Compile-Time Options

****

-ad=<partition_filename>

Specifies the partition files used in mixed signal simulation.

-ams

Enables the use of Verilog-AMS code in VCS 2-step mode.

-ams_discipline <discipline_name>

Specifies the default discrete discipline in VerilogAMS in VCS 2-step mode.

-ams_iereport

Provides the auto-inserted connect modules (AICMs) information in VCS 2-step mode.

-as <assembler>

Specifies an alternative assembler. Only applicable in incremental compile mode, which is the default. Not supported on IBM RS/6000 AIX.

-ASFLAGS <options>

Passes options to assembler. Not supported on IBM RS/6000 AIX.

-assert <keyword_argument>

The keyword arguments and what they do are as follows:

disable_cover

Disables coverage for SVA cover statements.

dumpoff

Disables the dumping of SVA information in the VPD file.

dve

Enables SystemVerilog assertions tracing in the VPD file that you load into DVE. This tracing enables you to see assertion attempts.

enable_diag

Enables further control of SystemVerilog assertions result reporting with runtime options.

filter_past

Ignores SystemVerilog assertion subsequences containing past operators that have not yet eclipsed the history threshold.

vpiSeqBeginTime

Enables you to see the simulation time that a SystemVerilog assertion sequence starts when using Verdi.

vpiSeqFail

Enables you to see the simulation time that a SystemVerilog

assertion sequence doesn't match when using Verdi.

-C

Stops after generating the intermediate C or assembly code.

-cc <compiler>

Specifies an alternative C compiler.

-CC <options>

Works the same as -CFLAGS.

-CFLAGS <options>

Pass options to C compiler. Multiple -CFLAGS are allowed. Allows passing of C compiler optimization levels.

-cm line|cond|fsm|tgl|branch|assert

Specifies compiling for the specified type or types of coverage.
The arguments specifies the types of coverage:
line Compile for line or statement coverage.
cond Compile for condition coverage.
fsm Compile for FSM coverage.
tgl Compile for toggle coverage.
branch Compile for branch coverage
assert Compile for SystemVerilog assertion coverage.

If you want VCS to compile for more than one type of coverage, use the plus (+) character as a delimiter between arguments, for example:
-cm line+cond+fsm+tgl

-cm_assert_hier <filename>

Limits SystemVerilog assertions coverage to the module instances listed in the specified file.

-cm_cond <arguments>

Modifies condition coverage as specified by the argument or arguments:
basic Only logical conditions and no multiple conditions.
std The default: only logical, multiple, sensitized conditions.
full Logical and non-logical, multiple conditions, no sensitized conditions.
allops Logical and non-logical conditions.
event Signals in event controls in the sensitivity list position are conditions.
anywidth Enables conditions that need more than 32 bits.
for Enables conditions if for loops.
tf Enables conditions in user-defined tasks and functions.
sop Condition SOP coverage instead of sensitized conditions. also tells VCS that when it reads conditional expressions

that contain the ^ bitwise XOR and ~^ bitwise XNOR operators, it reduces the expression to negation and logical AND or OR.

You can specify more than one argument. If you do use the + plus delimiter between arguments, for example:

-cm_cond basic+allops

-cm_constfile <filename>

Specifies a file listing signals and 0 or 1 values. VCS compiles for line and condition coverage as if these signals were permanently at the specified values and you included the -cm_noconst option.

-cm_count

Enables accounting of the following:

In toggle coverage, not just whether a signal toggled from 0 to 1 and 1 to 0, but also the number of times it so toggled.

In FSM coverage, not just whether an FSM reached a state, had such a transition, but also the number of times it did.

In condition coverage, not just whether a condition was met or not, but also the number of times the condition was met.

In Line Coverage, not just whether a line was executed, but also the number of times.

-cm_glitch <period>

Specifies a glitch period during which VCS does not monitor for coverage caused by value changes. The period is an interval of simulation time specified with a non-negative integer.

-cm_dir <directory_path_name>

Specifies an alternative name and location for the coverage database directory.

-cm_fsmcfg <filename>

Specifies an FSM coverage configuration file.

-cm_fsmopt <keyword_argument>

The keyword arguments are as follows:

allowTmp

Allows FSM extraction when there is indirect assignment to the variable that holds the current state.

configonly

Disables extraction of FSMs automatically and through FSM pragmas. FSMs will be prepared only on the basis of the entries in FSM config file specified through the -cm_fsmcfg option.

report2StateFsms

By default VCS does not extract two state FSMs. This keyword tells VCS to extract them.

reportvalues

Specifies reporting the value transitions of the reg that holds the current state of a One Hot or Hot Bit FSM where there are parameters for the bit numbers of the signals that hold the current and next state.

reportWait

Enables VCS to monitor transitions when the signal holding the current state is assigned the same state value.

reportXassign

Enables the extraction of FSMs in which a state contains the X (unknown) value.

sequence

Enables compilation and monitoring of sequence coverage. Along with states and transitions, sequences covered during simulation will also be recorded.

stopConcatCase

Ignores FSMs in which concatenation is used in the case expression.

stopSelectInPackedMDA

Ignores FSMs in which select is applied on multi dimensional packed node.

upto64

Restricts extraction of FSMs in which the width of the FSM variable does not exceed 64 bits.

-cm_fsmresetfilter <filename>

Filters out the transitions in FSM coverage which occurs in assignments controlled by if statements where the conditional expression (following the if keyword) is a signal specified in the file. This filtering out can be on the specified signal in any module or the module specified in the file. The FSM and whether the signal is true or false can also be specified.

-cm_hier <filename>

When compiling for line, condition, FSM or toggle coverage, specifies a configuration file that specifies module definitions, source files, or module instances and their subhierarchies, that you want VCS to exclude from coverage or be the only parts of the design compiled for coverage.

-cm_ignorepragmas

Tells VCS to ignore pragmas for coverage metrics.

-cm_report <arguments>

The arguments are as follows:

unencrypted_hierarchies

compile time options to enable monitoring coverage for encrypted designs

noinitial

compile-time option to disable the the monitoring of the contents of initial blocks for line, condition, branch, and path metrics

-cm_libs yv|celldefine

Specifies compiling for coverage source files in Verilog libraries when you include the yv argument. Specifies compiling for coverage module definitions that are under the `celldefine compiler directive when you include the celldefine argument. You can specify both arguments using the plus (+) delimiter.

-cm_line <arguments>

Modifies line coverage as specified by the argument or arguments:

contassign Enables line coverage for continuous assignments

svtb Enables line coverage for programs and class function/tasks

You can specify more than one argument. If you do use the + plus delimiter between arguments, for example:

-cm_line contassign+svtb

-cm_noconst

Tells VCS not to monitor for conditions that can never be met or lines that can never execute because a signal is permanently at a 1 or 0 value.

-cm_seqnoconst

Enables a more sophisticated constant analysis compared to -cm_noconst. This includes analysis of non-blocking assignments and continuous assignments with delays, as well as handling multiple assignments to the same bits of a signal. As with -cm_noconst, coverable objects that VCS detects can never be hit are marked "unreachable" in coverage reports and removed from the computation of the coverage score.

-cm_tglfile <filename>

Specifies displaying at runtime a total toggle count for one or more subhierarchies specified by the top-level module instance entered in the file.

-cm_tgl mda

Enables toggle coverage for Verilog 2001 and SystemVerilog unpacked multidimensional arrays.

-cpp

Specifies a C++ compiler.

-debug_access

Enables dumping to FSDB/VPD, and limited read/callback capability. Use'-debug_access+class' for testbench debug, and '-debug_access+all' for all debug capabilities. Refer the VCS user guide for more granular options for debug control under the switch '-debug_access' and refer to '-debug_region' for region control."

-debug_pp

Enables dumping to FSDB/VPD, and use of UCLI, VERDI and DVE.

-debug

Same as -debug_pp, but also including 'force' capability

-debug_all

Enables all debug and dumping capability.

-dve_opt <dve_option>

You can use the argument called -dve_opt to pass DVE arguments from simv to DVE. Each DVE argument must be preceded by -dve_opt argument. In cases where the argument requires an additional option, the = sign needs to be used.(E.g. -dve_opt -session=file.tcl)

-e <new_name_for_main>

Specifies the name of your main() routine in your PLI application.

-f <filename>

Specifies a file that contains a list of pathnames to source files and compile-time options.

-F <filename>

Same as the -f option but allows you to specify a path to the file and the source files listed in the file do not have to be absolute pathnames.

-file filename

This option is for problems you might encounter with entries in files specified with the -f or -F options. This file can contain more compile-time options and different kinds of files. It can contain options for controlling compilation and PLI options and object files. You can also use escape characters and meta-characters in this file, like $, `, and ! and they will expand, for example:

-CFLAGS '-I$VCS_HOME/include'

/my/pli/code/$PROJECT/treewalker.o

-P /my/pli/code/$PROJECT/treewalker.tab

You can comment out entries in this file with the Verilog //

and /* */ comment characters.

-full64

Compiles the design in 64 bit mode and creates a 64 bit executable

for simulating in 64 bit mode.

-gen_asm

Specifies generating intermediate assembly code. Not supported on

IBM RS/6000 AIX.

-gen_c

Specifies generating intermediate C code. This is the default in

IBM RS/6000 AIX.

-gen_obj

Generate object code; default on Linux, Solaris, and HP platforms.

Not supported on IBM RS/6000 AIX.

-h or -help

Lists descriptions of the most commonly used compile-time and runtime

options.

-ID

Displays the hostid or dongle ID for your machine.

-ignore <keyword_argument>

The keyword arguments are as follows:

unique_checks

Suppresses warning messages about SystemVerilog unique if and unique case

statements.

priority_checks

Suppresses warning messages about SystemVerilog priority if and priority

case statements.

all

Suppresses warning messages about SystemVerilog unique if, unique case,

priority if and priority case statements.

-j<number_of_processes>

Specifies the number of processes to use for parallel compilation.

There is no space between the j character and the number.

-kdb[=option]

Enables generating Verdi KDB database. The sub-options are as follows:

only

Generate Verdi KDB database only.

-l <filename>

(lower case L) Specifies a log file where VCS records compilation messages and runtime messages if you include the -R, -RI, or -RIG options.

-ld <linker>

Specifies an alternative linker.

-LDFLAGS <options>

Pass options to the linker.

Only applicable in incremental compile mode.

-lmc-swift

Enables the LMC SWIFT interface.

-lmc-swift-template <swift_model_name>

Generates a Verilog template for a SWIFT Model.

-l<name>

Links the <name> library to the resulting executable.

-load <shared_VPI_library>:<registration_routine>

Specifies the registration routine in a shared library for a VPI application.

-Marchive=<number_of_module_definitions>

Tells the linker to create temporary object files that contain the specified number of module definitions. Use this option if there is a command line buffer overflow caused by too many object files on the linker command line.

-Mdelete

Use this option for the rare occurrence when the chmod -x simv command in the make file can't change the permissions on an old simv executable. This option replaces this command with the rm -f simv command in the make file.

-mhdl

Enables VCS MX - Mixed HDL simulation.

-Mlib=<directory>

Specifies the directory where VCS looks for descriptor information to see if a module needs to be recompiled. Also specifies a central place for object files. You use this option for shared incremental

compilation.

-Mmakeprogram=<program>

Program used to make object (default is make).

-Mupdate[=0]

By default VCS overwrites the Makefile between compilations. If you wish to preserve the Makefile between compilations, enter this option with the 0 argument.

Entering this argument without the 0 argument, specifies the the default condition, incremental compilation and updating the Makefile.

-negdelay

Enables the use of negative values in IOPATH and INTERCONNECT entries in SDF files.

-noIncrComp

Disables incremental compilation.

-notice

Enables verbose diagnostic messages.

-ntb

Enables the use of the OpenVera Testbench language constructs described in the OpenVera Language Reference Manual: Native TestBench.

-ntb_cmp

Compiles and generates the testbench shell (file.vshell) and shared object files. Use this option when compiling the .vr file separately from the design files.

-ntb_define <macro>

Specifies any OpenVera macro name on the command line. You can specify multiple macro names using the + delimiter.

-ntb_filext <.ext>

Specifies an OpenVera file extension. You can specify multiple filename extensions using the + delimiter.

-ntb_incdir <directory_path>

Specifies the include directory path for OpenVera files. You can specify multiple include directories using the + delimiter.

-ntb_noshell

Tells VCS not to generate the shell file.Use this option when you recompile a testbench.

-ntb_opts <keyword_argument>

The keyword arguments are as follows:

check

Reports error, during compilation or simulation, when there is an out-of-bound or illegal array access.

dep_check

Enables dependency analysis and incremental compilation. Detects files with circular dependencies and issues an error message when VCS cannot determine which file to compile first.

no_file_by_file_pp

By default, VCS does file by file preprocessing on each input file, feeding the concatenated result to the parser. This argument disables this behavior.

print_deps[=<filename>]

Enter this argument with the dep_check argument. This argument tells VCS to display the dependencies for the source files on the screen or in the file that you specify.

tb_timescale=<value>

Specifies an overriding timescale for the testbench. The timescale is in the Verilog format (for example, 10ns/10ns).

use_sigprop

Enables the signal property access functions. (for example, vera_get_ifc_name()).

vera_portname

Specifies the following:

The Vera shell module name is named vera_shell.

The interface ports are named ifc_signal.

Bind signals are named, for example, as: \if_signal[3:0].

You can enter more than one keyword argument, using the + delimiter, for example:

-ntb_opts use_sigprop+vera_portname

-ntb_shell_only

Generates only a .vshell file. Use this option when compiling a testbench separately from the design file.

-ntb_sfname <filename>

Specifies the filename of the testbench shell.

-ntb_sname <module_name>

Specifies the name and directory where VCS writes the testbench shell

module.

-ntb_spath

Specifies the directory where VCS writes the testbench shell and shared object files. The default is the compilation directory.

-ntb_vipext <.ext>

Specifies an OpenVera encrypted-mode file extension to mark files for processing in OpenVera encrypted IP mode. Unlike the -ntb_filext option, the default encrypted-mode extensions .vrp, .vrhp are not overridden, and will always be in effect. You can pass multiple file extensions at the same time using the + delimiter.

-ntb_vl

Specifies the compilation of all Verilog files, including the design, the testbench shell file and the top-level Verilog module.

-o <name>

Specifies the name of the executable file that is the product of compilation. The default name is simv.

-ovac

Starts the OVA compiler to check the syntax of OVA files on the vcs command line.

-ova_debug or -ova_debug_vpd

Enables OVA attempt dumping into VPD.

-ova_file <filename>

Specifies an OpenVera Assertions file. Not required if the filename has an .ova extension.

-ova_filter_past

For assertions that are defined with the past operator, ignore these assertions where the past history buffer is empty. For instance, at the very beginning of the simulation the past history buffer is empty. So, a check/forbid at the first sampling point and subsequent sampling points should be igorned until the past buffer has been filled with respect to the sampling point.

-ova_filter_last

Ignores assertion subsequences containing past operators that have not yet eclipsed the history threshold.

-ova_enable_diag

Enables runtime options for controlling functional coverage reports.

-ova_inline

Enables compiling OVA code that is written in Verilog source files.

-ova_lint

Enables general rules for the OVA linter

-ova_lint_magellan

Enables Magellan rules for the OVA linter.

-override-cflags

Tells VCS not to pass its default options to the C compiler.

-override_timescale=<time_unit>/<time_precision>

Overridesthe time unit and precision unit of all the `timescale compiler directives in the source code and, like the -timescale option, privides a timescale for all module definitions that preced the first `timescale compiler directive.

-P <pli.tab>

Specifies a PLI table file.

-parameters <filename>

Changes parameters specified in the file to values specified in the file. The syntax for a line in the file is as follows:

assign <value> <path_to_parameter>

The path to the parameter is similar to a hierarchical name except that you use slash characters (/) instead of periods as delimiters.

-platform

Returns the name of the platform directory in your VCS installation directory.

-pvalue+<parameter_hierarchical_name>=<value>

Changes the specified parameter to the specified value.

-q

Suppresses VCS compiler messages.

-R

Run the executable file immediately after VCS links together the executable file. You can add any runtime option to the vcs command line.

-sim_res=<time_precision>

Defines simulation resolution. It also defines timescales for modules which don't have timescales after analysis.

-sv_pragma

Tells VCS to compile the SystemVerilog assertions code that follows

the sv_pragma keyword in a single line or multi-line comment.

-sysc

Tells VCS to look in the ./csrc directory for the subdirectories containing the wrapper and interface files needed by the VCS/SystemC cosimulation interface to connect the Verilog and SystemC parts of a mixed Verilog and SystemC design.

-syslib <libs>

Specifies system libraries to be linked with the runtime executable.

-timescale=<time_unit>/<time_precision>

If only some source files contain the `timescale compiler directive and the ones that don't appear first on the vcs command line, use this option to specify the time scale for these source files.

-u

Changes all characters in identifiers to uppercase.

-ucli

Specifies UCLI mode at runtime.

-V

Enables the verbose mode.

-v <filename>

Specifies a Verilog library file to search for module definitions.

-vera

Specifies the standard VERA PLI table file and object library.

-vera_dbind

Specifies the VERA PLI table file and object library for dynamic binding.

-vhdlelab "<command_line_options>"

Specifies elaboration and compilation options for scs, the VHDL compiler.

-Vt

Enables warning messages and displays the time used by each command.

-y <directory_pathname>

Specifies a Verilog library directory to search for module definitions.

+allmtm

Allows you to specify at runtime which values in min:typ:max delay value triplets in compiled SDF files using the +mindelays, +maxdelays,

or +typdelays runtime options.

+applylearn[+<filename>]

Compiles your design to enable only the ACC capabilities that you needed for the debugging operations you did during a previous simulation of the design.

The +vcs+learn+pli runtime option records where you used ACC capabilities in a file named pli_learn.tab. If you do not change the file's name or location, you can omit +<filename> from this option.

+autoprotect[<file_suffix>]

Creates a protected source file; all modules are encrypted.

+auto2protect[<file_suffix>]

Create a protected source file that does not encrypt the port connection list in the module header; all modules are encrypted.

+auto3protect[<file_suffix>]

Creates a protected source file that does not encrypt the port connection list in the module header or any parameter declarations that precede the first port declaration; all modules are encrypted.

+bidir+1

Tells VCS to finish compilation when it finds a bidirectional registered mixed-signal net.

+charge_decay

Enables charge decay in trireg nets. Charge decay will not work if you connect the trireg to a transistor (bidirectional pass) switch such as tran, rtran, tranif1, or rtranif0.

+csdf+precompile

Precompiles your SDF file into a format that is for VCS to parse when it is compiling your Verilog code.

+csdf+precomp+dir+<directory>

Specifies the directory path where you want VCS to write the precompiled SDF file.

+csdf+precomp+ext+<ext>

Specifies an alternative to the "_c" character string addition to the filename extension of the precompiled SDF file.

+define+<macro_name>=<value>

Defines a text macro. Test for this definition in your Verilog source code using the `ifdef compiler directive.

+delay_mode_distributed

Specifies ignoring the module path delays and use only the delay specifications on all gates, switches, and continuous assignments.

+delay_mode_path

For modules with specify blocks, specifies ignoring the delay specifications on all gates and switches and use only the module path delays and the delay specifications on continuous assignments.

+delay_mode_unit

Specifies ignoring the module path delays and change all the delay specifications on all gates, switches, and continuous assignments to the shortest time precision argument of all the `timescale compiler directives in the source code.

+delay_mode_zero

Change all the delay specifications on all gates, switches, and continuous assignments to zero and change all module path delays to zero.

+deleteprotected

Allows overwriting of existing files when doing source protection.

+error+<n>

Enables you to increase the maximum number of NTB errors at compile-time to <n>.

+incdir+<directory>

Specifies the directories that contain the files you specified with the `include compiler directive. You can specify more that one directory, separating each path name with the + character.

+libext+<extension>

Specifies that VCS only search the source files in a Verilog library directory with the specified extension. You can specify more than one extension, separating each extension with the + character.
For example, +libext++.v specifies searches library files with no extension and library files with the .v extension.
Enter this option when you enter the -y option.

+liborder

Specifies searching for module definitions in the libraries that follow, on the vcs command line, a library that contains an unresolved instance before searching the libraries that precede the library with the unresolved instance.

+librescan

Specifies always starting the search for unresolved module definitions with the first library specified on the vcs command line.

+libverbose

Tells VCS to display a message when it finds a module definition in a source file in a Verilog library directory that resolves a module instantiation statement that VCS read in your source files, a library file, or in another file in a library directory.

+lint=[no]ID|none|all,...

Enables or disables Lint messages about your Verilog code.

+maxdelays

Use maximum value when min:typ:max values are encountered in delay specifications SDF files.

+memcbk

Enables callbacks for memories and multidimensional arrays (MDAs). Use this option if your design has memories or MDAs and you are doing any of the following:

- o Writing a VCD or VPD file during simulation. For VCD files, at runtime, you must also enter the +vcs+dumparrays runtime option. For VPD files you must enter the $vcdplusmemon system task. VCD and VPD files are used for post-processing with DVE or debugging using SmartDebug.
- o Using the VCS/SystemC Interface
- o Interactive debugging with DVE
- o Writing an FSDB file for Verdi
- o Using any debugging interface application - VCSD/PLI (acc/pli) that needs to use value change callbacks on memories or MDAs. APIs like acc_add_callback, vcsd_add_callback, and vpi_register_cb need this option if these APIs are used on memories or MDAs.

+mindelays

Use minimum value when min:typ:max values are encountered in delay specifications and SDF files.

+multisource_int_delays

Enables multisource interconnect delays.

+nbaopt

Removes the intra-assignment delays from all the nonblocking assignment statements in your design.

+neg_tchk

Enables negative values in timing checks.

+nocelldefinepli+0|1|2

For specifying what VCS records in the VPD file about nets and registers defined under the `celldefine compiler directive.

0 enables recording the transition times and values of nets and registers in all modules defined under the `celldefine compiler directive or defined in a library that you specify with the -v or -y compile-time options.

1 disables recording the transition times and values of nets and registers in all modules defined under the `celldefine compiler directive.

2 disables recording the transition times and values of nets and registers in all modules defined under the `celldefine compiler directive or defined in a library that you specify with the -v or -y compile-time options whether the modules in these libraries are defined under the `celldefine compiler directive or not.

+noerrorIOPCWM

Changes the error condition, when a signal is wider or narrower than the inout port to which it is connected, to a warning condition, thus allowing VCS to create the simv executable after displaying the warning message.

+nolibcell

Specifies not defining modules in libraries as cells unless they are under the `celldefine compiler directive.

+nospecify

Suppresses module path delays and timing checks in specify blocks.

+notimingcheck

Suppresses timing checks in specify blocks.

+nowarnTFMPC

Suppress the "Too few module port connections" warning messages during Verilog Compilation.

+no_notifier

Disables the toggling of the notifier register that you specify in some timing check system tasks.

+no_tchk_msg

Disables the display of timing check warning messages but does not disable the toggling of notifier registers in timing checks. This is also a runtime option.

+optconfigfile+<filename>

Specifies the VCS configuration file.

+overlap

Enables accurate simulation of multiple non-overlapping violation windows for the same signals specified with negative delay values in timing checks.
See the section on "Using Multiple Non-Overlapping Windows" in the VCS/VCSi User Guide.

+pathpulse

Enables the search for the PATHPULSE$ specparam in specify blocks.

+pli_unprotected

Enables PLI and UCLI access to the modules in the protected source file being created (PLI and UCLI access is normally disabled for protected modules).

+plusarg_save

Enter this option in the file that you specify with the -f option so that VCS passes to the simv executable the options beginning with a plus + character that follow in the file.

+plusarg_ignore

Also enter this option in the file that you specify with the -f option so that VCS does not pass to the simv executable the options that follow in the file. Use this option with the +plusarg_save option to specify that other options should not be passed.

+print+bidir+warn

Tells VCS to display a list of bidirectional registered mixed-signal nets.

+protect[<file_suffix>]

Creates a protected source file; only encrypting `protect/`endprotect regions.

+pulse_e/<number>

Specifies flagging as error and drive X for any path pulse whose width is less than or equal to the percentage of the module path delay specified by the number argument.

+pulse_int_e/<number>

Same as the +pulse_e option but only applies to interconnect delays.

+pulse_int_r/<number>

Same as the +pulse_r option but only applies to interconnect delays.

+pulse_on_event

Specifies that when VCS encounters a pulse shorter than the module path delay, VCS waits until the module path delay elapses and then drives an X value on the module output port and displays an error message.

+pulse_on_detect

Specifies that when VCS encounters a pulse shorter than the module path delay, VCS immediately drives an X value on the module output port, and displays an error message. It does not wait until the module path delay elapses.

+pulse_r/<number>

Reject any pulse whose width is less than number percent of module path delay.

+putprotect+<target_dir>

Specifies the target directory for protected files.

-race

Specifies that VCS generate a report, during simulation, of all the race conditions in the design and write this report in the race.out file.

-race=all

Analyzes the source code during compilation to look for coding styles that cause race conditions.

-racecd

Specifies that VCS generate a report, during simulation, of the race conditions in the design between the `race and `endrace compiler directives and write this report in the race.out file.

+race_maxvecsize=<size>

Specifies the largest vector signal for which the dynamic race detection tool looks for race conditions.

+rad

Performs Radiant Technology optimizations on your design.

+sdfprotect[<file_suffix>]

Creates a protected SDF file.

+sdf_nocheck_celltype

Tells VCs not to check to make sure that the CELLTYPE entry in the SDF file does not match the module identifier for a module instance before back annotating delay values from the SDF file to the module instance.

+sdfverbose

Enables the display of more than ten warning and more than ten error messages about SDF back annotation.

+spl_read

Tells VCS to treat output ports as inout ports in order to facilitate more accurate multi-driver contention analysis across module boundaries. This option can have an adverse impact on runtime performance.

+systemverilogext+<ext>

Specifies a filename extension for source files containing SystemVerilog source code.

+tetramax

Enter this option when simulating TetraMAX's testbench in zero delay mode.

TetraMAX can run the simv executable file. This option tell VCS to prepare the simv executable for use by TetraMAX.

+timopt+<clock_period>

Enables Timing Check Optimizations, the +<clock_period> argument specifies the clock period of the fastest clock in the design.
Please refer to the VCS/VCSi User Guide for more information on this option.

Starts the Timopt timing optimizer. the +<clock_period> argument specifies the clock period of the fastest clock in the design.
Timopt applies timing optimizations to your design. Timopt also writes a timopt.cfg file in the current directory. This file contains clock signals and module definitions of sequential devices it's not sure of. You edit this file and recompile without the +<clock_period> argument to obtain more Timopt optimizations.

+transport_int_delays

Enables transport delays with full pulse control for single source nets.

+transport_path_delays

Turns on the transport behavior for I/O paths.

+typdelays

Use typical value when min:typ:max values are encountered in delay specifications and SDF files.

+v2k

Enables the use of new Verilog constructs in the 1364-2001 standard.

+vc[+abstract][+allhdrs][+list]

Enables the direct call of C/C++ functions in your Verilog code using the DirectC interface. The optional suffixes specify the following:

+abstract

Specifies that you are using abstract access trough vc_handles to the data structures for the Verilog arguments.

+allhdrs

Writes the vc_hdrs.h file that contains external function declarations that you can use in your Verilog code.

+list

Displays on the screen all the functions that you called in your Verilog source code.

+vcs+dumpvars

A substitute for entering $dumpvars, without arguments, in your Verilog code.

+vcs+flush+log

Increases the frequency of flushing both the compilation and simulation log file buffers.

+vcs+flush+all

Shortcut option for entering all three of the +vcs+flush+log, +vcs+flush+dump, and +vcs+flush+fopen options.

+vcs+initmem+0|1|x|z

Initializes all bits of all memories in the design.

+vcs+initreg+0|1|x|z

Initializes all bits of all regs in the design.

+vcs+lic+vcsi

Checks out three VCSi licenses to run VCS.

+vcsi+lic+vcs

Checks out a VCS license to run VCSi when all VCSi licenses are in use.

+vcs+lic+wait

Tells VCS to wait for a network license if none is available.

+vcsi+lic+wait

Tells VCSi to wait for a network license if none is available.

+vcs+fsdbon

A compile-time substitute for $fsdbDumpvars option. The +vcs+fsdbon switch enables dumping for the entire design. If you do not add a corresponding -debug* switch, then -debug_access is automatically added. Note that you must also set VERDI_HOME.

+vcs+vcdpluson

A compile-time substitute for $vcdpluson option. The +vcs+vcdpluson switch enables dumping for the entire design. If you do not add a corresponding -debug* switch, then -debug_access is automatically added.

+vcs+mipdexpand

Intended to use with +oldsdf. When back annotating SDF delay values from an ASCII text SDF file at run-time, if the SDF file contains PORT entries for the individual bits of a port, using this compile-time option enables VCS to backannotate these PORT entry delay values. Similarly, using this compile-time option enables a PLI application to pass delay values to individual bits of a port.

+verilog1995ext+<ext>

Specifies a filename extension for source files containing Verilog 1995 source code.

+verilog2001ext+<ext>

Specifies a filename extension for source files containing Verilog 2001 source code.

+vhdllib+<logical_libname>

This option specifies the VHDL logical library to use for VHDL design entity instances that you instantiate in your Verilog design.

+vpi

Enables the use of VPI PLI access routines.

+warn=[no]ID|none|all,...

Enables or disables warning messages.

Runtime Options

****

-a <filename>

Specifies appending all messages from simulation to the bottom of the text in the specified file as well as displaying these messages to the standard output.

-assert <keyword_argument>

The keyword arguments and what they do are as follows:

dumpoff

Disables the dumping of SVA information in the VPD file during simulation.

filter

Blocks reporting of trivial SystemVerilog assertion implication successes. These happen when an implication construct registers a

success only because the precondition (antecedent) portion is false (and so the consequence portion is not checked). With this option, reporting only shows successes in which the whole expression matched.

finish_maxfail=<N>

Terminates the simulation if the number of SystemVerilog assertion failures for any assertion reaches N. N must be supplied, otherwise no limit is set.

global_finish_maxfail=<N>

Stops the simulation when the total number of failures, from all SystemVerilog Assertions, reaches N.

maxcover=<N>

Disables the collection of coverage information for cover statements after the cover statements are covered N number of times. <N> must be a positive integer, it can't be 0.

maxfail=<N>

Limits the number of SystemVerilog assertion failures for each assertion to N. When the limit is reached, the assertion is disabled. N must be supplied, otherwise no limit is set.

maxsuccess=<N>

Limits the total number of reported SystemVerilog assertion successes to N. N must be supplied, otherwise no limit is set. The monitoring of assertions continues, even after the limit is reached.

nocovdb

Tells VCS not to write the <program_name>.db file for assertion coverage.

nopostproc

Disables the display of the SVA coverage summary at the end of simulation.

quiet|quiet1

quiet Disables messages, in standard output, about assertion failures.

quiet1 Disables messages, in standard output, about assertion failures, but displays the summary of them at the end of simulation. The never triggered assertions are also reported.

report[=<filename>]

Generates a SystemVerilog assertion report file in addition to displaying results on your screen. By default the file's name and location is ./simv.vdb/report/ova.report, but you can change this by entering the filename pathname argument.

success

Enables reporting of successful SystemVerilog assertion matches in addition to failures. The default is to report only failures.

verbose

Adds more information to the report specified by the report=<filename> keyword, including assertions that never triggered and attempts that did not finish, and a summary with the number of assertions present, attempted, and failed.

You can enter more than one keyword, using the plus + separator, for example:

-assert maxfail=10+maxsuccess=20+success+filter

-cm line|cond|fsm|tgl|branch|assert

Specifies monitoring for the specified type or types of coverage.
The arguments specifies the types of coverage:
line Monitor for line or statement coverage.
cond Monitor for condition coverage.
fsm Monitor for FSM coverage.
tgl Monitor for toggle coverage.
branch Monitor for branch coverage.
assert Monitor for SystemVerilog assertions coverage.

If you want VCS to monitor for more than one type of coverage, use the plus + character as a delimiter between arguments, for example:
-cm line+cond+fsm+tgl

-cm_dir <directory_path_name>

Specifies an alternative name and location for the coverage database directory.

-cm_glitch <period>

Specifies a glitch period during which VCS does not monitor for coverage caused by value changes. The period is an interval of simulation time specified with a non-negative integer.This runtime option only works for toggle coverage.

-cm_log <filename>

Specifies a log file for monitoring for coverage during simulation.

-cm_name <filename>

Specifies unique name of that test during simulation.

-cm_tglfile <filename>

Specifies displaying at runtime a total toggle count for one or

more subhierarchies specified by the top-level module instance entered in the file.

-E <program>

Starts the program that displays the compile-time options that were on the vcs command line when you created the simv (or simv.exe or some other name specified with the -o option) executable file.

-grw <filename>

Sets the name of the $gr_waves output file to the specified file. The default filename is grw.dump.

-gui[=<dve|verdi>]

Starts user specified graphical user interface. If no argument is given, VCS will start Verdi when a valid VC_HOME environment variable is detected. Otherwise DVE will be started by default.

-i <filename>

Specifies a file containing UCLI commands that VCS executes when simulation starts.

-k <filename> | off

Specifies an alternative name or location for the vcs.key file into which VCS writes the UCLI interactive commands that you enter during simulation. The off argument tells VCS not to write this file.

-l <filename>

Specifies writing all messages from simulation to the specified file as well as displaying these messages in the standard output. This option begins with the letter "l" (lowercase "L") for log file.

-ova_filter

Blocks reporting of trivial if-then successes. These happen when an if-then construct registers a success only because the if portion is false (and so the then portion is not checked). With this option, reporting only shows successes in which the whole expression matched. This option is enabled by the -ova_enable_diag compile-time option.

-ova_max_fail <N>

Limits the number of reported failures for each assertion to N. The monitoring of assertions continues, even after this limit is reached. This option is enabled by the -ova_ebable_diag compile-time option.

-ova_max_success <N>

Limits the number of successes for each assertion to N. The monitoring of assertions continues, even after the limit is reached. This option is enabled by the -ova_enable_diag compile-time option.

-ova_name <name | /<pathname>/<name>

Specifies an alternative name or location and name for the ./simv.vdb/scov/results.db and ./simv.vdb/reports/ova.report files. You use this option if you want data and reports from a series of simulation runs. It's a way of keeping VCS from overwriting these files from a prvious simulation.

If you just specify a name the alternatively named files will be in the default directories. If you specify a pathname, with an argument containing the slash character /, you specify a different location and name for these files, for example:

-ova_name /net/design1/ova/run2

This example tells VCS to write run2.db and run2.report in the /net/design1/ova directory.

-ova_report [<filename>]

Specifies writing an OpenVera Assertions report file. The default file name and location is simv.vdb/report/ova.report but you can specify a different name and location as an argument to this option.

-ova_simend_max_fail <N>

Terminates the simulation if the number of failures for any assertion is reached. This option is enabled by the -ova_enable_diag compile-time option.

-ova_success

Enables the reporting of successful matches. This option is enabled by the -ova_enable_diag compile-time option.

-ova_quiet [1]

Disables displaying functional coverage results on the screen. The optional 1 argument specifies displaying a summary of these results.

-ova_verbose

Adds more information to the end of the report including assertions that never triggered and attempts that did not finish, and a summary with the number of assertions present, attempted, and failed.

-q

Quiet mode. Suppress printing of VCS header and summary information, the proprietary message at the beginning of simulation, and the VCS Simulation Report at the end of simulation (time, CPU time, data structure size, and date)

-sverilog

Enables the use of the Verilog language extensions in the Accellera SystemVerilog specification.

-ucli

Enables the use of UCLI commands.

-V

Verbose mode. Print VCS version and extended summary information.
Prints VCS compile and run-time version numbers, and copyright
information, at start of simulation.

-vcd <filename>

Sets the output VCD file name to the specified file.
The default filename is verilog.dump.
A $dumpfile system task in the Verilog source code will override
this option.

-verdi

Starts the Verdi graphical user interface.

-verdi_opts

Pass runtime options to verdi gui.

+vcdfile+<filename>

Specifies the VCD file you want to use for post-processing.

-vhdlrun "<scsim command line options>"

VCS-MX option to pass scsim command line options for the VHDL part of
a mixed HDL design.

-xzcheck [nofalseneg]

Checks all the conditional expressions in the design and displays
a warning message every time VCS evaluates a conditional expression
to have an X or Z value.

nofalseneg

Suppresses the warning message when the value of a conditional
expression transitions to X or Z and then to 0 or 1 in the same
simulation time step.

+maxdelays

Species using the compiled SDF file for maximum delays generated
by the +allmtm compile-time option.
Also specifies using maximum delays for SWIFT VMC or SmartModels
or Synopsys hardware models if you also enter the
+override_model_delays runtime option.

+mindelays

Specifies using the compiled SDF file for minimum delays generated
by the +allmtm compile-time option.
Also specifies using minimum delays for SWIFT VMC or SmartModels
or Synopsys hardware models if you also enter the
+override_model_delays runtime option.

+no_notifier

Suppresses the toggling of notifier registers that are optional arguments of timing check system tasks.

+no_pulse_msg

Suppresses pulse error messages, but not the generation of StX values at module path outputs when a pulse error condition occurs.

+no_tchk_msg

Disables the display of timing check warning messages but does not disable the toggling of notifier registers in timing checks. This is also a compile-time option.

+notimingcheck

Suppress timing checks.

+ntb_cache_dir=<path_name_to_directory>

Specifies the directory location of the cache that VCS maintains as an internal disk cache for randomization.

+ntb_debug_on_error

Causes the simulation to stop immediately when a simulation error is encountered. In addition to normal verification errors, This option halts the simulation in case of runtime errors as well.

+ntb_enable_solver_trace=<value>

Enables a debug mode that displays diagnostics when VCS executes a randomize() method call. Allowed values are:

0 - Do not display (default).

1 - Displays the constraints VCS is solving.

2 - Displays the entire constraint set.

+ntb_enable_solver_trace_on_failure=<value>

Enables a mode that displays trace information only when the VCS constraint solver fails to compute a solution, usually due to inconsistent constraints. When the value of the option is 2, the analysis narrows down to the smallest set of inconsistent constraints, thus aiding the debugging process. Allowed values are 0, 1, 2. The default value is 2.

+ntb_exit_on_error[=<value>]

Causes VCS to exit when value is less than 0. The value can be:

0: continue

1: exit on first error (default value)

N: exit on nth error.

When value = 0, the simulation finishes regardless of the number of errors.

+ntb_load=path_name_to_libtb.so

Specifies loading the testbench shared object file libtb.so.

+ntb_random_seed=<value>

Sets the seed value used by the top level random number generator at the start of simulation. The random(seed) system function call overrides this setting. The value can be any integer number.

+ntb_solver_mode=<value>

Allows choosing between one of two constraint solver modes. When set to 1, the solver spends more pre-processing time in analyzing the constraints, during the first call to randomize() on each class. When set to 2, the solver does minimal pre-processing, and analyzes the constraint in each call to randomize(). Default value is 2.

+ntb_stop_on_error

Causes the simulation to stop immediately when a simulation error is encountered, turning it into a cli debugging environment. In addition to normal verification errors, ntb_stop_on_error halts the simulation in case of run time errors. The default setting is to execute the remaining code within the present simulation time.

+override_model_delays

Enables you to use the +mindelays, +typdelays, or +maxdelays runtime options to specify timing for SWIFT SmartModels or Synopsys hardware models.

+sdfverbose

Enables the display of more than ten warning and ten error messages about SDF back annotation.

+typdelays

Specifies using the compiled SDF file for typical delays generated by the +allmtm compile-time option.
Also specifies using typical delays for SWIFT VMC or SmartModels or Synopsys hardware models if you also enter the +override_model_delays runtime option.

+vcs+dumparrays

Enables dumping memory and multi-dimensional array values in the VCD file. You must also have use the +memcbk compile-time option.

+vcs+dumpoff+<t>+<ht>

Turn off value change dumping ($dumpvars system task) at time <t>.
<ht> is the high 32 bits of a time value greater than 32 bits.

+vcs+dumpon+<t>+<ht>

Suppress $dumpvars system task until time <t>.
<ht> is the high 32 bits of a time value greater than 32 bits.

+vcs+dumpvarsoff

Suppress $dumpvars system tasks.

+vcs+finish+<t>+<ht>

Finish simulation at time <t>.

<ht> is the high 32 bits of a time value greater than 32 bits.

+vcs+grwavesoff

Suppress $gr_waves system tasks.

+vcs+ignorestop

Tells VCS to ignore the $stop system tasks in your source code.

+vcs+flush+log

Increases the frequency of flushing both the compilation and simulation log file buffers.

+vcs+flush+dump

Increases the frequency of flushing all the buffers for VCD files.

+vcs+flush+fopen

Increases the frequency of flushing all the buffers for files opened by the $fopen system function.

+vcs+flush+all

Shortcut option for entering all three of the +vcs+flush+log, +vcs+flush+dump, and +vcs+flush+fopen options.

+vcs+learn+pli

Keeps track of where you use ACC capabilities for debugging operations so that you can recompile your design and in the next simulation enable them only where you need them. With this option VCS writes the pli_learn.tab secondary PLI table file. You input this file when you recompile your design with the +applylearn compile-time option.

+vcs+lic+vcsi

Checks out three VCSi licenses to run VCS.

+vcsi+lic+vcs

Checks out a VCS license to run VCSi when all VCSi licenses are in use.

+vcs+lic+wait

Wait for network license if none is available when the job starts.

+vcsi+lic+wait

Tells VCSi to wait for a network license if none is available.

+vcs+mipd+noalias

If during a simulation run, acc_handle_simulated_net is called before MIPD annotation happens, a warning message is issued. When this happens you can use this option to disable such aliasing for all ports whenever mip, mipb capabilities have been specified. This option works for regular sdf annotation and not for compiled SDF.

+vcs+nostdout

Disables all text output from VCS including messages and text from $monitor and $display and other system tasks. VCS still writes this output to the log file if you include the -l option.

+vcs+stop+<t>+<ht>

Stop simulation at time <t>. <ht> is the high 32 bits of a time value greater than 32 bits (optional).
See the section on "Specifying A Long Time Before Stopping Simulation" in the VCS/VCSi User Guide.

+vera_load=<filename.vro>

Specifies the VERA object file.

+vera_mload=<filename>

Specifies a text file that contains a list of VERA object files.

Options for Specifying How VCS Writes the VPD File

****

-vpd_file <filename>

At runtime, defines an alternative name of the VPD file that VCS writes instead of the default name vcdplus.vpd.

-vpd_fileswitchsize <number_in_MB>

Specifies a size for the VPD file. When the VPD file reaches this size, VCS closes the VPD file and opens a new one with the same design hierarchy as the previous VPD file. There is a number suffix added to the VPD file name to differentiate them.

-vpd_bufsize <MB>

VCS uses an internal buffer to store value changes before it writes them to the VPD file on disk. VCS makes this buffer size either 5 MB or large enough to record 15 value changes for all nets and registers in your design, whichever is larger.
You can use this option to override the buffer size that VCS calculates for the buffer size. You specify a buffer size in megabytes.

-vpd_filesize <MB>

Specifies the maximum size of the VPD file. When VCS reaches this limit, VCS overwrites the oldest simulation history data in the file with the newest.

-vpd_noupdate

Turn file locking off when writing the VPD file. By default file locking is enabled allowing DVE to read the VPD file while the simulation is still running.

-vpd_compression low|medium|high

Use different algorithms to compress the data in VPD files.

+vpdnocompress

Disables the compression of data in VPD files.

+vpdignore

Tells VCS to ignore $vcdplus system tasks so VCS does not write a VPD file.

Options For Calling The vcd2vpd and vpd2vcd Utilities

****

-vcd2vpd <vcd_filename> <vcdplus_filename>

Tells VCS to find and run the vcd2vpd utility that converts a VCD file to a VPD file. VCS inputs to the utility the specified VCD file and the utility outputs the specified VPD file.

-vpd2vcd <vcdplus_filename> <vcd_filename>

Tells VCS to find and run the vpd2vcd utility that converts a VPD file to a VCD file. VCS inputs to the utility the specified VPD file and the utility outputs the specified VCD file.

Environment Variables

****

DISPLAY_VCS_HOME

Enables the display at compile time if the path to the directory specifies with the VCS_HOME environment.

LM_LICENSE_FILE

The complete path of the VCS license file or port@host.

PATH

On UNIX add $VCS_HOME/bin to this environment variable.

SNPS_SIM_DEFAULT_GUI

Specifies the default graphical user interface. This environment variable overrides option -gui without "=<dve|verdi>".

VC_HOME

Specifies the directory where you installed Verification Compiler.

VCS_HOME

Specifies the directory where you installed VCS.

VCSI_HOME

Specifies the directory where you installed VCSi.

TMPDIR

Specifies the directory for temporary compilation files.

VCS_CC

Specifies the C compiler.

VCS_COM

Specifies the path to the VCS compiler executable named vcs1 (or vcs1.exe).

VCS_LOG

Specifies the runtime log file name.

VCS_RUNTIME

Specifies which runtime library named libvcs.a VCS uses.

VCS_SWIFT_NOTES

Enables the printf PCL command.

VCS_WARNING_ATSTAR

Specifies the number of signals in a Verilog-2001 implicit sensitivity list that must be exceeded before VCS displays a warning. The default limit is 100 signals.