

Quick Guide

Questa SIM 2020

Key Commands

[add memory](#)

opens the specified memory in the MDI frame of the Main window

[add testbrowser](#)

adds .ucdb files to the Test Management Browser

[add watch](#)

adds signals or variables to the Watch window

[add wave](#)

adds VHDL signals and variables, and Verilog nets and registers to the Wave window

[alias](#)

creates a new Tcl procedure that evaluates the specified commands

SVA & PSL -----

[assertion active](#)

instructs the simulator to report on any active assertion directives at the end of simulation (EOS)

[atv log](#)

enables or disables assertion thread viewing (ATV) for the specified assertion(s)

[assertion count](#)

returns the sum of the assertion failure counts for the specified set of assertion directive instances

[assertion fail](#)

configures fail tracking for SystemVerilog and PSL assertions

[assertion pass](#)

configures pass tracking for SystemVerilog and PSL assertions

[assertion report](#)

produces a textual summary of SystemVerilog and PSL assertion results

[change](#)

modifies the value of a VHDL variable or Verilog register variable

[checkpoint](#)

saves the state of your simulation

[compare add](#)

compares signals in a reference design against signals in a test design

[configure](#)

invokes the List or Wave widget configure command for the current default List or Wave window

COVERAGE -----

[coverage attribute](#)

displays attributes in the currently loaded database

[coverage clear](#)

clears all coverage data obtained during previous run commands

[vcover diff](#)

reports the coverage differences between two test runs

[coverage file](#)

sets the name of the coverage data file to be automatically saved at the end of simulation

[coverage goal](#)

Sets the value of UCDB-wide goals

[coverage ranktest](#)

ranks coverage data according to user-specified tests

[coverage report](#)

produces a textual output of the coverage statistics that have been gathered up to this point

[coverage summaryinfo](#)

prints coverage numbers of the specified coverage types without loading the entire database

[coverage tag](#)

adds or removes tags from specified objects

[coverage testnames](#)

displays test names in the current UCDB file loaded

[delete](#)

removes objects from either the List or Wave window

support.mentor.com

SUPPORT

[do](#)

executes commands contained in a macro file

[drivers](#)

displays in the Main window the current value and scheduled future values for all the drivers of a specified VHDL signal or Verilog net

[dumplog64](#)

dumps the contents of the *vsim.wlf* file in a readable format

[echo](#)

displays a specified message in the Main window

[edit](#)

invokes the editor specified by the EDITOR environment variable

[environment](#)

displays or changes the current dataset and region environment

[examine](#)

examines one or more objects, and displays current values (or the values at a specified previous time) in the Main window

[find](#)

displays the full pathnames of all objects in the design whose names match the name specification you provide

[force](#)

applies stimulus to VHDL signals and Verilog nets

[history](#)

lists the commands executed during the current session

[next](#)

continues a search; see the [search](#) command

[noforce](#)

removes the effect of any active [force](#) commands on the selected object

[notepad](#)

opens a simple text editor

[printenv](#)

echoes to the Main window the current names and values of all environment

[variables](#)

[profile on](#)

enables runtime profiling of where your simulation is spending its time and where memory is allocated

[property list](#)

changes one or more properties of the specified signal, net, or register in the List Window

[property wave](#)

changes one or more properties of the specified signal, net, or register in the Wave Window

[pwd](#)

displays the current directory path in the Main window

[qverilog](#)

compiles, optimizes, and simulates a Verilog or SystemVerilog design in one step

[radix](#)

specifies the default radix to be used

[report](#)

displays the value of all simulator control variables, or the value of any simulator state variables relevant to the current simulation

[restart](#)

reloads the design elements and resets the simulation time to zero

[restore](#)

restores the state of a simulation that was saved with a [checkpoint](#) command during the current invocation of vsim

[resume](#)

resumes execution of a macro file after a [pause](#) command or a breakpoint

[right](#)

searches right (next) for signal transitions or values in the specified Wave window

[run](#)

advances the simulation by the specified number of timesteps

[sccom](#)

compiles SystemC design units

[sdfcom](#)

compiles SDF files

[search](#)

searches the specified window for one or more objects matching the specified pattern(s)

[seetime](#)

scrolls the List or Wave window to make the specified time visible

[ucdb2html](#)

converts a .ucdb file into HTML

Questa SIM 2020

Quick Guide

[vcd dumpports](#)

creates a VCD file that captures port driver data

[vcd2wlf](#)

translates VCD files into WLF files

[vcom](#)

compiles VHDL design units

[vcover attribute](#)

displays attributes in the currently loaded database

[vcover merge](#)

merges multiple code or functional coverage data files offline

[vcover ranktest](#)

ranks the specified input files according to their contribution to cumulative coverage

[vcover report](#)

reports on multiple code or functional coverage data files offline

produces summary statistics from multiple coverage data files

[vcover testnames](#)

displays test names in the current UCDB file loaded

[vdel](#)

deletes a design unit from a specified library

[vdir](#)

lists the contents of a design library

[verror](#)

prints a detailed description of a message number

[vgencomp](#)

writes a Verilog module's equivalent VHDL component declaration to standard

[output](#)

[view](#)

opens a Questa SIM 2020 window and brings it to the front of the display

[vlib](#)

creates a design library

[vlog](#)

compiles Verilog design units and SystemVerilog extensions

[vmake](#)

creates a makefile that can be used to reconstruct the specified library

[vmap](#)

defines a mapping between a logical library name and a directory

[vopt](#)

produces an optimized version of your design

[vsim](#)

loads a new design into the simulator

[when](#)

instructs Questa SIM 2020 to perform actions when the specified conditions are met

[where](#)

displays information about the system environment

[wlf2log](#)

translates a Questa SIM 2020 WLF file to a QuickSim II logfile

[wlf2vcd](#)

translates a Questa SIM 2020 WLF file to a VCD file

[wlfinan](#)

outputs information about or a new WLF file from an existing WLF file

[xml2ucdb](#)

creates an HTML report of code coverage from a .ucdb file

Key Command Arguments

Use <command> -help for a full list.

QVERILOG

The qverilog command compiles, optimizes, and simulates Verilog and SystemVerilog designs in a single step.

1. automatic work library creation
2. support for all standard vlog arguments
3. support for C/C++ files via the SystemVerilog DPI
4. implicit “run -all; quit” unless using -i, -gui, -do (see -R below)
5. vopt performance invoked (see the vopt section of this guide)

Key arguments to qverilog

<filename> Verilog source code file to compile, one is required
[-R <sim_options>] vsim command options applied to simulation

SCCOM

-link Links source code, required
[CPP option] C++ compiler option
[-g] Compile with debugging info
-vv Echo subprocess invocations on stdout
[-scv] Includes SystemC verification library
<filename(s)> SystemC files to be compiled

VCOM

[-2008 | -2002 | -93 | -87] Choose VHDL 2008, 2002, 1993, or 1987
[-check_synthesis] Turn on synthesis checker
[-debugVA] Print VITAL opt status
[-explicit] Resolve ambiguous overloads
[-help] Display vcom syntax help
[-f <filename>] Pass in arguments from file
[-norangecheck] Disable run time range checks
[-nodebug] Hide internal variables & structure
[-novitalcheck] Disable VITAL95 checking
[-nowarn <#>] Disable individual warning msg
[-quiet] Disable loading messages
[-refresh] Regenerate library image
[-version] Returns vcom version
[-work <libname>] Specify work library
<filename(s)> VHDL file(s) to be compiled

VLOG

[-vlog95compat] Disable Verilog 2001 keywords
[-compat] Disable event order optimizations
[-f <filename>] Pass in arguments from file
[-hazards] Enable run-time hazard checking
[-help] Display vlog syntax help
[-nodebug] Hide internal variables & structure
[-quiet] Disable loading messages
[-R <simargs>] Invoke VSIM after compile
[-refresh] Regenerate lib to current version
[-sv] Enables SystemVerilog keywords
[-version] Returns vlog version
[-v <library_file>] Specify Verilog source library
[-work <libname>] Specify work library
<filename(s)> Verilog file(s) to be compiled

VOPT

Design optimization options

1. Optimized designs simulate faster, while non-optimized designs provide object visibility for debugging.
2. QIS is the preferred method to use when optimizing designs with vopt for debug/visibility purposes. See “Questa Dumping and Visibility” in the User’s Manual.
2. Use +acc with vopt or vsim -voptargs with +acc for selective design object visibility during debugging.
3. Read “Optimizing Designs with vopt” in the User’s Manual for additional information.

Key arguments to vopt

-o <name> Optimized design name
<design> Top-level design unit
[-debug] Generates debug databases
[-debug.cell] Generates debug databases with cell internals
[-debug.livesim] For Visualizer LiveSim debugging.
[-access=+<instance/module>] Provides read visibility to signals and scopes.
[-access=w+<instance/module>] Provides write visibility to signals and scopes.
[-cellaccess=+<instance/module>] Read visibility to cell internals.
[-cellaccess=w+<instance/module>] Write visibility to cell internals.
[+acc=[<spec>]+<module>]] Enable design object visibility
+cover=bcefsx Specifies coverage type(s)
-nocover Disable coverage on all source files
-g Assigns a value to generics and parameters with no value
-G Forces value assignment for generics and parameters

Key arguments to vsim

[-vopt] Run vopt if not automatically invoked
[-voptargs=”<args>”] Arguments passed to vopt, use +acc args for design visibility

VSIM

[-assertdebug] Keep data for debugging assertion failures
[-assertfile <filename>] Alternative file for recording assert messages
[-assume] Simulate PSL and Verilog assume directives same as assert directives[-c] Run in cmd line mode
[-coverage] Invoke Code Coverage
[-do “cmd” | <file>] Run cmd or file at startup
[-elab] Create elaboration file
[-f <filename>] Pass in args from file
[-gG<name=value>] Set VHDL Generic values
[-hazards] Enable hazard checking
[-help] Display vsim syntax help
[-l <logfile>] Save transcript to log file
[-lineinfo] Provides complete file or line information in vsim
[-load_elab] Simulate an elaboration file
[-noassume] Do not simulate PSL and Verilog assume directives
[-nops] Disable PSL assertions
[-nosva] Disable System Verilog concurrent assertions
[+notimingchecks] Disable timing checks
[-qfd] Enables visibility clients (Tcl, VPI, UVM backdoor)
[-quiet] Disable loading messages
[-restore <filename>] Restore a simulation
[-sdf {min|typ|max} <region>=<sdf file>] Apply SDF timing data e.g., sdfmin /top=MySDF.txt
[-sdfnowarn] Disable SDF warnings
[-sv_seed <seed>] Specify a seed for the Random Number

Generator of the root thread
[-t [<mult>]<unit>] Time resolution
[-vcdstim [<instance>=<filename>]] Stimulate the top-level design or instances from an Extended VCD file
[-version] Returns vsim version
[-vopt] Run vopt automatically
[-voptargs=”<args>”] Arguments to pass to vopt
[-view <filename>] Log file for VSIM to view
[-wlf <filename>] Log file to create
[-libname>.<design_unit>] Configuration, Module, Entity/Arch, or optimized design to simulate
[-wlf cachesize] Specify WLF reader cache size (per WLF file.)
[-wlfslim <size>] Specify the number of Megabytes to be saved in event log file
[-wlfslim <duration>] Specify the duration of time to be saved in event log file

Code Coverage

Key Arguments to vcom/vlog

+cover=bcefsx Specifies coverage type(s)

Key Arguments to vopt

+cover=bcefsx Specifies coverage type(s)
-nocover Disable coverage on all source files

Key Arguments to vsim

-coverage Enables statistics collection

SVA & PSL

Key arguments to vcom and vlog

[-pslfile <file>] External PSL file
[-nops] Ignore embedded PSL assertions

Key arguments to vsim

[-nops] Ignore embedded PSL assertions
[-nosva] Ignore SystemVerilog concurrent assertions

Key modelsim.ini variables

AssertionFail* Control assertion failure behavior
AssertionFormat* Define messages for VHDL assertion types
AssertionPass* Control assertion pass behavior
BreakOnAssertion Stop the simulator after assertion message
Cover* Control cover directive behavior
IgnoreSVA* Control SVA message logging
Sv_Seed Seed random number generator

Wave Window

add wave <item> Wave specific signals/nets
add wave * Wave signals/nets in scope
add wave -r /* Wave all signals/nets in design
add wave abus(31:15) Wave a slice of a bus
view wave Display wave window
view wave -new Display additional wave window
write wave Print wave window to file
<left mouse button> Select signal / Place cursor
<middle mouse button> Zoom options
<right mouse button> Context Menu
<ctrl-f> Find next item
<tab> (go right) Search forward for next edge