VCS

VCS Flow

There are two ways to simulate design using VCS:

- Two-step Flow
- Three-step Flow

Two-step Flow

The two-step flow is supported only for Verilog HDL and SystemVerilog designs. Simulating a design using two-step flow involves two basic steps:

- Compilation
- Simulation

Compilation

Compiling is the first step to simulate your design. In this phase, VCS builds the instance hierarchy and generates a binary executable `simv`. This binary executable is later used for simulation.

```
vcs [compile options] verilog_files
```

Commonly used options:

Options	Specification
`-v filename`	Enables you to specify a Verilog library file.
`-full64`	Enables compilation and simulation in 64-bit mode.
`-v95`	Specifies not recognizing Verilog 2001 keywords.
`+v2k`	Support IEEE Std 1364-2001 Verilog.
`-sverilog`	Enables SystemVerilog constructs specified in the IEEE Standard of SystemVerilog, IEEE Std 1800-2009.
`-file` or `-f`	Specifies a file containing a list of files and compile-time options.
`-R`	Runs the executable file immediately after VCS links it together.
`-l filename`	Specifies a file where VCS records compilation messages.
`+define+macro=value+`	Defines a text macro in your source code to a value or character string. (`` `ifdef ``)
`timescale=1ns/1ns`	Specifies the time scale (time_unit/time_precision).
`debug_access+all`	Specifies the debug region.
`-o filename`	Specifies the binary executable filename.

Simulation

During compilation, VCS generates a binary executable, simv. You can use simv to run the simulation.

```
./simv

# records simulation messages
./simv -l sim.log

# or your specified executable filename
./***.simv

# use gui mode
./simv -gui
```

Three-step Flow

Simulating a design using three-step flow involves three basic steps:

- Analysis
- Elaboration
- Simulation

VCS uses these three steps to compile any design irrespective of the HDL, HVL, and other supported technologies used.

Analysis

Analysis is the first step to simulate your design. In this phase, you analyze your VHDL, Verilog, SystemVerilog, and OpenVera files using vhdlan or vlogan accordingly.

- Analyzing VHDL files:

```
vhdlan [vhdlan_options] file1.vhd file2.vhd
```

- Analyzing Verilog or SystemVerilog files:

```
vlogan [vlogan_options] file1.v file2.v
vlogan [vlogan_options] file1.sv file2.sv
```

Before you analyze your design using vhdlan or vlogan, ensure that the library mappings are defined in the synopsys_sim.setup file, and that the specified physical library for the logical library exists.

- vhdlan options:

Options	Specification
`-full64`	Analyzes the design for 64-bit simulation.
`-work library`	Maps a design library name to the logical library name WORK that receives the output of vhdlan.
`-vhdl87`	Enables to analyze non-portable VHDL code that contains object names that are now VHDL-93 reserved words by default.
`-vhdl02`	Enables to analyze the VHDL 2002 protected type.

Options	Specification
`-vhdl08`	Enables to analyze the VHDL 2008 constructs.
`-f filename`	Specifies a file that contains a list of source files.
`-l filename`	Specifies a log file where VCS records the analyzer messages.
`-timescale=1ns/1ns`	Specifies the time scale (time_unit/time_precision).

Elaboration

Elaborating is the second step to simulate your design. In this phase, using the intermediate files generated during analysis, VCS builds the instance hierarchy and generates a binary executable simv. This binary executable is later used for simulation.

```
vcs [elab_options] [libname.]design_unit
```

- `libname`: The library name where you analyzed your top module, entity, or the configuration.

Simulation

During compilation, VCS generates a binary executable, simv. You can use simv to run the simulation.

Example

```
vcs -full64 +v2k -debug_access+all -timescale=1ns/1ns tb_counter.v Counter.v -l com.log

vcs -full64 +v2k -debug_access+all -timescale=1ns/1ns tb_counter.v Counter.v -l com.log -R

vcs -full64 +v2k -debug_access+all -timescale=1ns/1ns tb_counter.v Counter.v -l com.log -o tb_cnt.simv

vcs -full64 +v2k -debug_access+all -timescale=1ns/1ns -f ./filelist.f -l com.log -o tb_cnt.simv


./simv

./simv -l sim.log

./simv -gui
```

```
mkdir -p work

vhdlan -full64 -vhdl08 tb_counter.vhd Counter.vhd

vhdlan -full64 -vhdl08 tb_counter.vhd Counter.vhd -work work

vcs -full64 -debug_all tb_counter

vcs -full64 -debug_all tb_counter -l com.log

./simv

./simv -l sim.log

./simv -gui
```