {
#include “intc_cmdif.h”
...
```

(5) Access to defined command in command script

Example of command script

```
% intc EnableMsg            → Get current value
false
% intc EnableMsg true       → Set current value
```

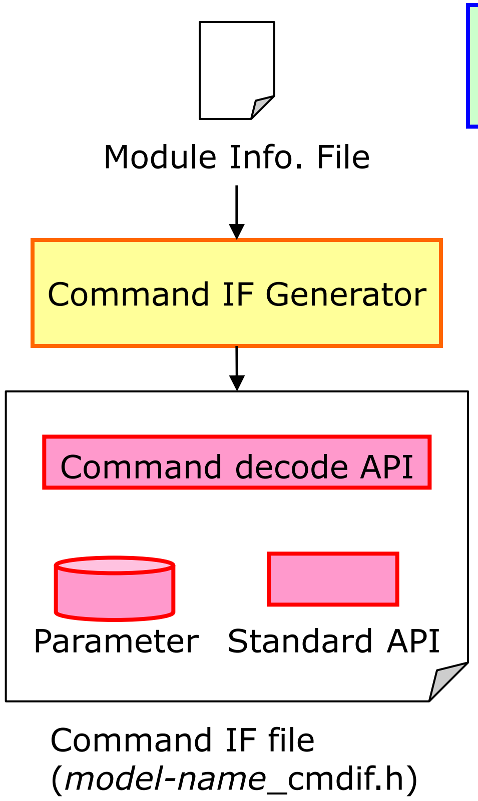
© 2012-2015 Renesas System Design. All rights reserved.

Basic usage

- **Parameter RW command**
- Action command
- Suppress standard command
- Parent class ID
- Command prefix
- Register/Port
- Hardware break
- Python IF

© 2012-2015 Renesas System Design. All rights reserved.

Usage of Command IF Generator



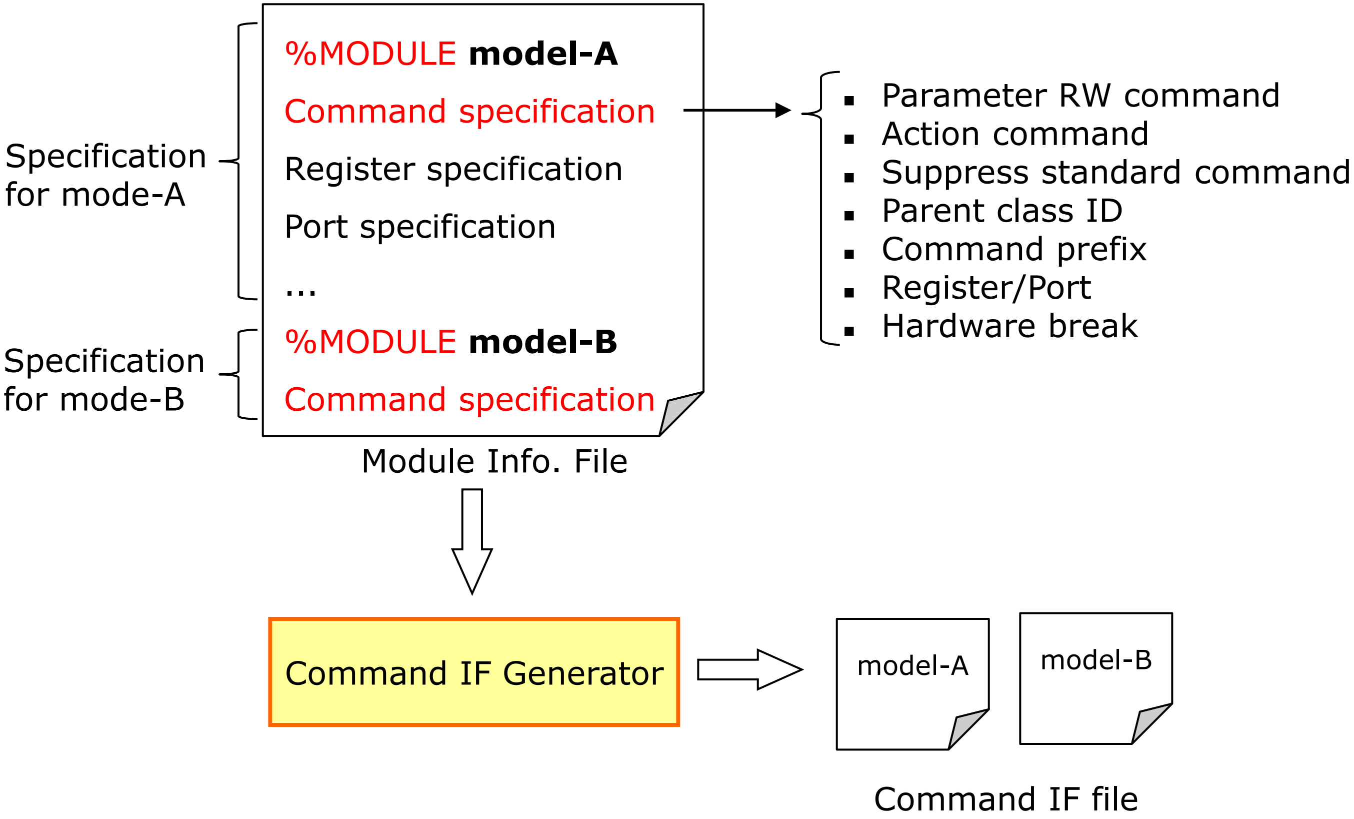
```
python3 gen_cmdif.py
  <Module Info. File> [-h/--help] [-v/--version] [-p/--pythonif] [--sample]
```

- Command IF generator works on Python version 3
- Generate Command IF file from Module Info. file which includes command specification
- Enable to specify 1 or more than Module Info. file and generate each command files.
- Eliminate “C” keyword at the beginning of the model name
- Generated file name is composed of model-name + cmdif.h
- Dump a command usage by -h/--help option and dump a generator version by -v/--version option.
- Dump Python I/F file when -p/--pythonif is specified.
- Dump sample code of Module Information file when –sample is specified.

© 2012-2015 Renesas System Design. All rights reserved.

Formula of Module Info. File-1 (1/4)

- At first, define a model name and follows 0 or more than command specification. As command IF generator skips to read other commands, Module Info. file enables to include other generator’s command too.
- Model name is specified after %MODULE attribute.



© 2012-2015 Renesas System Design. All rights reserved.

Formula of Module Info. File-1 (2/4)

Parameter RW command

```
%CMD_RW  command-name  (command-argument)+   @ usage
```

- Summary
  - Define a command to write/read a parameter

- Description rule
  - %CMD_RW attribute follows a command name, arguments and an usage.
  - There are must / need type arguments. The former is surrounded by <> and the latter is surrounded by []. First argument should be “must” type.
  - Detailed specification is explained in the next slides

- Generated code
  - Declare data member whose name is specified command. If two or more than arguments are specified, the struct type data member is declared.
  - Initialize the data member in “CommandInit” function which should be called in constructor
  - Generate a command IF to access to the parameter
  - Command with the argument is write action, and command without the argument is read action.
  - Verify a specified command argument in command script.

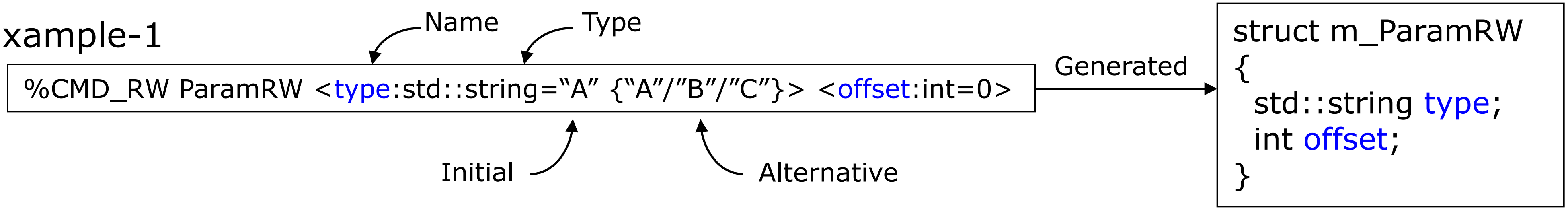
© 2012-2015 Renesas System Design. All rights reserved.

Formula of Module Info. File-1 (3/4)

Command argument `Name : Type = Initial {Alternative}`

Item	Required	Explanation
Name	Must	If the number of argument is 1, data member’s name is command name and the argument name is not used. If the number of argument is more than 2, struct type variable is generated whose name is command name and whose factor name is argument name.
Type	Must	Declare a variable of the type. The following type is available. (unsigned/signed)char/short/int, double, sc_time, (std::)string, enum, uint64_t, sc_dt::uint64 Note: sc_time’s unit is always SC_NS.
Initial	Must	Set it to an initial value in constructor
Alternative	If needs	List of acceptable factors which are separated by /. If a type is enum, this is always needed. “+” means positive value and “-” means negative value.

Example-1



```
%CMD_RW ParamRW <type:std::string=“A” {“A”/”B”/”C”}> <offset:int=0>
```

```
struct m_ParamRW
{
  std::string type;
  int offset;
}
```

© 2012-2015 Renesas System Design. All rights reserved.

Formula of Module Info. File-1 (4/E)

Example-2

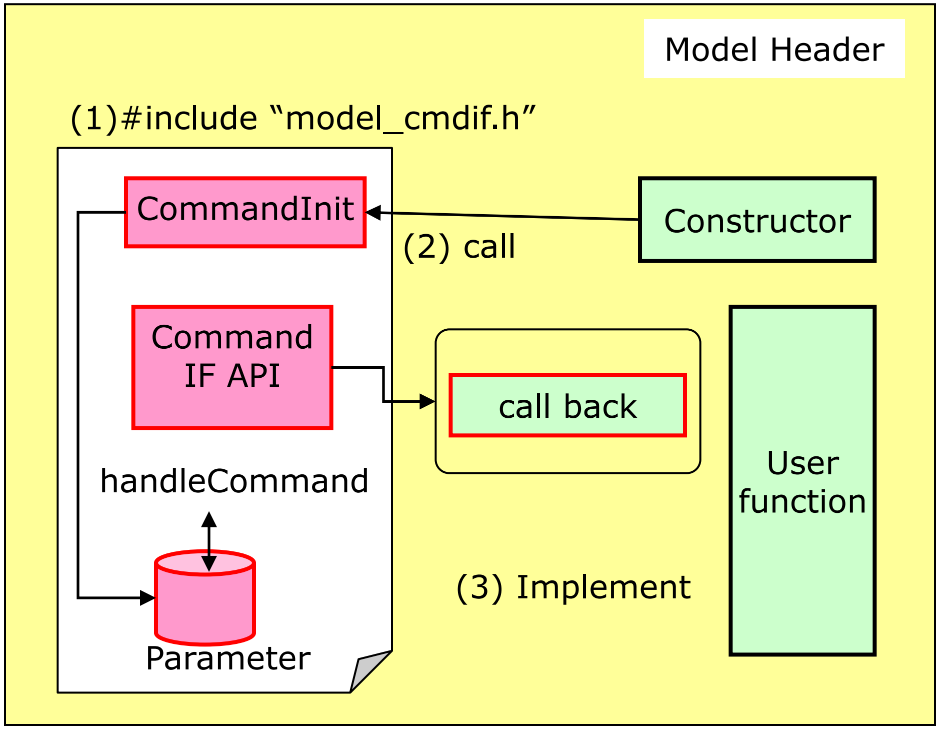
Example	Generated variable	Initial	Input check
cmd <A:int=0>	int cmd;	0	Is it number ?
cmd <A:bool=false>	bool cmd;	false	Is it true or false ?
cmd <A:sc_time=0>	sc_time cmd;	0	Is it double ?
cmd <A:std::string=“X” {“X”/”Y”/”Z”}>	std::string cmd;	X	Is it A, B or C ?

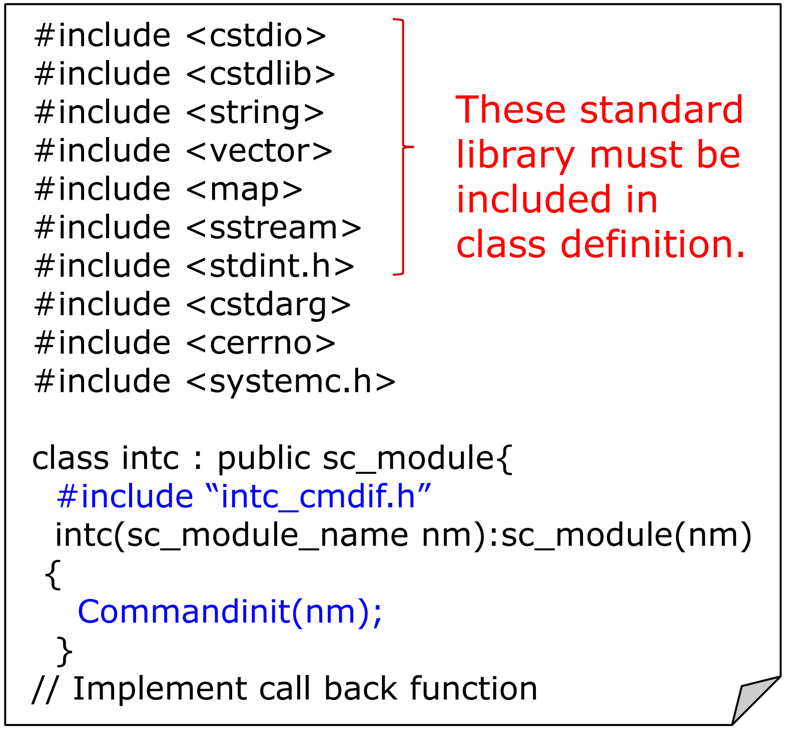
cmd <A:int=0> [B:std::string=“X”]	struct m_cmd {
	    int A;
	    std::string B;
	}
		A : 0
		B : X
			Is A number ?
			Is B string ?
			If the 2nd argument “B” is not specified, “B” is current value.

© 2012-2015 Renesas System Design. All rights reserved.

Integration into a user code

1. Include some standard library which command IF API uses. The information about the included standard library will be dumped when executing generator.
2. Include the generated file in class definition
3. Call CommandInit API with the model’s instance name which initializes the parameter variables.
4. Implement the call back functions which are explained in next slide.





```
#include <cstdio>
#include <cstdlib>
#include <string>
#include <vector>
#include <map>
#include <sstream>
#include <stdint.h>
#include <cstdarg>
#include <cerrno>
#include <systemc.h>

class intc : public sc_module{
  #include “intc_cmdif.h”
  intc(sc_module_name nm):sc_module(nm)
 {
    Commandinit(nm);
  }
// Implement call back function
```

These standard library must be included in class definition.

Note: Command IF file defines private attribute.

© 2012-2015 Renesas System Design. All rights reserved.

Call back function-1

No.	Name	Explanation
1	General call back	**string CommandCB(const vector<string> args)**
		This function is always called when parameter RW command is used. Write an relative action in this function if needed.
		argument 1: a command list without an instance name
		return : Return a message string. If an error occurs, it should include “error” keyword
2	Dump profile	**bool DumpProfile(FILE *p_file = stdout, sc_time_unit time_unit = SC_NS)**
		Dump a profile information
		argument 1: File pointer to dump the profile argument 2: Time unit
		Return: true => succeed to dump, false => fail to dump.
3	Clear profile	**void ClearProfile()**
		Initialize an internal profile information. In this function, clear all the variables relative to profile information in a user code.
4	Reset	**void AssertReset(double start_time , double period)**
		Reset a target model. Initialize all the internal information from the function
		argument 1: Reset start time. The unit is always SC_NS. argument 2: Reset period time from start_time. The unit is always SC_NS

RENESAS CONFIDENTIAL

© 2012-2015 Renesas System Design. All rights reserved.

Command usage

- A command consists of an instance name and the command with/without the arguments.
- A command has write/read action. If the command with the arguments works as write action. On the other hand, the command without the arguments works read action.
- Example

```
% top.X param1 10      <= Write “10” to “param1”
% top.X param1         <= Return current “param1”’s value
10
```

© 2012-2015 Renesas System Design. All rights reserved.

Advanced usage

- Parameter RW command
- **Action command**
- **Suppress standard command**
- **Parent class ID**
- **Command prefix**
- Register/Port
- Hardware break
- Python IF

© 2012-2015 Renesas System Design. All rights reserved.

Formula of Module Info. File-2 (1/3)

Action command

```
%CMD_ACTION  command-name  (command-argument)*  @ usage
```

- Summary
  - Call a user defined function whose name is same as command name

- Description rule
  - %CMD_ACTION attribute follows a command name, some arguments and an usage. The number of arguments is more than 0.
  - There are must / need type arguments. The former is surrounded by <> and the latter is surrounded by []. First argument should be “must” type.
  - Detailed argument specification is same as Parameter RW command’s. And only need type argument has the initial value.

- Generated code
  - Call a specified function from command IF code
  - Set default value to an argument who type is need, when a user does not specified it
  - Verify a specified command argument in command script.

© 2012-2015 Renesas System Design. All rights reserved.

Formula of Module Info. File-2 (2/3)

Suppress standard command

```
%CMD_NODUMP_API  (standard-command)+
```

- Summary
  - Suppress to generate specified standard command.
- Description rule
  - Specify standard command name to %CMD_NODUMP_API attribute.
  - Insert a space between each standard command name
  - Refer to “standard command” slide to get detailed command name.
- Generated code
  - Not generate specified standard command

© 2012-2015 Renesas System Design. All rights reserved.

Formula of Module Info. File-2 (3/E)

Example-1 (Action command)

Definition example	Called function
%CMD_ACTION cmd <A:int=0> : void @ usage	void cmd (int A);
%CMD_ACTION cmd <A:bool=false> : bool @ usage	bool cmd(bool A);
%CMD_ACTION cmd <A:int> <B:string> : void @ usage	void cmd(int A, string B)
%CMD_ACTION cmd [A:bool=true] : bool @ usage	bool cmd(bool A);

Example-2 (Suppress standard command)

Definition example	Explanation
%CMD_NODUMP_API DumpProfile ClearProfile	Control DumpProfile and ClearProfile command generation. A user doesn’t need to implement them.

© 2012-2015 Renesas System Design. All rights reserved.

Standard command (1/2)

Type-A : Action command
Type-P : Parameter RW command

Type	Name	Explanation
A	help	**help**
		Dump a model’s command list with that explanation
		Return:Command list with that explanation
A	DumpProfile	**DumpProfile [file_name:string=stdout] [time_unit:sc_time_uint=SC_NS] : string**
		Dump profile information. A user must implements this API
		argument 1:Output file name (Note: “stdout” is treated as Standard Output not “stdout” file) argument 2:Time unit
		Dump profile information. A user must implements this API
A	ClearProfile	**ClearProfile**
		Initialize the profile internal information. A user must implements this API.
A	AssertReset	**AssertReset <start_time> <period>**
		Reset a target model
		argument 1: Reset start time. The unit is always SC_NS. argument 2: Reset period time from start_time. The unit is always SC_NS

© 2012-2015 Renesas System Design. All rights reserved.

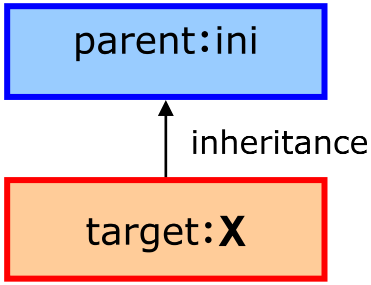
Standard command (2/E)

Type	Name	Explanation
P	MessageLevel	**MessageLevel <string=fatal|error {fatal|error|warning|info}>**
		Specify the message level. 1 or more than levels can be specified at once, and they are separated by |
		Argument : Message Level
		Return : write-mode => NULL, read-mode => current value
P	EnableInsertInput	**EnableInsertInput <bool=false>**
		Dump a specified command or not
		Argument : true => Dump a specified command, false => Suppress to dump it
P	EnableDumpResult	**EnableDumpResult <bool=false>**
		Dump a command result or not
		Argument : true => Dump a result, false => Suppress to dump it

© 2012-2015 Renesas System Design. All rights reserved.

Combination with hierarchal class

- Command IF has a solution to access to a parent class's command
- A specified command is transferred to not only a target model but also all the inheritance classes.
  - **Example-1)** top.X help <= Send "help" command to both X and ini
- If you want to send a command to specified model, insert "model ID" to command name. "model ID" is defined by a user and it should be unique in a target model. "command" which is one of "model ID" is reserved and it means a target model.
  - **Example-2)** top.X command help <= Send "help" command to just X
  - **Example-3)** top.X ini help <= Send "help" to just "ini"
- If a specified command is not defined, command IF's action is as follows.


parent:ini

inheritance

target:**X**

has a command		Read	Write
target	parent		
No	No	Error	Error
No	Yes	return parent's message	apply to just parent
Yes	No	return target's message	apply to just target
Yes	Yes	return both messages	apply to both

© 2012-2015 Renesas System Design. All rights reserved.

Formula of Module Info. File-3 (1/2)

Parent class ID

```
%CMD_BASEID  (model-ID)+
```

- Summary
  - Make valid of parent’s model ID. A parent class must has its model ID and a model inherits it must specify the model ID in this attribute.
- Description rule
  - Specify 1 or more than parent’s model ID.
  - Separate each model IDs by a space
  - Common model has its model ID. Refer to the following table
- Generated code
  - Generate a IF code to access to specified parent command
- Example
  - %CMD_BASEID ini tgt

© 2012-2015 Renesas System Design. All rights reserved.

Formula of Module Info. File-3 (2/E)

Command prefix

```
%CMD_ID  model-ID
```

- Summary
  - Define its original model ID which is accessed by the child class.
  - This attribute should be used by a common class.

- Description rule
  - Specify 1 model ID which begins alphabet and follows alphabet or number.

- Generated code
  - Generate a model ID decode code

- Example
  - %CMD_ID gtimer

© 2012-2015 Renesas System Design. All rights reserved.

Reserved model ID

Common model	model ID
TLM common class / Initiator IF	ini
TLM common class / Target IF	tgt
Register IF class	reg
Port Interface	port

© 2012-2015 Renesas System Design. All rights reserved.

Other features

- Parameter RW command
- Action command
- Suppress standard command
- Parent class ID
- Command prefix
- **Register/port**
- **Hardware break**
- **Python IF**

© 2012-2015 Renesas System Design. All rights reserved.

Register control (1/2)

- Command IF has a solution to access to a Register IF class's command.
- Register IF class supports to read/write command.

Set up

1. Inherit Register IF class
2. Add "**reg**" model-ID to %CMD_BASEID.

Register access command

- Insert "reg" command-ID to command name.
- Without specifying any command, dump all the register names to standard output.

© 2012-2015 Renesas System Design. All rights reserved.

Register control (2/E)

Type	Name	Explanation
A	-	**-**
		If a user does not specify any command name, dump all the register names
		Return:All the register name
P	(register name)	**(register name) [unsigned int]**
		Update / refer to a register. It is possible to use wild card to specify the register name
		argument 1: Written data
		Return : Read data when an argument is not specified
A	force	**(register name) force <unsigned int>**
		keep a specified value. SW never update the register value until “release” command specifies
		argument 1 : forced value
A	release	**(register name) release**
		Clear a “force” command

Port control (1/3)

- Command IF has a solution to access to a target model’s output port.

Set up

1. Add “**port**” model-ID to %CMD_BASEID.
2. Specify port information by %PORT attribute. See next slide too.

Port access command

- Insert “port” command-ID to command name.
- Without specifying any command, dump all the port names to standard output.

Limitation

- Writing to port before starting simulation is prohibited, it may cause segmentation fault.

© 2012-2015 Renesas System Design. All rights reserved.

Port control (2/3)

Port

```
%PORT  <name> <size> <direction> <type> <initial-value>
```

- Summary
  - Define a port specification to access from command IF. Command IF enables to read both input and output ports, and enables to write only output ports.

- Description rule
  - Specify port name, size direction type and initial value. “initial value” is required only if the direction is “out”. “direction” should be “in”, “out” or “inout” and type should be “bool”, “sc_uint”, “sc_biguint”, “double”, “uint64_t” or “sc_dt::uint64”. This formula is shared with other generators and command IF does not refer to initial value.

- Generated code
  - Generate a port access code

- Example
  - %PORT o1_port 8 out sc_uint 255
  - %PORT o1_port 64 in double 255

© 2012-2015 Renesas System Design. All rights reserved.

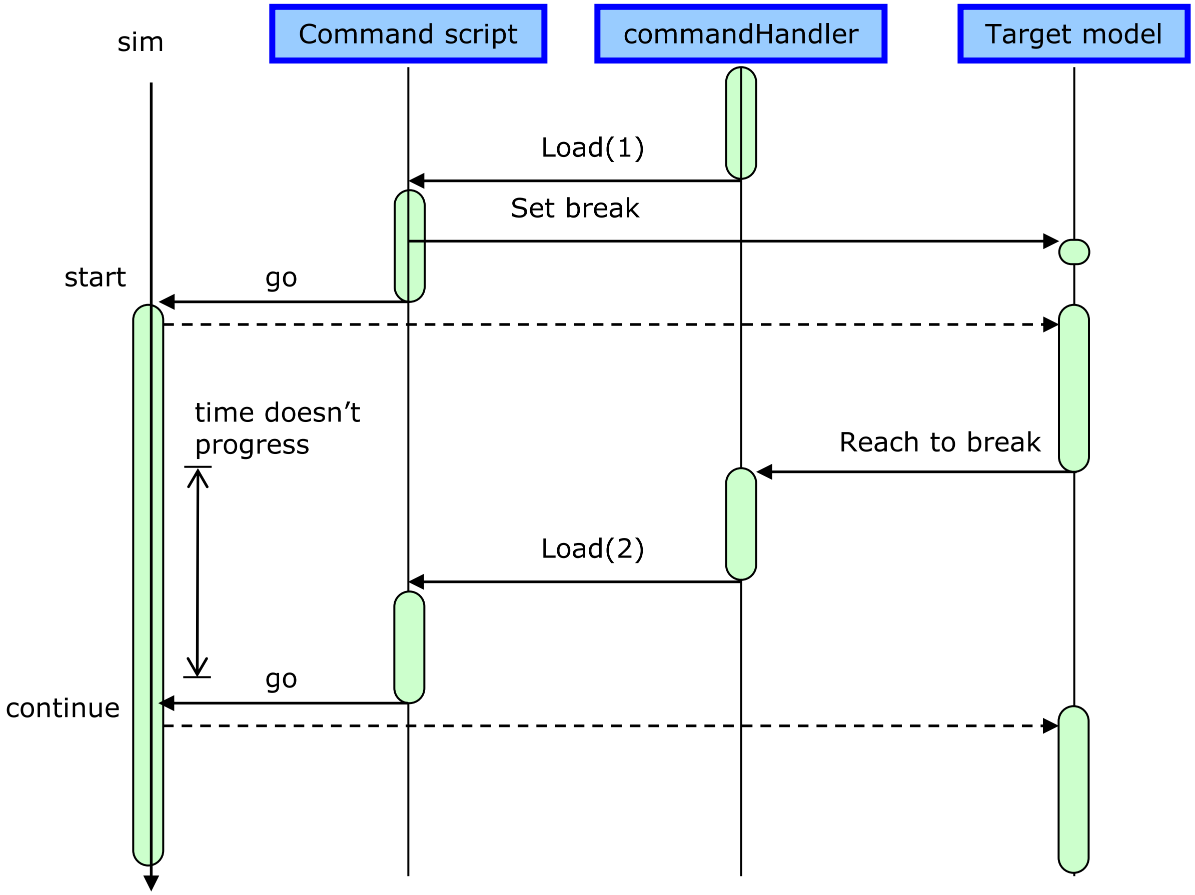
Port control (3/E)

Type	Name	Explanation
A	-	**-**
		If a user does not specify any command name, dump all the port names
		Return:All the register name
P	(port name)	**(port name) [unsigned int]**
		Update / refer to a port. It is possible to use wild card to specify the port name
		argument 1: Written data (only output port)
		Return : Read data when an argument is not specified

© 2012-2015 Renesas System Design. All rights reserved.

Hardware Break (1/4)

- A target model enables to stop the simulation and moves to command IF with commandHandler.



© 2012-2015 Renesas System Design. All rights reserved.

Hardware Break (2/4)

How to use ?

1. Register break-names by %CMD_HWBRK attribute in Module Info. file. break-names are used in hw_break function’s argument.

   Example-1）%CMD_HWBRK debug1 debug2

2. Insert “hw_break” function in a model source code where hardware break should be triggered. “hw_break” follows one of registered break-names.

   Example-2） if (condition) hw_break(“debug2”);

3. Set break-names by HWBreak command in Command script.

   Example-3） top.X HWBreak debug1|debug2

© 2012-2015 Renesas System Design. All rights reserved.

Hardware Break (3/4)

Hardware break

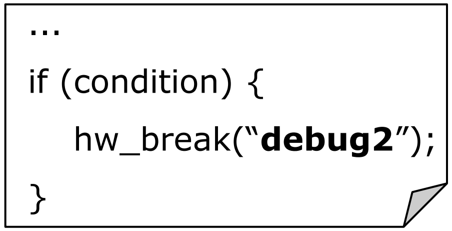
```
%CMD_HWBRK  (break-name)+
```

- Summary
  - Register specified break names which is used in hw_break function’s argument.

- Description rule
  - Specify 1 or more than break name. Insert a space between break-names.

- Generated code
  - Generate a hardware break code and register the specified break-name.

- Example
  - %CMD_HWBRK debug1 debug2



```
…
if (condition) {
    hw_break(“debug2”);
}
```

User model code

When this condition is valid, hw_break function is called. In this function, check whether “debug1” is registered by %CMD_HWBRK or not. If it is registered, command IF script is continued to load.

© 2012-2015 Renesas System Design. All rights reserved.

Hardware Break (4/E)

Type	Name	Explanation
P	HWBreak	**HWBreak <break-names>**
		Enable specified break-names
		Argument 1: break-name. Insert “|” between break-names
		Return : write mode => “”, read mode => valid break-names
A	go	**go**
		start/continue simulation

© 2012-2015 Renesas System Design. All rights reserved.

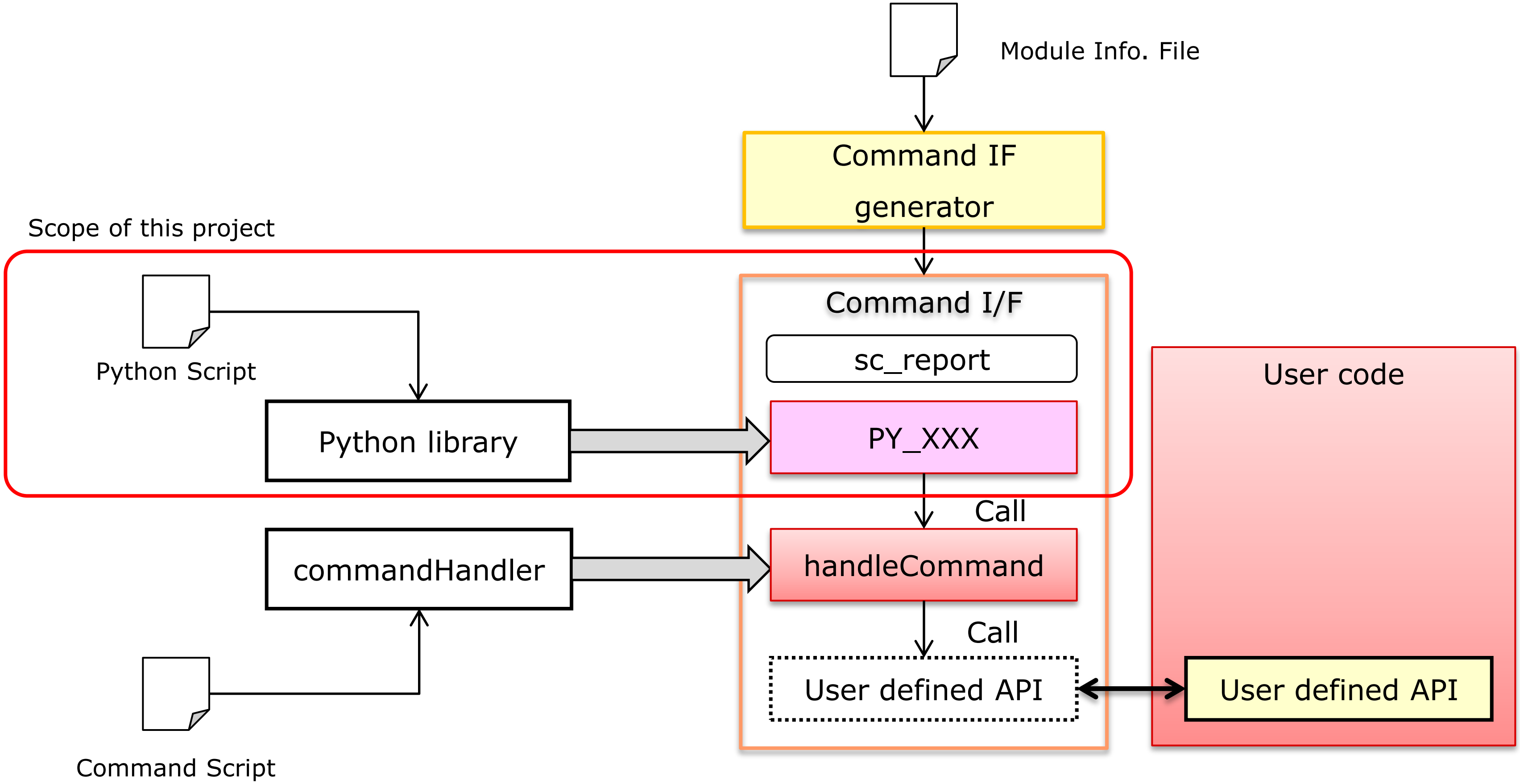
Public API

No	Explanation
1	**void CommandInit(string name=“”)**
	Initialize all the data member which are defined by parameter RW command and standard command
	argument 1 :instance name
2	**void re_printf(string group, const char *message, ...)**
	Only when a specified group is valid by MessageLevel command, dump the message. When REGIF_SC_REPORT macro is specified during a building binary, sc_report feature is available.
	argument 1 :message group argument 2 :message
3	**void WritePort(std::string port_name, T port_val)**
	Write specified value into specified port of which port name is defined by Command IF. (Please take a note that the type of written value must match the port type)
	argument 1 :name of port argument 2 :written value

© 2012-2015 Renesas System Design. All rights reserved.

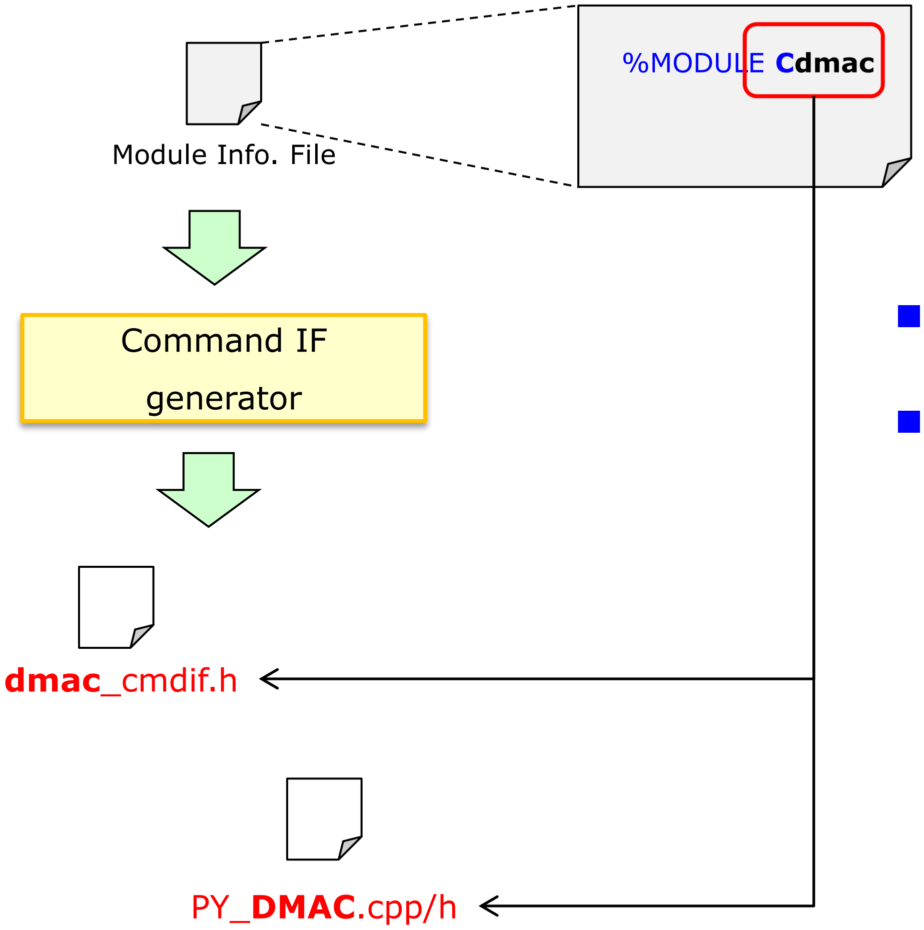
Python IF (1/2)

- Add-on Pyhton I/F which is used in M40 project by -p/--pythonif option.



© 2012-2015 Renesas System Design. All rights reserved.

Python IF (2/E)



- A prefix of generated file is composed of a fixed keyword and an argument of %MODULE attribute.
- The prefix of generated Python I/F file is “PY_” followed by %MODULE attribute’s argument except for initial “C” keyword. The class name should be capital letter.

Python I/F file name

PY_(%MODULE).cpp/h

© 2012-2015 Renesas System Design. All rights reserved.

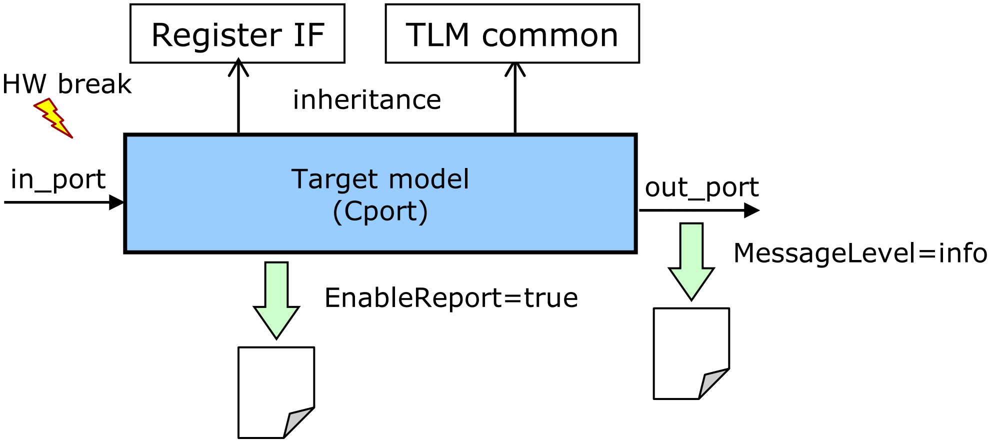
Example of use

© 2012-2015 Renesas System Design. All rights reserved.

Target specification

Design a command IF for Port class whose name is “Cport”.

1. Port class has some registers and a TLM target IF. Therefore port model inherits both Register IF and TLM common class.
2. Port class has an output port whose name is “out_port”. Port class dumps out_port value with current time when out_port value is changed and “**MessageLevel**” includes “info”.
3. Port class has an input port whose name is “in_port”, and when the number of in_port transformation reaches 50 or 100 times, Port class sets HW break.
4. Port class dumps a frequency report of in_prt and out_port transformation at the end of simulation if “**EnableReport”** is true.



Port class’s commands

No	Command
1	Register IF class’s commands
2	TLM common class target IF’s commands
3	MessageLevel
4	EnableReport
5	HWBreak

© 2012-2015 Renesas System Design. All rights reserved.

Module Info. file and user code

Module Info. file

```
%MODLE Cport
%CMD_BASEID reg tgt
%CMD_NODUMP_API  AssertReset DumpProfile ClearProfile CommandCB
 %CMD_RW EnableReport <bool = false> @ Dump a frequency of port transformation
%CMD_HWBRK time50 time100
```

User code (sub set)

```
#include <string>
#include <vector>
#include <map>
#include <cstdarg>
#include <cerrno>
#include <cstdio>
#include <systemc.h>

class Cport : public … {
#include "port_cmdif.h"
public:
    Cport(…  { // constructor
        CommandInit(this->name());
…
    ~Cport()  { // destructor
        if (EnableReport)  {
            printf(…  // report a frequency of all the ports.
```

```
// check the input port transformation
void count_up() {
    mCount ++;
    if (mCount >= 100) {
        hw_break("time100");
    }
    else if (mCount >= 50) {
        hw_break("time50");
…

// called when out_port is changed
void cb_out_port() {
    dbg_printf("info", "[%s] out_port = %d\n"
        , out_port.read().to_int()
        , sc_time_stamp().to_string().c_str());
…
```

© 2012-2015 Renesas System Design. All rights reserved.

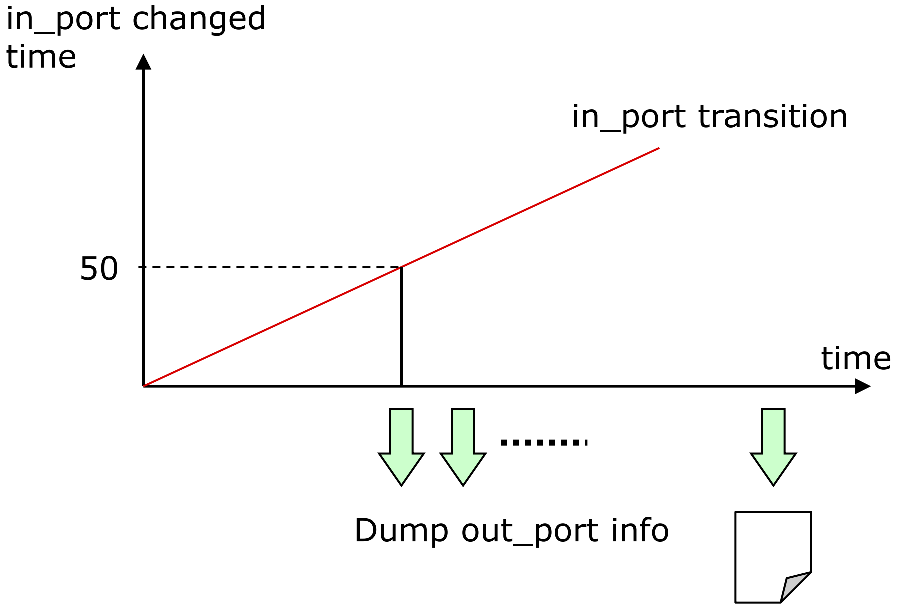
Generate command IF and prepare command script

Generate command IF file

```
% python3 gen_cmdif.py  port_cmdif.txt


    gen_cmdif.py --- 3.0

Loading model.txt ...

Success to run a script
% ls
port_cmdif.h   port_cmdif.txt
```

Command script

```
top.port EnableReport true

top.port HWBreak time50
# top.port HWBreak time100

go
top.port MessageLevel error|info
go
quit
```



- Port class dumps out_port information from the number of in_port transformation reachs 50 times.
- Finally Port class dumps a report.

Revision History

Rev.No	Contents	Agreed by Customer	Approved by RVC	Checked	Created
Ver2.0-1 Ver2.0-6	•基本仕様の作成	-		-	Watanabe '10/06/30 '10/07/28
Ver2.0-7	•記述順番の変更	-		Asano 10/09/21	Watanabe '10/09/09
Ver2.0-8	•変更履歴誤記修正 •[]の例をスライド10(例2)に追加 •MessageLevelプロトタイプ誤記修正	Asano 10/09/21		Asano 10/09/21	Watanabe '10/09/21
Ver2.0-9	•コマンドIFファイルの最後でprivate宣言していることを追記	-		-	Watanabe '10/09/24
Ver3.0-1	•Add phase 3 new features	-		Shibuya Suzuki Imoto Masuda	Watanabe '12/06/04
Ver3.0-2	•Feed back group review	Watanabe '12/06/06		-	Watanabe '12/06/06
Ver4.0-1	•Add Python IF	Watanabe '13/08/27		Imoto '13/08/26	Watanabe '13/08/26

© 2012-2015 Renesas System Design. All rights reserved.

Revision History

Rev.No	Contents	Agreed by Customer	Approved by RVC	Checked	Created
Ver5.0-1	•Slide6, Slide40: Remove defining PYTHONPATH. •Slide9: Support type uint64_t and sc_dt::uint64. •Slide11: Update notice about standard library. •Slide27: Add limitation for writing to port. •Slide28: Add more class type of port.		Vu Pham 02/13/15	Vu Pham 02/13/15	Son Tran 02/09/15
Ver5.0-2	•Update Format following latest company format. •Slide9: Add “signed” type. •Slide18: Add note for argument 1 of callback function DumpProfile. •Slide34: Add public API WritePort.		Vu Pham 03/19/15	Vu Pham 03/19/15	Son Tran 03/19/15

© 2012-2015 Renesas System Design. All rights reserved.