ATSSIM SystemVerilog Compiler and Simulator

USER MANUAL Ver. 1.03

Avinya Technology Systems P. Ltd.
1st Floor, HIG/ C-54, Shailendra Nagar.
Raipur (Chhattisgarh) - 492001. INDIA.
www.avinyatechnology.com

Note: Word "Organization" in above context refers to the company or party, who has legally and varifiably purchased or acquired HDL Compiler and Simulator tool from Avinya Technology Systems Private Limited.

TABLE OF CONTENTS

DISCLAIMER

Avinya Technology Systems has made every attempt to ensure accuracy and reliability of information present in this dcument. However, the information is provided "as is" without warrant of any kind. Avinya Technology Systems does not accept any responsibility or liability for the accuracy, content, completeness, legality or the reliability of information contained in this document.

No warranties, promises or/and representation of any kind, expressed or implied, are given as to the nature, standard, accuracy or otherwise of information provided in this document nor to the suitability or otherwise of the information to your particular application.

We shall not be lliable for any loss or damage of whatever nature (direct, indirect, consequential nature or other) whether arising in contract, tort or otherwise, which may arise as a result of your use of (or enabality to use) this tool or from your use of information in this user manual.

Any reference to the third party brand, logo, tool or information is copyright or IP of that respective third party. The content of such third party information is not under our control and we do not or will not take any responsibility for the information or content thereon.

---

2. Implemented Features of SystemVerilog

Design simulator tool is developed to compile and simulate verilog/ systemverilog designs and verification environment as per IEEE Standard for SystemVerilog – Unified Hardware Design, Specification, and Verification Language (IEEE std 1800-2012).

Following are indicative features of Systemverilog, which are supported by simulator.

Supported Features	Reference to LRM
General Features	Please, refer chapter 1 to 7 and 9 to 13 of IEEE std 1800-2012. All features in these chapters are implemented except few relatively less used features which are listed in scheduled to be added features below.
Class	Static class properties, Static methods, Inheritance, Virtual Methods, Polymorphism, Out-of-block declaration, Parameterized classes etc. Please, refer chapter 8 of IEEE std 1800-2012.
Clocking Blocks	refer chapter 14 of IEEE std 1800-2012
Interprocess synchronization and communication	mailboxes and semaphores
Constrained random value generation	random methods (randomize(), rand_mode(), constraint_mode() etc.). refer chapter 18 of IEEE std 1800-2012.
Functional Coverage	Covergroup, Coverpoint, Cross, bins and cross bins support. Option and type_option structure member support.

Utility and I/O system tasks	Most utility and Input/ Output system tasks and functions are supported. Please, refer to chapter 20 and 21 of IEEE std 1800-2012.
Compiler directives	All commonly used compiler directives are supported. Please, refer chapter 22 of IEEE std 1800-2012.
Module,Program, Interface, Package and generate block support	Please, refer chapter 23,24, 25, 26 and 27 of IEEE std 1800-2012.
Assertion	Immediate and Concurrent Assertions.
Direct Programming Interface (DPI)	DPI support for C/ C++ import / export function or task is available. C/C++ models of verification components can be seamlessly integrated in the verification environment. Please, refer chapter 35 of IEEE std 1800-2012.

4. How to use tool and Command description

How to run compiler and simulator?

Please follow few steps as under to compile and simulate your design:

If you have defined ATSSIM_LIC_PATH and ATSSIM_PKG_PATH environment variables as described in support main menu, create a work directory in any name and go to that directory.

If ATSSIM_LIC_PATH and ATSSIM_PKG_PATH environment variables are not defined then user must work in directory, where atssim license and atssim package is stored.

User can run atssim in following 2 modes.

Command Mode: User can compile or simulate any source file with a single command as under:

cmd_prompt>exeatssim -top [top_module_name] [source_code_files]

Interactive Mode: run atssim command on command prompt and you enter in interactive mode.

cmd_prompt>exeatssim

Note: Use exeatstsim instead of exeatssim for trial version of simulator.

Designs with DPI export/ import functions can be compiled and simulated in command mode only, right now.

How to get help on command line simulation options?

In command mode give a command as under and you will get all available options.

cmd_prompt>exeatssim -help

In interactive mode you can type help and you will get all compilation and simulation options on screen.

Note: Use exeatstsim instead of exeatssim for trial version of simulator.

What will be my simplified simulation flow?

When you will invoke atssim to simulate any source code, atssim will run following 3 phase in that order:

Compilation phase: Atssim will compile your source code first and show all syntactical errors. You need to make all necessary changes in your source code (as per systemverilog LRM), which are shown as errors in compilation phase. Atssim will not move to next phase unless all compilation errors are resolved.

Elaboration phase: Once there is no compilation error in source code, design will undergo elaboration phase. This phase creates storage for variables/ objects and entities and provides necessary interconnection between created objects and entities. This phase also initializes all

variable and entities with their default values as defined in source code.

Simulation phase: Successfully elaborated design moves in simulation phase and simulates design in a virtual time domain. There may be run time errors in design, which may not figure out at compile or elaboration time. Designers must watch simulation log carefully to remove run time errors in design.

Where is compiled package database library?

Whenever any package is compiled successfully then atssim creates a folder named ‘atspackage’ in same working directory. All compiled packages database are saved in this folder with a name convention <package_name>.db. Once a package database is compiled and saved in atspackege folder, this can be used by designer in any design using export <package_name>.* systemverilog construct.

If design has multiple packages with same name

Whenever a design is compiled or simulated, all defined packages in that design are saved in atspackage folder. Therefore, atssim will pickup most recently compiled package if you have multiple packages with same name.

How to update my package to most recent version of atssim package?

As long as your license is valid user can give ‘sudo atssim -update‘ command to update atssim package. Please ensure that you are connected to internet and logged in as root user.

COMMAND LINE HELP

exeatssim - will call command line systemverilog compiler and simulator.

Following are various commands and options, which can be used on atssim command prompt.

> exeatssim [options] [src_code_file_names]

src_code_file_names : acceptable source code file extensions are *.sv, *.v or *.svh.

: acceptable c/c++ source code files extensions are *.cc, *.cp, *.hh, *.hp, *.c, *.C, *.h, *.H, *.cxx, *.cpp, *.CPP, *.c++, *.hxx, *.hpp, *.HPP, *.h++. This is used when DPI c/c++ import or export functions or tasks are used in design.

Options:

-c or -compile : will only compile systemverilog source codes.

Note: Compiled database is saved inside a directory named atsdblib, which is created in current directory. If -odbf <filename> is not given then database file is saved in the name of 1st source code file.

The extension of any database file is (.db).

+incdir+<dir> : include search path for source code files.

+define+<name> : any `define <name>, which user wants to give in command line.

-odbf <filename> : any file name, which user wants to give to compiled database.

-idbf <filenames>: will load any compiled database in simulator for

simulation run.

Note : filename must have same name as saved in atsdblib folder. File name extension .db is optonal. Any available database file could be loaded on simulator.

-R : will load recently compiled design's snapshot (this is usually available in current directory in name of recentsnapshot.db)

-top <top_module>: Describe top module in design hirarchy before compilation.

-ni : If user wants to run simulation in non-interactive mode. Use this switch, this is used for compilation+elaboration+ simulation in one go. Used also in testcases regression.

-f <file_name> : write all options in a file named file_name and use this option to direct atssim for picking all commandline options from file file_name.

-l <log_file> : Give a log_file to save simulator's output to a file named log_file.

-etsim <time> : To give endtime of simulation in applicable time units. <time> must be a positive decimal number. default value is 3000.

-rseed <seed_val> : initialize RNG (Random Number Generator) of Top level. seed_val must be a positive integer. Default value is 2222.

-I<include_dir> : Use this option to search c/c++ source codes in include_dir folder. This option is useful when DPI import or export c/c++ functions are defined in design.

-nopobj : Compiler has inbuilt arrangement to save object codes of DPI export/ import c/c++ function in defined in a package. The object codes of such c/c++ source codes are saved in atspackagelib directory with a name <pkg_name>_objcode.o. When ever any design will import this package then will by default pick up this object code file for compilation, in case if user do not want compiler to pick up <pkg_name>_objcode.o file by default then user must add -nopobj switch on command line.

In this case user will have to supply c/c++ objects externally using -O<obj_file> option for DPI import/ export functions.

-O<object_file> : User can provide c/c++ object codes externally when using DPI c/c++ import/ export function or tasks.

-createlic : will update licence.dat file in current directory. New users will be able to generate a valid licence. And if a licence.dat file is damaged, it will be restored for existing users. Type 'help createlic' on command prompt to get more information.

-update : will download latest version of atssim on the machine if current version is not latest one.

-ver : will display current version of this software.

-checklic : will tell validity of license in days. It will show number of days passed after expiry of license, in case license has expired.

-clean : will delete 2 local files (authentication.dat and hdddata.dat).

use this opption when you are not able to run simulator and your license is valid. This will clean up temporary authenticatio files.

quit or exit : will exit from atssim.

User can use 'q' also to exit.

-help or -h : will display contents of this file 'help.txt'.

press 'q' to exit help.

5. Simulation of UVM base Testbenches

What is UVM

Universal Verification Methodology (UVM) is an automated and customizable verification framework written in Systemverilog. This HDL design verification framework empowers verification engineer to create a very efficient verification environment (testbench) to test a complex designs within a short time. UVM leverages re-usability of the verification components to reduce time and labor required to create verification environment.

How to know more about UVM

UVM framework is developed by one of the working groups of accellera (please, visit accellera.org for more information and downloading of UVM package for free).

Tested features of UVM package

atssim simulator has been successfully tested to run all example testcase supplied with uvm-1.2 package.

How to run uvm testcase?

We suggest following procedure to run uvm testcase:

current version supports uvm 1.2, therefore download uvm 1.2 package and extract it in a folder.

First of all compile uvm.sv file to get uvm_pkg databse in atspackage folder of your working folder. The compilation will take 3 – 8 minutes based on the speed of your machine/ server. We recommend to create

uvm_pkg.db file in atspackage folder so that for any subsequent uvm based testcase, you need not to recompile whole uvm_pkg, but you shall directly export it in your design. Thus making simulation process faster.

The above compilation activity is a one time and then you can use compiled uvm_pkg in all your subsequent testbench without bothering to compile it again.

Now to create any uvm based verification environment add export uvm_pkg.* in the beginning of your testbench to export whole uvm_pkg from atspackage folder within fraction of second. We also suggest to replace all code lines with `include uvm.sv by export uvm_pkg.* in your source code to avoid recompilation of uvm_pkg.

6. How to view vcd/ dump file

User must use $dumpfile to create a dump or VCD file with in initial construct.

User must use $dumpvars system task in initial construct to dump desired variable in dump file created using $dumpfile task.

Use free shareware waveform viewer gtkwave to view/ analyse signal wave form of VCD file. This could be installed on ubuntu/linux, as this is available as inbuilt package.

7. Trouble shooting, How to get technical Support and FAQs

Procedure to get Technical Support

If you are facing a technical problem while simulating your design follow below procedure to seek assistance.

We have placed a set of technical support FAQs in support menu for simple and known problems, Please walk through these FAQs and check if you can solve your issue. This section of FAQs is periodically updated for reported issues from users.

If your problem is not solved through FAQs then file an issue/ bug through support ticket system in support menu. Please, follow mentioned guidelines to file a technical support ticket on the system.

How to solve license related issues

If you are not able to run simulator on your machine due to a License issue. Please follow below procedure.

Please refer license FAQs in support menu and check if your problem is resolved.

If your issue is not resolved through License FAQs, then write to support@avinyatechnology.com or file a support ticket on system.

What is minimum hardware requirement to install this product?

All systems with x86 architecture or above with linux OS (recommended ubuntu 16.04, RedHat 7 or above) installed can run atssim. We recommend at least 4 GB RAM for individual license run to simulate module level designs. If users are running system level testcases using

UVM methodology, then large RAM (16 BG or more) is required based on complexity of design.

The best setup for industries is to use a high speed server having a RAM of 16 GB or higher in a linux environment.

What is recommended number of licenses on a single machine?

We do not recommend more than 5 licenses (means 5 users) on any server, because when system level designs are simulated with UVM based verification environment then it consumes big amount of RAM and thus slowing down performance of system considerably.

Does user need internet connection to run atssim?

atssim simulator is configured to access license server once in a 24 hours span for verifying license validity. Therefore, internet access is required at least once in an interval of 24 hours.

How frequently product update will be available?

Product update will be placed on site once in a week every Friday EOD. Occasionally we can place updated package on Wednesday too.

How can I install this product?

Please refer installation guide.

Is it a cycle based or event based simulator?

This product consists of a compiler and event based simulator.