Verification Environment Generator Guide

System C Verification Environment : Verification Environment Generator Guide (ver 1.01)

Summary

The purpose of this designer manual is to describe the method to generate Verification Environment

Table of Contents

1.Introduction....4
2.Two kinds of Verification Environment....5
2.1.Stimulus Environment....5
2.2.SH4A Environment....6
2.3.Unsupported functionality....6
3.Verification-Environment-Generation Environment....7
4.Two steps to generate verification environment....9
4.1.Step 1 – Preparing configuration files....9
4.1.1.System Configuration file for verification environment....9
4.1.2.Configuration file for Stimulus model....15
4.2.Step 2 – Running gen_env.pl script....15
4.2.1.Block chart....15
4.2.2.Supported functions of gen_env.pl....17
4.2.3.Error messages....18
4.2.4.Usage....22
4.2.5.Example....23

Index of Tables

Table 3.1: The description of verification-environment-generation environment....8
Table 4.1: List of keywords in SCI file....10
Table 4.2: Supported functions of gen_env.pl....17
Table 4.3: Error message of gen_env.pl....21
Table 4.4: Options of gen_env.pl....22

Index of Figures

Figure 1.1: Scope of this guide in model verification task....4
Figure 2.1: Stimulus Environment....5
Figure 2.2: SH4A Environment....6
Figure 4.1: Structures of SCI file....9
Figure 4.2: A example of creating SCI file for verification environment....13
Figure 4.3: An example of SCI file for SH4A environment....14
Figure 4.4: Example SCI file of stimulus model (stimulusSCI.txt)....15
Figure 4.5: Stage 1 of the running flow of gen_env.pl....15
Figure 4.6: Stage 2 of the running flow of gen_env.pl....16
Figure 4.7: Stage 3 of the running flow of gen_env.pl....16
Figure 4.8: The working directory of the verification environment before executing gen_env.pl....23
Figure 4.9: System Configuration file for Stimulus Environment....24
Figure 4.10: Configuration file for Stimulus model....24
Figure 4.11: The env_rvc_set.csh script....25
Figure 4.12: The working directory of the verification environment after executing gen_env.pl....26

1. Introduction

Model verification is very important in model development because the quality of each model is based on the quality of verification. Primarily, the environment creation is a decisive step in a way that it affects the all next verification steps.

This document describe environment generator script that help tester or verifier create verification environment automatically and create Address_map_info.txt file.

The position of this guide in verification task is described in below :

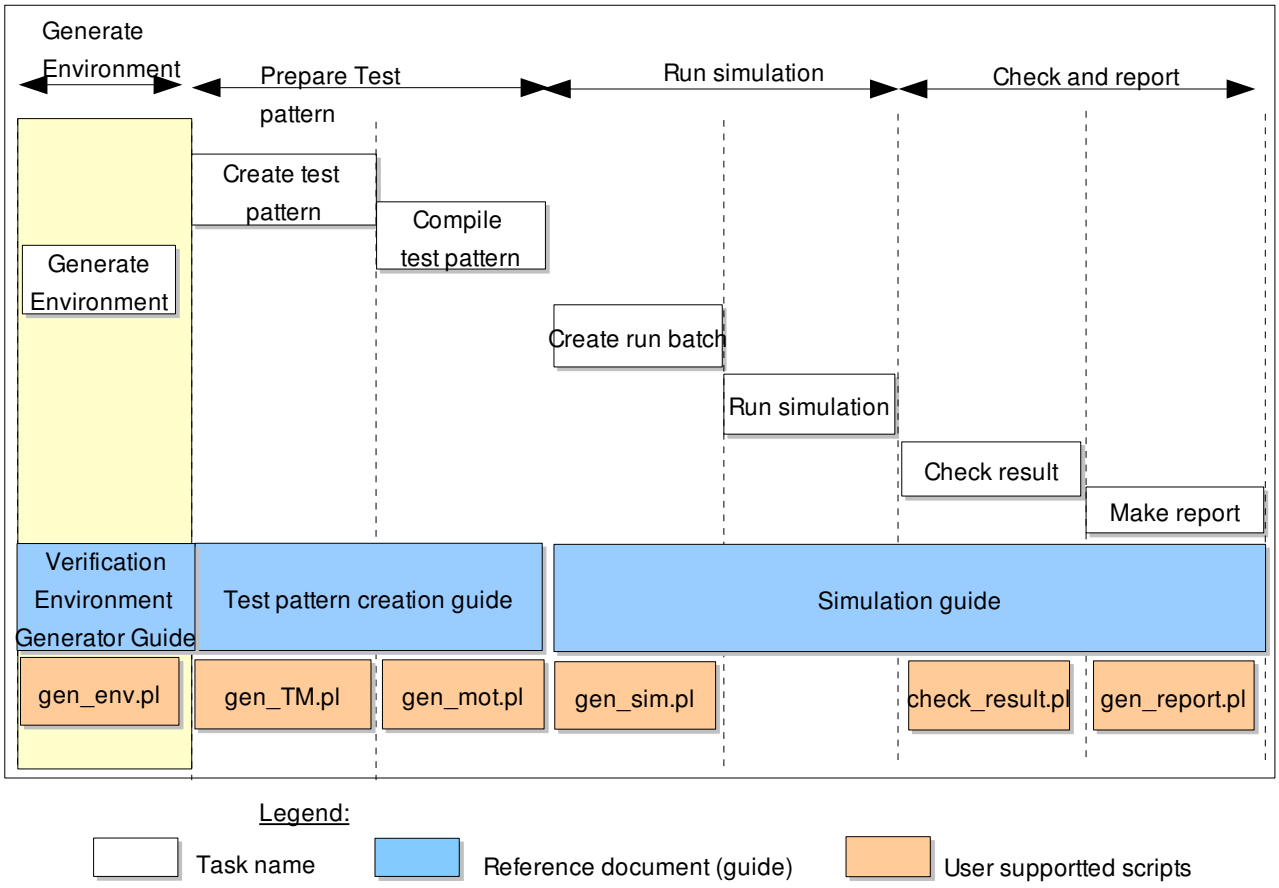


Figure 1.1: Scope of this guide in model verification task

1. Two kinds of Verification Environment

The environment generator help users create two types of environment

1.1.Stimulus Environment

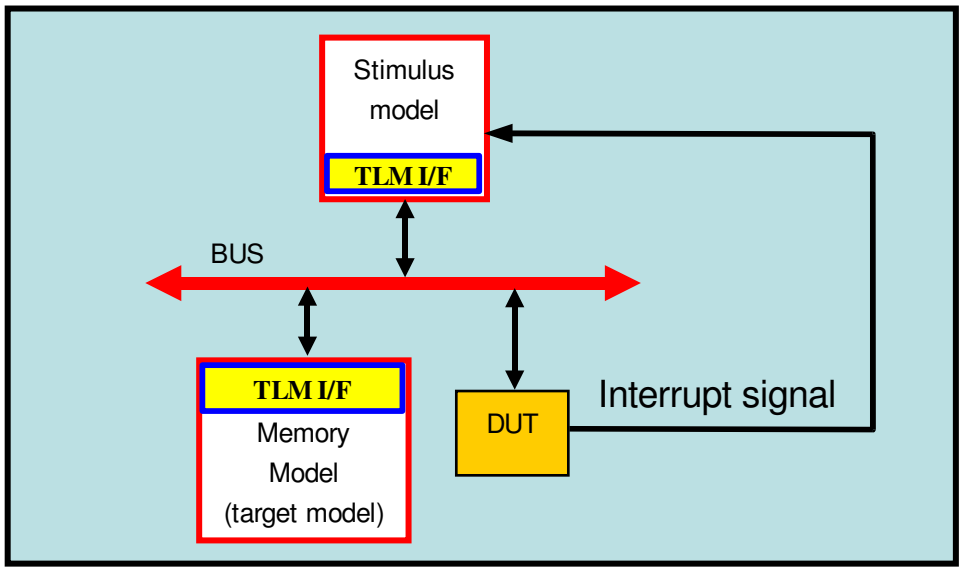


Figure 1.1: *Stimulus Environment*

Stimulus Environment includes three kinds of model:

- Initiator model is Stimulus model
- Bus model is simple bus connecting between Initiator and target models
- Target models are Memory model and DUT (Design Under Test)

All models are connected together according to Loosely-timed TLM interface

1.1.SH4A Environment

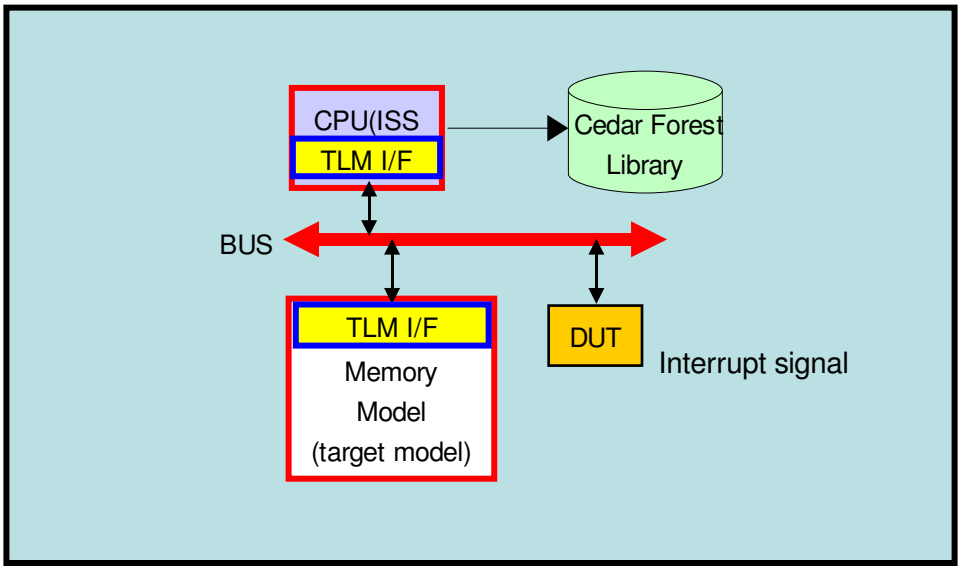


Figure 1.2: SH4A Environment

SH4A Environment includes three kinds of model:

- Initiator model is CPU model
- Bus model is simple bus connecting between Initiator and target models
- Target models are Memory model and DUT (Design Under Test)

All models are also connected together according to Loosely-timed TLM interface. Particularly, the environment uses Cedar Forest Library to execute SH4A functionalities.

1.1.Unsupported functionality

– Verification Environment supports for only Loosely-Timed TLM protocol, NOT Approximately-Timed TLM

– For SH4A Environment, Interrupt model is not supported

– For Stimulus Environment, Stimulus model supports for processing interrupts of only one DUT

1. Verification-Environment-Generation Environment

This part describes the environment (mandatory file and folders) of generating verification environment so that users are able to check whether or not the file and folders are sufficient before using the environment generation script.

File/Folder	Explanation
1.*TLMForest* folder	This folder includes all mandatory source and header files together with the script files, make files, skeleton files, and text files which are necessary to generate the expected verification environment
1.1 Model files	
address_map.h	Caddress_map class is used for processing address between bus, initiator and target model
simple_memory.h	Csimple_memory class is memory model
simple_bus_lt.h	Csimple_bus_lt class is implemented as Loosely-time TLM Bus model
sitmulus.h	Cstimulus class is stimulus model
tlm_if.h tlm_tgt_if.h tlm_ini_if.h	These files implement TLM common classes
reg_super.h re_register.h re_register.cpp	These files implement Register common classes
cpu.h cpu.cpp cpu_utility.cpp cpu_command.cpp forest_utility.h cedar_forest_api.h memory_body.h memory_body.cpp cedar_forest_api.h	All models are used to execute SH4A functions
1.2 Skeleton files	
tlmforest.skl	Skeleton file of tlmforest.h
main_with_sm.skl	Skeleton file of forest_main.cpp in Stimulus environment
main_with_cpu.skl	Skeleton file of forest_main.cpp in SH4A environment
1.3 Text files	
tlmforest_SCI.txt	System Configuration file includes environment settings
stimulusSCI.txt	The text file includes settings for Stimulus model
1.4 Makefile files	
Makefile	The files include primary compilation settings

File/Folder	Explanation
Makefile.defs	
1.5 Script files	
gen_env.pl	Environment generation script
env_set.csh	The script setting environment variables according to whether complication machine is in RTC or RVC
env_rvc_set.csh	The script setting environment variables in RVC is called by env_set.csh
env_rtc_set.csh	The script setting environment variables in RTC is called by env_set.csh
makelink.bat	The script makes links to cedar_forest_api.h and Cedar Forest library in CedarX3 folder
2. *CedarX3* folder	The folder includes all necessary files to build Cedar Forest library

Table 1.1: The description of verification-environment-generation environment

2. Two steps to generate verification environment

This part describes how to generate verification environment and Address_map_info.txt by using gen_env.pl.

- Step 1: Preparing Configuration files
- Step 2: Running gen_env.pl script

2.1.Step 1 – Preparing configuration files

2.1.1.System Configuration file for verification environment

SCI (system configuration) file is used to describe all settings of verification environment which must be defined before the other step. This file includes two parts:

- Part 1: Environment description defines common information for all the environment.
- Part 2: Bus description defines information for each bus. If more than 1 bus are used, this part will be written many times with different information from each other.

The structure of System Configuration file is described as the following:

```
#################################################################
## Part 1: Environment descriptions
#################################################################
%ENV_INFO
%CLOCK_RATIO
%CHANNEL_SYS
%INTERRUPT_INFO
%SECTION
#################################################################
## Part 2: Bus descriptions
#################################################################
%BUS_INFO # for indicate Bus ID in multi buses.
%TOP_MODULE_PARAM
%TOP_MODULE_LIST
%VC_SPECIFIC_SYS
%MEMORY_INFO
%ADDRESS_MAP
%BASE_ADDRESS
%OFFSET_ADDRESS
%END   # end of configuration file
```

Figure 2.1: Structures of SCI file.

All information is marked by keywords which are written in SCI file regardless of Mandatory or Optional . The meaning of keywords in this file is described in table 2.1.

Keyword	Meaning	Note
%ENV_INFO	Information of environment: environment with cpu or stimulus, LT or AT	Mandatory
%CLOCK_RATIO	Clock ratio: name, cycle, duty cycle, start time, positive first or negative first	Mandatory
%CHANNEL_SYS	Signals information: signal's name, bit width, signal or port in or port out, type	Mandatory
%INTERRUPT_INFO	Interrupt general information	Mandatory
%SECTION	Section address is used to check result	Optional
%MEMORY_INFO	Address range of memory model	Mandatory
%BASE_ADDRESS	Base address of each model that has registers	Optional
%OFFSET_ADDRESS	Offset address of registers of DUT	Optional
%BUS_INFO	Information of bus: bus identifier, bus width	Mandatory
%TOP_MODULE_PARAM	System top level parameters	Mandatory
%TOP_MODULE_LIST	List informations of all modules: file name, class name, instance name, VC identifier.	Mandatory
%VC_SPECIFIC_SYS	Specific system of each module: its names and its signal names.	Mandatory
%ADDRESS_MAP	Memory range of target model. It includes: start address, end address and module identifier.	Optional
%END	End of configuration file	Mandatory
#	Commented part	Optional

Mandatory:	There must be at least one information line (Not commend line) after these keywords
Optional:	There doesn't have to be at least one information line (Not commend line) after these keywords

Table 2.1: List of keywords in SCI file

2.1.1.1.Detailed description

– **%ENV_INFO**: There is only one next line:

<Environment_id> <TLM_coding_style>

- <Environment_id>: Integer number to choose verification environment with stimulus or bus

  0: SH4A verification environment

  1: Stimulus verification environment

- <TLM_coding_style>: String to choose TLM/LT or TLM/AT coding style.

  LT: TLM/LT (Loosely-Timed) Environment

  AT: TLM/AT (Approximately-Timed) Environment. This style is not supported in this project.

- **%CLOCK_RATIO**: There are many lines. Each line is described as followings

  <clock_name> <period> <duty_cycle> <start_time> <positive_first>

  - <clock_name> : String to define name of a clock
  - <period> : Real number to define period of that clock
  - <duty_cycle> : Real number to define duty cycle of that clock
  - <start_time> : Real number to define the starting time to activate of the defined clocks
  - <positive_first>: String to define whether the defined clocks are positive or negative at the first time they are activated. This information has 2 values only:

    true : Positive

    false : Negative

- **%CHANNEL_SYS**: There are many lines. Each line is described as followings

  <signal_name> <bit_width> <signal/port in/port out> <class_name_of_data_type>

  - <signal_name>: String to define name of a signal
  - <bit_width> : Integer number to define bit width of that signal
  - <signal/port in/port out> : String to define type of signal. The information has only 4 values:

    in : port in

    out: port out

    inout : port in/out

    signal : signal
  - <class_name_of_data_type>: String to define class name of data type.

- **%INTERRUPT_INFO**: The next line is an integer number that indicate the width of interrupt signal of DUT

- **%SECTION**: There are many lines. Each line is described as followings

  <section_name> <section_address>

  - <section_name>: String to define name of a section. The usual values are: PASS, FAIL, REST
  - <section_address>: Hex number to define address of that section.

- **%BUS_INFO**: There are many lines. Each line is described as followings

  <header_file> <class_name> <instance_name> <bus_id> <bus_width>

  - <header_file> : String to define name of header file of a bus

- <class_name> : String to define name of bus class
- <instance_name> : String to define instance name of that bus
- <bus_id> : Integer number to define the index of that bus
- <bus_width> : Integer number to define the bus width

– **%TOP_MODULE_PARAM**: The content of this part is the next two lines:

NUM_OF_INT <*initiator_number*>

NUM_OF_TGT <*target_number*>

- <initiator_number> : Integer number to define the number of initiators connected to this bus
- <target_number> : Integer number to define the number of targets connected to this bus

– **%TOP_MODULE_LIST**: There are many lines. Each line is described as followings

<header_file> <class_name> <instance_name> <VC_id> <socket_name> <initiator_or_target>

- <header_file> : String to define name of header file of a module
- <class_name>: String to define name of class module
- <instance_name> : String to define instance name of that module
- <VC_id> : Integer number to define the index of module. This number will be started by “0” number for initiators VC and targets VC. For example: there are 2 modules: initiator module A and target module B, the index of these 2 modules will be 0, not 0 and 1.
- <socket_name> : String to define the socket name of that module
- <initiator_or_target> : String to define whether module is initiator module or target module. This part has two values:

  INI: Initiator module

  TGT: Target module

– **%VC_SPECIFIC_LIST**:There are many lines. Each line is described as followings

<instance_name> <port_name> <signal_name>

- <instance_name>: String to define instance name of a module
- <port_name>: String to define name of port of that module
- <signal_name>: String to define name of signal that are connected to port of that module. For example: “reset”

– **%MEMORY_INFO**: The content of this part is only one line that defines address range of memory model

<start_address> <end_address>

- <start_address>: Hex number (started by “0x”) to define the start address of memory model
- <end_address>: Hex number (started by “0x”) to define the end address of memory model

– **%ADDRESS_MAP**: There are many lines. Each line is described as followings

<start_address> <end_address> <target_id>

- <start_address>: Hex number (started by “0x”) to define the start address of memory model
- <end_address>: Hex number (started by “0x”) to define the end address of memory model
- <target_id>: Integer number to define the index of target VC

– **%BASE_ADDRESS**: There are many lines. Each line is described as followings

<base_address> <offset_mask> <model_id>

- <base_address> : Hex number (started by “0x”) to define the base address of each model
- <offset_mask> : Mask of offset address of each model
- <model_id> : Integer number to define the index of each model

– **%OFFSET_ADDRESS**: There are many lines. Each line is described as followings

<register_name> <offset_address> <model_id>

- <register_name> : String to define name of a register of a module.
- <base_address> : Hex number (started by “0x”) to define the offset address of that register.
- <model_id> : Integer number to define the index of each model.

The **%BASE_ADDRESS** and **%OFFSET_ADDRESS** information are used to create address information that will be used for TM generator.

2.1.1.1.Example

Supposing that the following verification environment are built

- Two buses
- One memory model
- One CPU model
- One interrupt model
- One bridge model
- One DUT (TMU) model

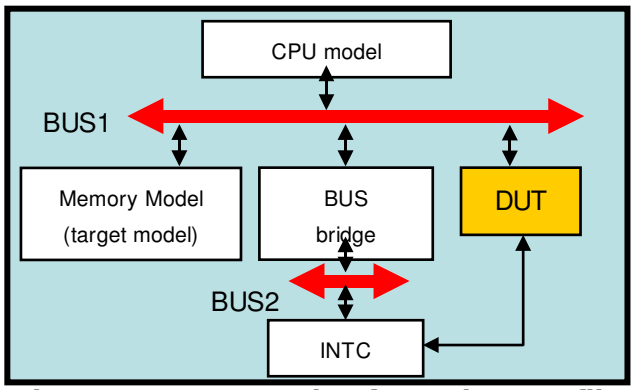


Figure 2.2: A example of creating SCI file for verification environment

Configuratic

```
%ENV_INFO
  0              LT    # CPU
%CLOCK_RATIO
#              clock name   period          duty_cycle    start time     positive_first
               sclk         5               0.5           0              true
               pclk         30              0.5           0              true
%CHANNEL_SYS
#              signal_name    bits_width    port_in/out_from_VC    class_name
               pclk           1             in                     bool
               reset          1             inout                  bool
               intr             1           signal                 sc_uint
%INTERRUPT_INFO
#  INTR_WIDTH
   1
%SECTION
#  section_name      section_address
   PASS              0x0C700000
   FAIL              0x0C700100
   REST              0x0C700200
%BUS_INFO ##########Bus 1
#  header_file                  class_name       instance_name  bus_id  bus_width
   bus.h                        bus              bus_1          0       32
%TOP_MODULE_PARAM
   NUM_OF_INI      2     # number of initiator VCs
   NUM_OF_TGT      3     # number of target VCs
%TOP_MODULE_LIST
#  head_file class_name instance_name  VC_indx Socket_name      ini/tgt
   cpu.h         cpu        cpu            0        cpu_ini            INI
   tmu.h         tmu        tmu            1        tmu_ini            INI
   tmu.h         tmu        tmu            0        tmu_tgt            TGT
   memory.h      memory     mem            1        mem_tgt            TGT
   bridge.h      bridge     bridge         2        bridge_tgt         TGT
%VC_SPECIFIC_SYS
#  instance_name   port_name              signal_name
   stim            reset      reset
   tmu             pclk       pclk
%MEMORY_INFO
#  start_address   end_address
   0x20000000      0x2FFFFFFF
%ADDRESS_MAP
#  initial addr      end addr        tgt_VC_indx
   0x00000000      0x0FFFFFFF      2  # bridge
   0x10000000      0x13FFFFFF      0  # tmu1
%BASE_ADDRESS
#  address           offset_mask          id
   0xFE410000        0x0000FFFF           1
%OFFSET_ADDRESS
#  reg_name     address     base_address_id
   TMU_TSTR                 0x0004      1
   TMU_TCOR_0               0x0008      1
%BUS_INFO #########Bus 2
   bus.h                    bus              bus_2          1       32
%TOP_MODULE_PARAM
   NUM_OF_INI      1     # number of initiator VCs
   NUM_OF_TGT      1     # number of target VCs
%TOP_MODULE_LIST
   bridge.h    bridge          bridge             0        bridge_ini        INI
   intc.h      intc            intc               0        intc_tgt          TGT
%VC_SPECIFIC_SYS
   intc            reset           reset
%ADDRESS_MAP
   0x14000000      0x16FFFFFF      0  # intc
%BASE_ADDRESS
   0x10000000      0x0000FFFF           0
%OFFSET_ADDRESS
   INTC_NMMI                0x0004      0
%END
%END
```

Figure 2.3: An example of SCI file for SH4A environment

Configuration file of stimulus model defines the input text pattern file and the maximum transaction

```
// Stimulus model
// Instance name      Parameter name        Parameter value
stimulus_module       TM_file_name          TM.txt
stimulus_module       max_transaction       1000
```

Figure 2.4: Example SCI file of stimulus model (stimulusSCI.txt)

Instance name must be defined as instance name of stimulus model which was defined in System Configuration file of Verification Environmen

TM_file_name [Test pattern name] : Define which input text pattern file will be processed

max_transaction [Number of maximum transaction]: Define number of maximum transactions

2.1.Step 2 – Running gen_env.pl script

2.1.1.Block chart

The running flow of script gen_env.pl comprises three stages described as the three following block charts

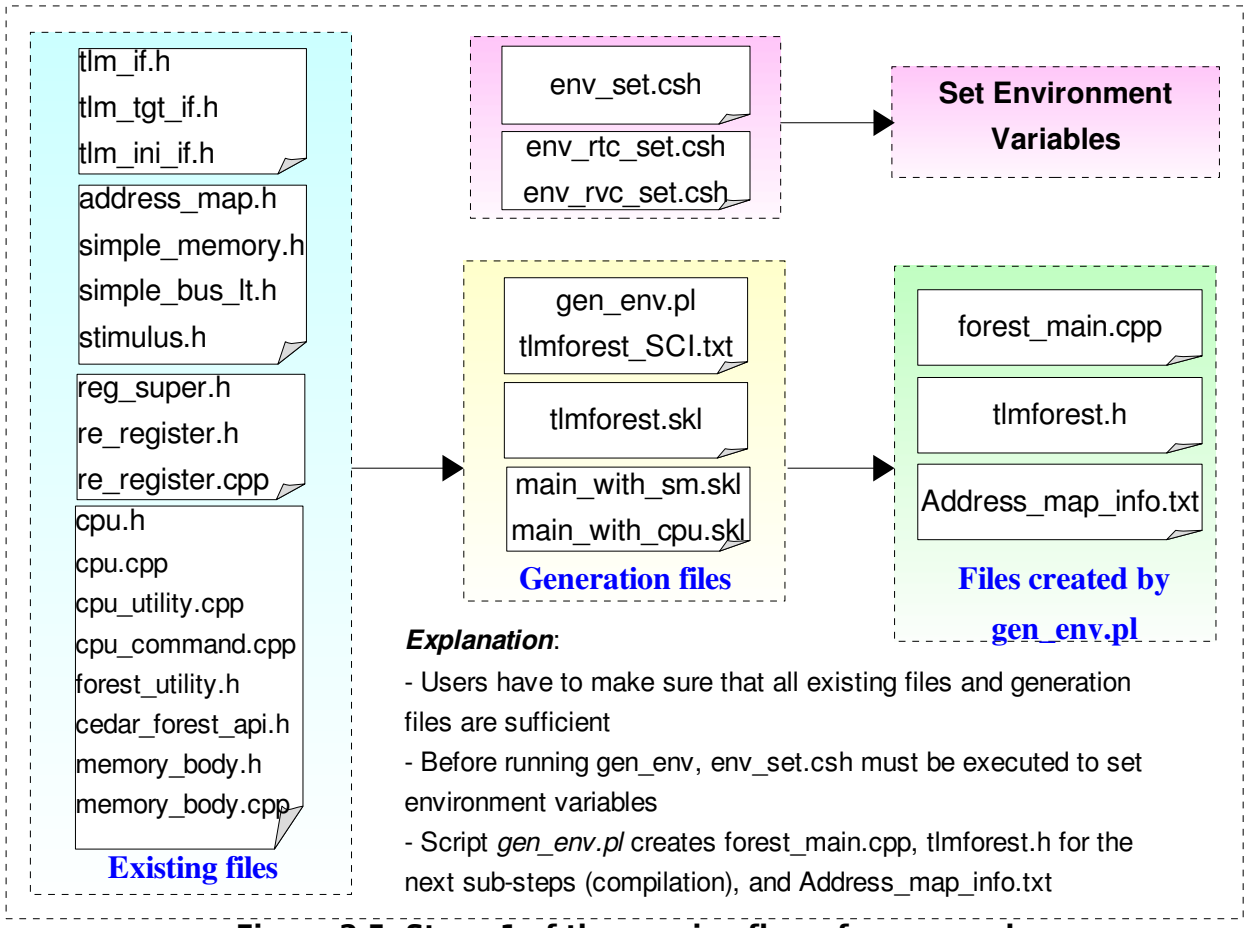
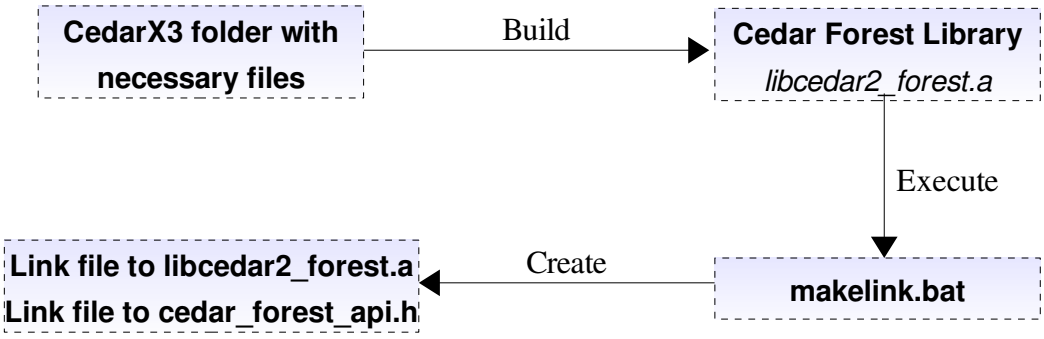


Figure 2.5: Stage 1 of the running flow of gen_env.pl



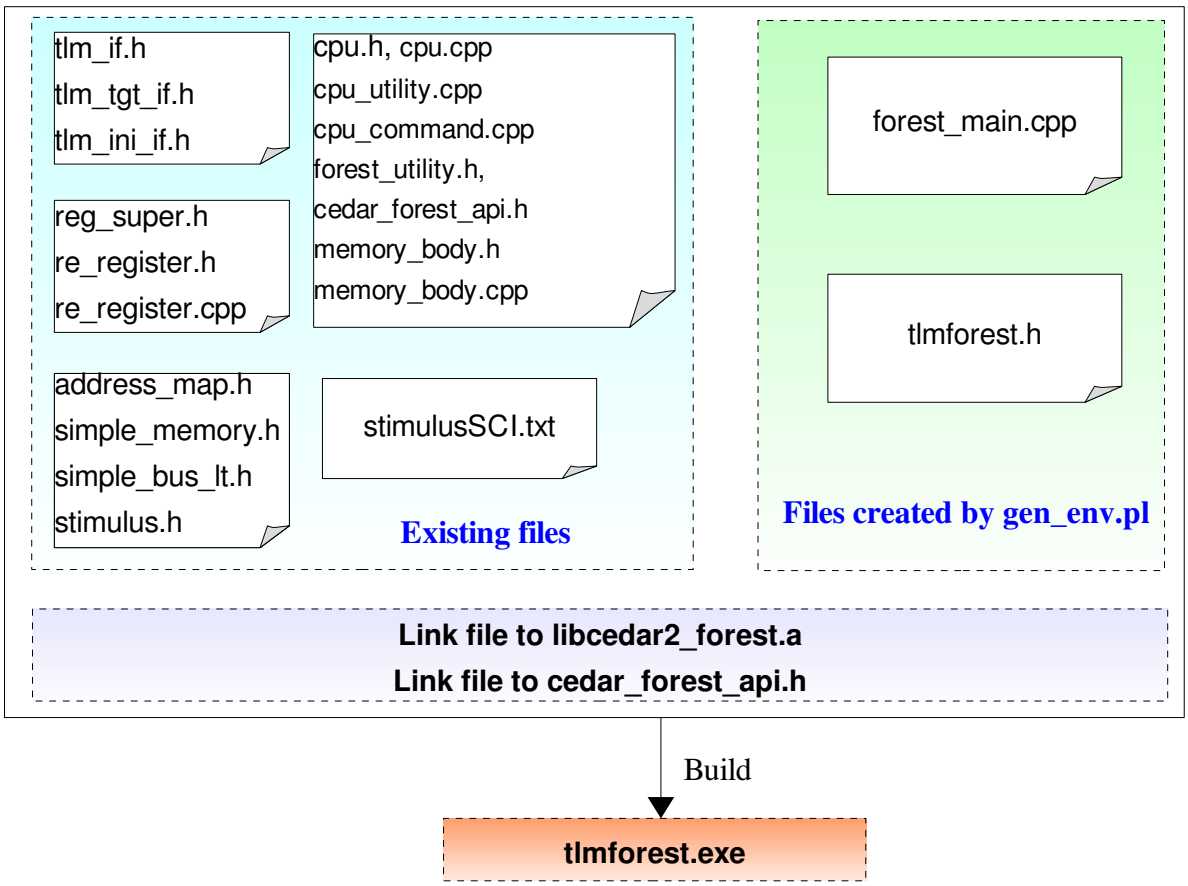
Explanation:

Script *gen_env.pl* continues:

Building Cedar Forest Library from CedarX3 folder

Executing *makelink.bat* to create link files of libcedar2_forest.a and cedar_forest_api.h from CedarX3 folder to verification environment.

Figure 2.6: Stage 2 of the running flow of gen_env.pl



Explanation:

Script *gen_env.pl* continues building tlmforest.exe from existing files, the files created by gen_env.pl except for Address_map_info.txt, and the two link files

Figure 2.7: Stage 3 of the running flow of gen_env.pl

2.1.2.Supported functions of gen_env.pl

Type	Content
Script name	gen_env.pl
Language	Perl version 5.8.5 or later
Purpose	- Build TLM/LT verification environment with CPU or Stimulus model - Create Address_map_info.txt file
Input file	All input files are described in Table 3.1
Output file	- Executable file *tlmforest.ext* - *Address_map_info.txt* file
Usage	*perl gen_env.pl* **[option]**
Option	**[option]** ***-i <directory path>*** : the path to the directory storing the skeleton and design of models. Default value: current directory. ***-oforest <directory path>***: the destination storing the output *tlmforest.exe*. Default value: current directory. ***-oaddress <directory path>***: the destination storing the output Address_map_info.txt. Default value: current directory. ***-sci <sci file name>*** : the name of input sci file. Default value: tlmforest_SCI.txt in the current directory. ***-build_cedar*** : build the CedarX3. Default value: Not build ***-help*** : print this helpful message.

Table 2.2: Supported functions of gen_env.pl

2.1.3.Error messages

No.	Severity	Error Message	Explanation
1	Error	[gen_env.pl] Error: Can't open Env SCI file: <*sci_file_name*>	System configuration file for verification environment doesn't exist
2	Error	[gen_env.pl] Error: Unknown argument : <*argument*> Please run again with option -help for the direction	Specified options of gen_env.pl is wrong
3	Error	[gen_env.pl] Error: Unknown command <*command*>	A command beginning with “%” symbol is wrong
4	Error	[gen_env.pl] Error: Can't open skeleton file: <*design_directory*>/tlmforest.skl	The skeleton file tlmforest.skl doesn't exist in <*design_directory*>
5	Error	[gen_env.pl] Error: Can't open skeleton file of forest_main	The skeleton file to create forest_main.cpp doesn't exist
6	Error	[gen_env.pl] Error: Can't create forest_main.cpp	The forest_main.cpp can not be created
7	Error	[gen_env.pl] Error: Can't create Address_map_info.txt	The Address_map_info.txt can not be created
8	Error	[gen_env.pl] Error: Repeated models are declared in %BASE_ADDRESS	At least one model is described two times in %BASE_ADDRESS
9	Error	[gen_env.pl] Error: Offset address defined isn't equivalent to base addess: <*offset_address*>	The model index is not equivalent between base address and offset adress
10	Error	[gen_env.pl] Error: Cannot create temp file to build environment	The temporary Shell script used to build Environment can not be created
11	Error	[gen_env.pl] Error: tlmforest.exe is not created! Compilation fails	Building verification environment is failed
12	Error	[gen_env.pl] Error: Can't open old file: <*design_directory*>/tlmforest.old	The tlmforest.old which is a copy of the current tlmforest.h doesn't exist
13	Error	[gen_env.pl] Error: Duplicated bus instance name: <*instance_name*>	At least one instance name of one model is described two times
14	Error	[gen_env.pl] Error: Duplicated ports or signals: <*port*>	At least one port of one model is described two times
15	Error	[gen_env.pl] Error: Unknown ports or signals: <*port*>	At least one port of one model is not described
16	Error	[gen_env.pl] Error: Lack of NUM_OF_INI information	NUM_OF_INI information is not described
17	Error	[gen_env.pl] Error: Redundant NUM_OF_INI information	NUM_OF_INI information is described more than 1 time
18	Error	[gen_env.pl] Error: Lack of NUM_OF_TGT information	NUM_OF_TGT information is not described
19	Error	[gen_env.pl] Error: Redundant NUM_OF_TGT information	NUM_OF_TGT information is described more than 1 time

20	Error	[gen_env.pl] Error: <*command*> is not declare	The <*command*> is not described in SCI file
21	Error	[gen_env.pl] Error: [%ENV_INFO] Lack of Environment information	The environment information is not described in %ENV_INFO command
22	Error	[gen_env.pl] Error: [%ENV_INFO] Wrong Environment mode	The environment information is described as not 0 and 1
23	Error	[gen_env.pl] Error: [%ENV_INFO] Wrong Environment Interface	The environment interface is described as not LT and AT
24	Error	[gen_env.pl] Error: [%ENV_INFO] No support AT Interface	AT environment interface is not supported
25	Error	[gen_env.pl] Error: [%CLOCK_RATIO] Lack of clock information	Clock information is not described in %CLOCK_RATIO command
26	Error	[gen_env.pl] Error: [%CLOCK_RATIO] Wrong clock name	Clock information is described wrongly
27	Error	[gen_env.pl] Error: [%CLOCK_RATIO] Wrong period value	Period value of a clock is described wrongly
28	Error	[gen_env.pl] Error: [%CLOCK_RATIO] Wrong duty cycle	Duty cycle of a clock is described wrongly
29	Error	[gen_env.pl] Error: [%CLOCK_RATIO] Duty cycle is bigger than 1	Duty cycle of a clock is described as bigger than 1
30	Error	[gen_env.pl] Error: [%CLOCK_RATIO] Wrong start time	Starting time of a clock is described wrongly
31	Error	[gen_env.pl] Error: [%CLOCK_RATIO] Wrong positive first value. It must be true or false	Wrong positive/negative value of a clock is described wrongly
32	Error	[gen_env.pl] Error: [%INTERRUPT_INFO] Lack of Interrupt information	Interrupt information is not described in %INTERRUPT_INFO command
33	Error	[gen_env.pl] Error: [%INTERRUPT_INFO]The Interrupt index must be number	Interrupt information is not an integer number
34	Error	[gen_env.pl] Error: [%SECTION]Lack of Section information	Section information is not described in %SECTION command
35	Error	[gen_env.pl] Error: [%SECTION]Section address is wrong format	Section information is not described wrongly
36	Error	[gen_env.pl] Error: [%BUS_INFO]Lack of Bus info parameter	Bus information is not described in %BUS_INFO command
37	Error	[gen_env.pl] Error: [%BUS_INFO]Wrong bus id	Bus ID is described wrongly
38	Error	[gen_env.pl] Error: [%BUS_INFO]Wrong bus width	Bus width is described wrongly
39	Error	[gen_env.pl] Error: [%TOP_MODULE_PARAM]Lack of System top level parameter	System top level parameter is not described in %TOP_MODULE_PARAM

40	Error	[gen_env.pl] Error: [%TOP_MODULE_PARAM] Parameter name of top module is not correct	Name of top module is described wrongly
41	Error	[gen_env.pl] Error: [%TOP_MODULE_PARAM]The index of top module parameter must be number	Name of top module is described wrongly
42	Error	[gen_env.pl] Error: [%TOP_MODULE_LIST]Wrong filename of header file	The file name of header file is described wrongly
43	Error	[gen_env.pl] Error: [%TOP_MODULE_LIST]Wrong class name of module	The class name is described wrongly
44	Error	[gen_env.pl] Error: [%TOP_MODULE_LIST]Wrong instance name of module	The instance name is described wrongly
45	Error	[gen_env.pl] Error: [%TOP_MODULE_LIST]Wrong module id	The module ID is described wrongly
46	Error	[gen_env.pl] Error: [%TOP_MODULE_LIST]Wrong socket name of module	The socket name is described wrongly
47	Error	[gen_env.pl] Error: [%TOP_MODULE_LIST]Wrong type of module. It must be INI or TGT	The type of module is described wrongly
48	Error	[gen_env.pl] Error: [%VC_SPECIFIC_SYS]Lack of information of systemc signal	The signal information is not described in %VC_SPECIFIC_SYS command
49	Error	[gen_env.pl] Error: [%MEMORY_INFO]Lack of memory information	The memory information is not described in %MEMORY_INFO command
50	Error	[gen_env.pl] Error: [%MEMORY_INFO]The address of memory information is wrong format	The address of memory information is wrong format
51	Error	[gen_env.pl] Error: [%MEMORY_INFO]The end address must be bigger than start address	The ending address is smaller than the stating address
52	Error	[gen_env.pl] Error: [%ADDRESS_MAP]Lack of memory map information	The memory map information is not described in %ADDRESS_MAP command
53	Error	[gen_env.pl] Error: [%ADDRESS_MAP]The address of memory map information is wrong format	The address of memory map information is wrong format
54	Error	[gen_env.pl] Error: [%ADDRESS_MAP]The target VC index must be number	The target VC index is not a number

55	Error	[gen_env.pl] Error: [%ADDRESS_MAP]The end address must be bigger than initial address	The ending address is smaller than the initial address
56	Error	[gen_env.pl] Error: [%BASE_ADDRESS]Lack of base address information	Base address information is not described in %BASE_ADDRESS command
57	Error	[gen_env.pl] Error: [%BASE_ADDRESS]The address of base address information is wrong format	The address of base address information is wrong format
58	Error	[gen_env.pl] Error: [%BASE_ADDRESS]The model id must be number	The model ID is not a number
59	Error	[gen_env.pl] Error: [%OFFSET_ADDRESS]Lack information	Information is not described in %OFFSET_ADDRESS command
60	Error	[gen_env.pl] Error: [%OFFSET_ADDRESS]Wrong register name	The register name is described wrongly
61	Error	[gen_env.pl] Error: [%OFFSET_ADDRESS]Wrong format of offset address	The format of offset address is described wrongly
62	Error	[gen_env.pl] Error: [%OFFSET_ADDRESS]Wrong model id	The model ID is described wrongly

Table 2.3: Error message of gen_env.pl

2.1.4.Usage

As mentioned in 4.2.1. Block chart, there are three stages in the running flow of gen_env.pl script. This part describes how to use gen_env.pl script

(1) **Preparing the necessary files**

- The first step, the most important step, is to prepare the necessary files because gen_env.pl script can not run successfully without one of these file. All necessary folders and files are described detailedly in 3.Verification-Environment-Generation Environment
- Users have to prepare carefully System Configuration file for the verification environment and configuration file for Stimulus model, in case of Stimulus environment, because these files decide how the environment verification is generated.
- Users are recommended prepare the structure of folders and files which is the same as described in this document. Or else, users have to execute gen_env.pl with its options carefully.

(1) **Setting environment variables**

- The next step is to set environment variables which are necessary for compilation in the next step by executing *env_set.csh* script
- env_set.csh script calls *env_rtc_set.csh* or *env_rvc_set.csh* according to whether users' machines are in RTC or RVC
- Users are recommended checking *env_rtc_set.csh* and *env_rvc_set.csh* and modify them if any existing settings in these files are not suitable.

(1) **Executing gen_env.pl script**

- The final step is to execute gen_env.pl with its options which users have to choose carefully:

Option	Notes to use
-i <dir_path>	- This option indicates the path to the directory storing the skeleton and design of models. Default value: current directory. - Users must remember that CedarX3 folder must placed at the same folder structure with <*dir_path*> *Example*: If <*dir_path*> is */common/work/Verification_Environment*, the CedarX3 --/common/work/ |__Verification_Environment | |__CedarX3 must be placed as the following
-oforest <dir_path>	The destination folder storing the output *tlmforest.exe*. Default value: current directory
-oaddress <dir_path>	The destination folder storing the output Address_map_info.txt. Default value: current directory
-sci <sci_file_name>	The name of System configuration file of the verification environment. Default value: tlmforest_SCI.txt in the current directory
-build_cedar	Build the CedarX3. Default value: Not build Users are recommended choosing the option in the first time run gen_env.pl

Table 2.4: Options of gen_env.pl

– Afterwards, users execute gen_env.pl with the chosen options

2.1.1.Example

(1) Preparing the necessary: The working directory is organized as the same as the following figure

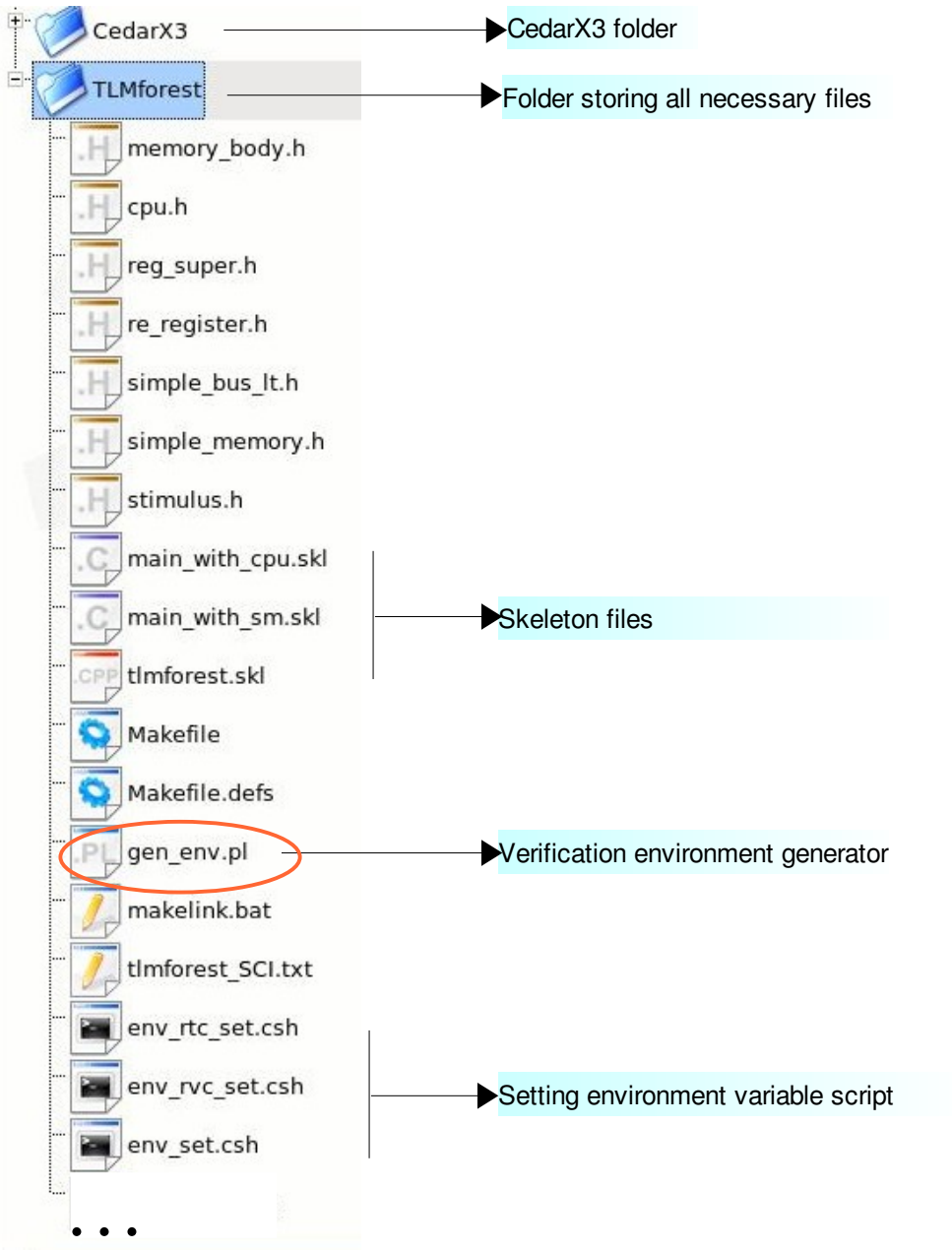


Figure 2.8: The working directory of the verification environment before executing gen_env.pl

The System Configuration file for the verification environment is tlmforest_SCI.txt which is written as the following

```
%ENV_INFO
  1   LT   # Stimulus
%CLOCK_RATIO   # clock paramter
  sclk   5    0.5   0     true
  pclk   30   0.5   0     true
%CHANNEL_SYS
  sclk     1       in              bool
  pclk     1       in              bool
  reset    1       inout           bool
  pa_little_sh  1  in              bool
  intr     9       signal          sc_uint
%INTERRUPT_INFO
  9
%SECTION   # PASS, FAIL section
P           0x00000000
P_MAIN      0x0C400000
P_PASS      0x0C700000
P_FAIL      0x0C700100
P_REST      0x0C700200
_INT_VBR    0x0C500000
%BUS_INFO # for indicate Bus ID in multi buses.
  simple_bus_lt.h  simple_bus_lt   simpleBusLT  0    32
%TOP_MODULE_PARAM   # system top level parameters
  NUM_OF_INI    1   # number of initiator VCs
  NUM_OF_TGT    2   # number of target VCs
%TOP_MODULE_LIST    # parameter for each bus module
  stimulus.h stimulus  stim       0    m_ini_socket   INI
  tmu.h   tmu     tmu       0    hpb_tgt_socket TGT
  simple_memory.h simple_memory   mem      1    m_tgt_socket   TGT
%VC_SPECIFIC_SYS   # systemc signal for each module
  tmu         intreq_tmu  intr
  stim        intr_sig    intr
  tmu         reset       reset
%MEMORY_INFO
  0x2C100000   0x2EEEEEEE
%ADDRESS_MAP
  0xFE410000   0xFE41FFFF  0  # tmu1
  0x2C000000   0x2FFFFFFF  1  # mem
%BASE_ADDRESS
  0xFE410000   0x0000FFFF  0
%OFFSET_ADDRESS   # offset address of registers in each model.
  TMU_TSTR     0x0004   0
  TMU_TCOR_0   0x0008   0
  TMU_TCNT_0   0x000C   0
  TMU_TCR_0    0x0010   0
  TMU_TCOR_1   0x0014   0
  TMU_TCNT_1   0x0018   0
  TMU_TCR_1    0x001C   0
  TMU_TCOR_2   0x0020   0
  TMU_TCNT_2   0x0024   0
  TMU_TCR_2    0x0028   0
%END
```

Stimulus Environment is chosen

Settings for Address_map_info.txt

There are only 1 bus

TMU is DUT

Figure 2.9: System Configuration file for Stimulus Environment

Because this is Stimulus environment, the configuration file of Stimulus model is also prepared

```
// sitmulus_SCI.txt
// Instance name        Parameter name              Parameter value
stim                TM_file_name          %TM_NAME
stim                max_transaction       1000
```

Figure 2.10: Configuration file for Stimulus model

(2) Setting environment variables: Because the machine is in RVC, *env_rvc_set.csh* is called by *env_set.csh*. The env_rvc_set.csh is available as the following figure

```
#!/bin/csh -f
#####################################
# CPAN environment setting
set path = (/shsv/sld/Common/Lib/02_Perl/CPAN_for_use_Spreed/ $path)
setenv PERL5LIB /shsv/sld/Common/Lib/02_Perl/CPAN_for_use_Spreed
# Toolchain directory define
setenv TOOL_PATH /common/appl/Renesas/shc/SHCV90200
# Source SHC
source ${TOOL_PATH}/shc.CSHRC_9.02.00
# C++ compiler
setenv CC g++
setenv TARGET_ARCH linux
setenv INCDIR  "-I./include -I/shsv/sld/Common/Lib/99_Others/include"
setenv LIBDIR "-L./libs/${TARGET_ARCH} -L/usr/local/lib -L/usr/lib
-L/shsv/sld/Common/Lib/99_Others"
# 32-bit machine
setenv LD_LIBRARY_PATH
"/shsv/sld/Common/Tools/gcc4.2.2_32bit_srv/lib":"$LD_LIBRARY_PATH"
setenv PATH "/shsv/sld/Common/Tools/gcc4.2.2_32bit_srv/bin":"$PATH"
# compile SystemC
setenv SC_TYPE systemc-2.2.0
# install directory
setenv SYSTEMC /shsv/sld/Common/Lib/01_SystemC/${SC_TYPE}
setenv TLM /shsv/sld/Common/Lib/04_TLM/TLM2.0-2008-06-09
# SystemC 2.2 is used
setenv SC_SIGNAL_WRITE_CHECK DISABLE
```

Figure 2.11: The env_rvc_set.csh script

(3) Executing gen_env.pl: The script gen_env.pl is executed as the following with option -build_cedar

> perl gen_env.pl -build_cedar

The working directory after executing gen_env.pl is the following

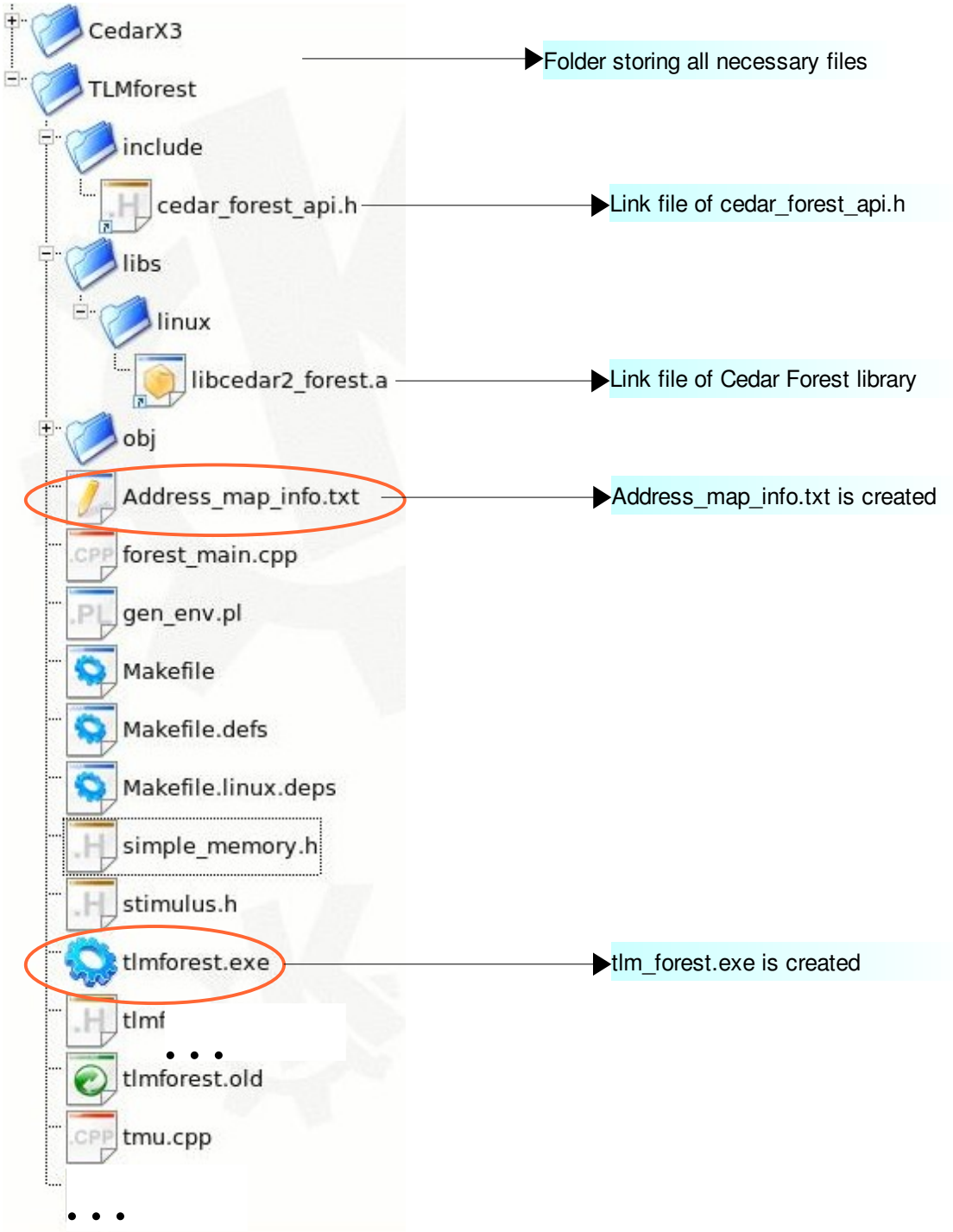


Figure 2.12: The working directory of the verification environment after executing gen_env.pl

Revision History

Rev.No	Contents	Approval	Checked	Created
1.0	Create new.		Chau Nguyen 10/06/09	Khang Le 09/25/09
1.01	- In Table 3.1, At “env_rtc_set.csh“, “The script setting environment variables in **RVC** is called by env_set.csh” is changed to “The script setting environment variables in **RTC** is called by env_set.csh” - In Table 4.2: Supported functions of gen_env.pl, Language is changed from “Perl version 5.8.8” to “Perl version 5.8.5 or later”	A.Imoto 10/14/09	Chau Nguyen 10/13/09	Khang Le 10/09/09