OpenSTA

Command Line Arguments

The command line arguments for `sta` are shown below.

```
sta [-help] [-version] [-no_init]
  -help              show help and exit
  -version           show version and exit
  -no_init           do not read ~/.sta
  -no_splash         do not print the splash message
  -threads count|max use count threads
  -exit              exit after reading cmd_file
  cmd_file           source cmd_file
```

When the STA starts up commands are first read from the user initialization file `~/.sta` if it exists. If a TCL command file *`cmd_file`* is specified, commands are read from the file and executed before entering an interactive TCL command interpreter. If -exit is specified the application exits after reading *`cmd_file`*. Use the TCL `exit` command to exit the application. The `-threads` option specifies how many parallel threads to use. Use `-threads max` to use one thread per processor.

A sample command file that reads a library and a Verilog netlist and reports timing checks is shown below.

```
read_liberty example1_slow.lib
read_verilog example1.v
link_design top
read_sdf example1.sdf
create_clock -name clk -period 10 {clk1 clk2 clk3}
set_input_delay -clock clk 0 {in1 in2}
report_checks
```

This example can be found in `examples/example1.tcl`.

Any number of Liberty and Verilog files can be read before linking the design. After linking the design, SDF or parasitics can be read.

An example command script using three process corners and +/-10% min/max derating is shown below.

```
define_corners wc typ bc
read_liberty -corner wc  example1_slow.lib
read_liberty -corner typ example1_typ.lib
read_liberty -corner bc  example1_fast.lib
read_verilog example1.v
link_design top
set_timing_derate -early 0.9
set_timing_derate -early 1.1
create_clock -name clk -period 10 {clk1 clk2 clk3}
set_input_delay -clock clk 0 {in1 in2}
report_checks -path_delay min_max
report_checks -corner typ
```

This example can be found in `examples/example5.tcl`. Other examples can be found in the directory `examples`.

TCL Interpreter

Keyword arguments to commands may be abbreviated. For example,

```
report_checks -unique
```

is equivalent to the following command.

```
report_checks -unique_paths_to_endpoint
```

Commands

`all_clocks`

The `all_clocks` command returns a list of all clocks that have been defined.

`all_inputs`

The `all_inputs` command returns a list of all input and bidirect ports of the current design.

`all_outputs`

The `all_outputs` command returns a list of all output and bidirect ports of the design.

`all_registers`	`[-clock` *`clock_names`*`]`
	`[-cells]`
	`[-data_pins]`
	`[-clock_pins]`
	`[-async_pins]`
	`[-output_pins]`
	`[-level_sensitive]`
	`[-edge_triggered]`

`clock_names`	`A list of clock names. Only registers clocked by these clocks are returned.`
`-cells`	`Return a list of register instances.`
`-data_pins`	`Return the register data pins.`
`-clock_pins`	`Return the register clock pins.`

-async_pins	Return the register set/clear pins.
-output_pins	Return the register output pins.
-level_sensitive	Return level-sensitive latches.
-edge_triggered	Return edge-triggered registers.

The all_registers command returns a list of register instances or register pins in the design. Options allow the list of registers to be restricted in various ways. The -clock keyword restrcts the registers to those that are clocked by a set of clocks. The -cells option returns the list of registers or latches (the default). The -data_pins, -clock_pins, -async_pins and -output_pins options cause all_registers to return a list of register pins rather than instances.

---

check_setup	[-verbose] [-unconstrained_endpoints] [-multiple_clock] [-no_clock] [-no_input_delay] [-loops] [-generated_clocks] [> *filename*] [>> *filename*]
-verbose	Show offending objects rather than just error counts.
-unconstrained_endpoints	Check path endpoints for timing constraints (timing check or set_output_delay).
-multiple_clock	Check register/latch clock pins for multiple clocks.
-no_clock	Check register/latch clock pins for a clock.
-no_input_delay	Check for inputs that do not have a set_input_delay command.
-loops	Check for combinational logic loops.
-generated_clocks	Check that generated clock source pins have been defined as clocks.

The check_setup command performs sanity checks on the design. Individual checks can be performed with the keywords. If no check keywords are specified all checks are performed.

---

connect_pin	*net* *port*|*pin*
net	A net to add connections to.
port	A port to connect to *net*.
Pin	A pin to connect to *net*.

The connect_pin command connects a port or instance pin to a net.

---

create_clock	-period period [-name clock_name] [-waveform edge_list] [-add] [pin_list]
period	The clock period.
clock_name	The name of the clock.
edge_list	A list of edge rise and fall time.
-add	Add this clock to the clocks on *pin_list*.
pin_list	A list of pins driven by the clock.

The create_clock command defines the waveform of a clock used by the design.

If no *pin_list* is specified the clock is *virtual*. A virtual clock can be refered to by name in input arrival and departure time commands but is not attached to any pins in the design.

If no clock name is specified the name of the first pin is used as the clock name.

If a wavform is not specified the clock rises at zero and falls at half the clock period. The waveform is a list with time the clock rises as the first element and the time it falls as the second element.

If a clock is already defined on a pin the clock is redefined using the new clock parameters. If multiple clocks drive the same pin, use the -add option to prevent the existing definition from being overwritten.

The following command creates a clock with a period of 10 time units that rises at time 0 and falls at 5 time units on the pin named clk1.

```
create_clock -period 10 clk1
```

The following command creates a clock with a period of 10 time units that is high at time zero, falls at time 2 and rises at time 8. The clock drives three pins named clk1, clk2, and clk3.

```
create_clock -period 10 -waveform {8 2} -name clk {clk1 clk2 clk3}
```

---

create_generated_clock	[-name *clock_name*] -source *master_pin* [-master_clock *master_clock*] [-pll_out *pll_out_pin*] [-pll_feedback *pll_fdbk_pin*] [-divide_by *divisor*] [-multiply_by *multiplier*] [-duty_cycle duty_cycle] [-invert] [-edges *edge_list*] [-edge_shift *shift_list*] [-add] *pin_list*
clock_name	The name of the generated clock.
master_pin	A pin or port in the fanout of the master clock that is the source of the generated clock.
master_clock	Use -master_clock to specify which source clock to use when multiple clocks are present on *master_pin*.
pll_out_pin	The pin from *pin_list* that is the phase locked loop output pin.
pll_fdbk_pin	A pin in the fanout of the *pll_out_pin* that the phased locked loop phase locks to the master clock pin.
divisor	Divide the master clock period by *divisor*.
multiplier	Multiply the master clock period by *multiplier*.
duty_cycle	The percent of the period that the generated clock is high (between 0 and 100).
-invert	Invert the master clock.
edge_list	
shift_list	Not supported.
-add	Add this clock to the existing clocks on *pin_list*.
pin_list	A list of pins driven by the generated clock.

The `create_generated_clock` command is used to generate a clock from an existing clock definition. It is used to model clock generation circuits such as clock dividers and phase locked loops.

The `-source`, `-pll_out` and `-pll_feedback` must all be pins on the same PLL instance. The delay between the PLL out and feedback pins is removed from the source latency of the generated clock.

The `-divide_by`, `-multiply_by` and `-edges` arguments are mutually exclusive.

The `-multiply_by` option is used to generate a higher frequency clock from the source clock. The period of the generated clock is divided by *`multiplier`*. The clock *`multiplier`* must be a positive integer. If a duty cycle is specified the generated clock rises at zero and falls at period * duty_cycle / 100. If no duty cycle is specified the source clock edge times are divided by *`multiplier`*.

The `-divide_by` option is used to generate a lower frequency clock from the source clock. The clock *`divisor`* must be a positive integer. If the clock divisor is a power of two the source clock period is multiplied by *`divisor`*, the clock rise time is the same as the source clock, and the clock fall edge is one half period later. If the clock divisor is not a power of two the source clock waveform edge times are multiplied by *`divisor`*.

The `-edges` option forms the generated clock waveform by selecting edges from the source clock waveform.

If the `-invert` option is specified the waveform derived above is inverted.

If a clock is already defined on a pin the clock is redefined using the new clock parameters. If multiple clocks drive the same pin, use the `-add` option to prevent the existing definition from being overwritten.

In the example show below generates a clock named `gclk1` on register output pin `r1/Q` by dividing it by four.

```
create_clock -period 10 -waveform {1 8} clk1
create_generated_clock -name gclk1 -source clk1 -divide_by 4 r1/Q
```

The generated clock has a period of 40, rises at time 1 and falls at time 21.

In the example shown below the duty cycle is used to define the derived clock waveform.

```
create_generated_clock -name gclk1 -source clk1 -duty_cycle 50 \
                       -multiply_by 2 r1/Q
```

The generated clock has a period of 5, rises at time .5 and falls at time 3.

In the example shown below the first, third and fifth source clock edges are used to define the derived clock waveform.

```
create_generated_clock -name gclk1 -source clk1 -edges {1 3 5} r1/Q
```

The generated clock has a period of 20, rises at time 1 and falls at time 11.

`create_voltage_area`

`[-name` *`name`*`]`
`[-coordinate` *`coordinates`*`]`
`[-guard_band_x` *`guard_x`*`]`
`[-guard_band_y` *`guard_y`*`]`
`cells`

This command is parsed and ignored by timing analysis.

---

current_design [*design*]

---

current_instance [*instance*]

instance Not supported.

---

define_corners *corner_name1 [corner_name2]...*

corner_name The name of a delay calculation corner.

Use the define_corners command to define the names of multiple process/temperature/voltage corners. The define_corners command must follow set_operating_conditions -analysis_type and precede any reference to the corner names and can only appear once in a command file. There is no support for re-defining corners.

For analysis type single, each corner has one delay calculation result and early/late path arrivals. For analysis type best_case/worst_case and on_chip_variation, each corner has min/max delay calculation results and early/late path arrivals.

---

delete_clock [-all] *clocks*

clocks A list of clocks to remove.

---

delete_from_list *list objects*

list A list of objects.

objects A list of objects to delete from *list*.

---

delete_generated_clock [-all] *clocks*

clocks A list of generated clocks to remove.

---

delete_instance *instance*

instance An instance to delete.

The network editing command `delete_instance` removes an instance from the design.

---

delete_net *net*

net A nets to delete.

The network editing command `delete_net` removes a net from the design.

---

disconnect_pin *net*
port|*pin*|-all

net The net to disconnect pins from.

port A port to connect to *net*.

pin A pin to connect to *net*.

-all Disconnect all pins from the net.

Disconnects a port or pin from a net. Parasitics connected to the pin are deleted.

---

elapsed_run_time

Returns the total clock run time in seconds as a float.

find_timing_paths	[-from *from_list*
	  |-rise_from *from_list*
	  |-fall_from *from_list*]
	[-through *through_list*
	  |-rise_through *through_list*
	  |-fall_through *through_list*]
	[-to *to_list*
	  |-rise_to *to_list*
	  |-fall_to *to_list*]
	[-unconstrained]
	[-path_delay min|min_rise|min_fall
	  |max|max_rise|max_fall
	  |min_max]
	[-group_count *path_count*]
	[-endpoint_count *endpoint_path_count*]
	[-unique_paths_to_endpoint]
	[-corner *corner_name*]
	[-slack_max *max_slack*]
	[-slack_min *min_slack*]
	[-sort_by_slack]
	[-path_group *group_names*]

from_list	A list of clocks, instances, ports or pins.
through_list	A list of instances, pins or nets.
to_list	A list of clocks, instances, ports or pins.
-unconstrained	Return unconstrained paths.
-path_delay min	Report min path (hold) checks.
-path_delay min_rise	Report min path (hold) checks for rising endpoints.
-path_delay min_fall	Report min path (hold) checks for falling endpoints.
-path_delay max	Report max path (setup) checks.
-path_delay max_rise	Report max path (setup) checks for rising endpoints.
-path_delay max_fall	Report max path (setup) checks for falling endpoints.
-path_delay min_max	Report max and max path (setup and hold) checks.

path_count	The number of paths to report in each path group.
endpoint_path_count	The number of paths to report for each endpoint.
-unique_paths_to_endpoint	Report multiple paths to an endpoint that traverse different pins without showing multiple paths with different rise/fall transitions.
corner_name	Only report paths for one process corner.
max_slack	Only report paths with less slack than *max_slack*.
min_slack	Only report paths with more slack than *min_slack*.
-sort_by_slack	Sort paths by slack rather than slack grouped by path group.
group_names	List of path group names to report. All path groups are reported if this option is not specified.

The find_timing_paths command returns a list of path objects for scripting. Use the get_property function to access properties of the paths.

---

get_cells	[-hierarchical] [-hsc *separator*] [-filter *expr*] [-regexp] [-nocase] [-quiet] [-of_objects *objects*] [*patterns*]
-hierarchical	Searches hierarchy levels below the current instance for matches.
separator	Character to use to separate hierarchical instance names in *patterns.*
expr	A filter expression of the form *attribute* ==|~= *pattern* where *attribute* is an attribute supported by the get_property command.

objects	The name of a pin or net, a list of pins returned by get_pins, or a list of nets returned by get_nets. The –hierarchical option cannot be used with –of_objects.
patterns	A list of cell (instance) name patterns.

The get_cells command returns a list of all cell instances that match *patterns*.

Without –regexp Unix style file glob pattern matching is used. With –regexp TCL regular expression matching is used. When –nocase is used regular expressions are case insensitive. The –nocase flag can only be used with –regexp.

---

get_clocks

[-regexp]
[-nocase]
[-quiet]
patterns

-quiet	Do not report an error if *patterns* do not match anything.
patterns	A list of clock name patterns.

The get_clocks command returns a list of all clocks that have been defined.

Without –regexp Unix style file glob pattern matching is used. With –regexp TCL regular expression matching is used. When –nocase is used regular expressions are case insensitive. The –nocase flag can only be used with –regexp.

---

get_fanin

-to *sink_list*
[-flat]
[-only_cells]
[-startpoints_only]
[-levels *level_count*]
[-pin_levels *pin_count*]
[-trace_arcs timing|enabled|all]

sink_list	List of pins, ports, or nets to find the fanin of. For nets, the fanin of driver pins on the nets are returned.
-flat	Without –flat only pins at the same hierarchy level as the sinks are returned. With –flat pins in the fanin at any hierarchy level are returned.
-only_cells	Return the instances connected to the pins in the fanin.
-startpoints_only	Only return pins that are startpoints.
level_count	Only return pins within *level_count* instance traversals.
pin_count	Only return pins within *pin_count* pin traversals.
-trace_arcs	With 'timing' and 'enabled' values only arcs that are not disabled are traversed. With a value of 'all' even disabled arcs are traversed.

The get_fanin command returns traverses the design from *sink_list* pins, ports or nets backwards and return the fanin pins or instances.

---

get_fanout

-from *source_list*
[-flat]
[-only_cells]
[-endpoints_only]
[-levels *level_count*]
[-pin_levels *pin_count*]
[-trace_arcs timing|enabled|all]

source_list	List of pins, ports, or nets to find the fanout of. For nets, the fanout of load pins on the nets are returned.
-flat	Without –flat only pins at the same hierarchy level as the sinks are returned. With –flat pins in the fanout at any hierarchy level are returned.
-only_cells	Return the instances connected to the pins in the fanout.
-endpoints_only	Only return pins that are endpoints.
level_count	Only return pins within *level_count* instance traversals.
pin_count	Only return pins within *pin_count* pin traversals.
-trace_arcs	With 'timing' and 'enabled' values only arcs that are not disabled are traversed. With a value of 'all' even disabled arcs are traversed.

The get_fanout command returns traverses the design from *source_list* pins, ports or nets backwards and return the fanout pins or instances.

---

get_lib_cells	[-of_objects *objects*] [-hsc *separator*] [-regexp] [-nocase] [-quiet] *patterns*

---

objects	A list of cell (instance) objects.
separator	Character that separates the library name and cell name in *pattern*.
-quiet	Do not report an error if *patterns* do not match anything.
patterns	A list of library cell name patterns of the form library_name/cell_name.

The get_lib_cells command returns a list of library cells that match *pattern*. The library name can be prepended to the cell name pattern with the *separator* character, which defaults to hierarchy_separator.

Without -regexp Unix style file glob pattern matching is used. With -regexp TCL regular expression matching is used. When -nocase is used regular expressions are case insensitive. The -nocase flag can only be used with –regexp.

---

get_lib_pins	[-hsc *separator*] [-regexp] [-nocase] [-quiet] *patterns*
separator	Character that separates the library name, cell name and port name in *pattern*.
-quiet	Do not report an error if *patterns* do not match anything.
patterns	A list of library port name patterns of the form library_name/cell_name/port_name.

The get_lib_pins command returns a list of library ports that match *pattern*. Use *separator* to separate the library and cell name patterns from the port name in *pattern*.

Without -regexp Unix style file glob pattern matching is used. With -regexp TCL regular expression matching is used. When -nocase is used regular expressions are case insensitive. The -nocase flag can only be used with –regexp.

---

get_libs	[-regexp] [-nocase] [-quiet] *patterns*
-quiet	Do not report an error if *patterns* do not match anything.
patterns	A list of library name patterns.

The get_libs command returns a list of clocks that match *patterns*.

Without -regexp Unix style file glob pattern matching is used. With -regexp TCL regular expression matching is used. When -nocase is used regular expressions are case insensitive. The -nocase flag can only be used with –regexp.

get_nets	[-hierarchical] [-hsc *separator*] [-regexp] [-nocase] [-quiet] [-of_objects *objects*] [*patterns*]
-hierarchical	Searches hierarchy levels below the current instance for matches.
separator	Character that separates hierarchical instance names and the net name in *pattern*.
-quiet	Do not report an error if *patterns* do not match anything.
objects	The name of a pin or instance, a list of pins returned by get_pins, or a list of instances returned by get_cells. The –hierarchical option cannot be used with –of_objects.
patterns	A list of net name patterns.

The get_nets command returns a list of all nets that match *patterns*.

Without –regexp Unix style file glob pattern matching is used. With –regexp TCL regular expression matching is used. When –nocase is used regular expressions are case insensitive. The –nocase flag can only be used with –regexp.

get_full_name	*object*
object	A library, cell, port, instance, pin or timing arc object.

Return the name of *object*. Equivalent to [get_property *object* full_name].

get_name	*object*
object	A library, cell, port, instance, pin or timing arc object.

Return the name of *object*. Equivalent to [get_property *object* name].

get_pins [-hierarchical]
[-hsc *separator*]
[-filter *expr*]
[-regexp]
[-nocase]
[-quiet]
[-of_objects *objects*]
[*patterns*]

-hierarchical	Searches hierarchy levels below the current instance for matches.
separator	Character that separates hierarchical instance names and the port name in *pattern*.
-quiet	Do not report an error if *patterns* do not match anything.
expr	A filter expression of the form *attribute* ==|~= *pattern* where *attribute* is an attribute supported by the get_property command.
objects	The name of a net or instance, a list of nets returned by get_nets, or a list of instances returned by get_cells. The –hierarchical option cannot be used with –of_objects.
patterns	A list of pin name patterns.

The get_pins command returns a list of all instance pins that match *patterns*.

Without –regexp Unix style file glob pattern matching is used. With –regexp TCL regular expression matching is used. When –nocase is used regular expressions are case insensitive. The –nocase flag can only be used with -regexp.

A useful idiom to find the driver pin for a net is the following.

```
get_pins -of_objects [get_net net_name] -filter “direction == output”
```

get_ports [-filter *expr*]
[-regexp]
[-nocase]
[-quiet]
[-of_objects *objects*]
[*patterns*]

-quiet	Do not report an error if *patterns* do not match any clocks.
expr	A filter expression of the form *attribute* ==|~= *pattern* where *attribute* is an attribute supported by the get_property command.
objects	The name of a net, or a list of nets returned by get_nets.
patterns	A list of port name patterns.

The get_ports command returns a list of all top level ports that match *patterns*.

Without –regexp Unix style file glob pattern matching is used. With –regexp TCL regular expression matching is used. When –nocase is used regular expressions are case insensitive. The –nocase flag can only be used with –regexp.

---

get_property	[-object_type *object_type*] *object* *property*
object_type	cell|pin|net|port|clock|timing_arc
object	An object or object name. -object_type is required if *object* is an object name.
property	A property name.

The properties for different objects types are shown below.

cell

```
base_name
filename
full_name
library
name
```

clock

```
full_name
is_generated
name
period
propagated
sources
```

edge

```
delay_max_fall
delay_min_fall
delay_max_rise
delay_min_rise
full_name
from_pin
sense
to_pin
```

instance (SDC cell)

```
cell
full_name
ref_name
liberty_cell
name
```

liberty_cell

```
area
base_name
dont_use
filename
full_name
is_buffer
is_inverter
library
name
```

liberty_library

```
filename
full_name
name
```

liberty_port

```
capacitance
direction
drive_resistance
drive_resistance_fall_max
drive_resistance_fall_min
drive_resistance_rise_max
drive_resistance_rise_min
full_name
intrinsic_delay
intrinsic_delay_fall_max
intrinsic_delay_fall_min
intrinsic_delay_rise_max
intrinsic_delay_rise_min
is_register_clock
lib_cell
name
```

library

```
name
full_name
```

net

```
full_name
name
```

path (PathEnd)

```
endpoint
endpoint_clock
endpoint_clock_pin
slack
startpoint
startpoint_clock
points
```

pin

```
activity
actual_fall_transition_max
actual_fall_transition_min
actual_rise_transition_max
actual_rise_transition_min
clocks
direction
full_name
is_register_clock
lib_pin_name
max_fall_slack
max_rise_slack
min_fall_slack
min_rise_slack
name
```

port

```
activity
actual_fall_transition_min
actual_fall_transition_max
actual_rise_transition_min
actual_rise_transition_max
direction
full_name
liberty_port
min_fall_slack
max_fall_slack
min_rise_slack
max_rise_slack
name
```

point (PathRef)

arrival
pin
required
slack

---

get_timing_edges

[-from *from_pins*]
[-to *to_pins*]
[-of_objects *objects*]
[-filter *expr*]
[*patterns*]

from_pin — A collection of pins.

to_pin — A collection of pins.

objects — A collection of instances or library cells. The –from and -to options cannot be used with –of_objects.

expr — A filter expression of the form
attribute ==|~= *pattern*
where *attribute* is an attribute supported by the get_property command.

The get_timing_edges command returns a collection of timing edges (arcs) to, from or between pins. The result can be passed to get_property or set_disable_timing.

---

group_path

-name *group_name*
[-weight *weight*]
[-critical_range *range*]
[-from *from_list*]
[-rise_from *from_list*]
[-fall_from *from_list*]
[-through *through_list*]
[-rise_through *through_list*]
[-fall_through *through_list*]
[-to *to_list*]
[-rise_to *to_list*]
[-fall_to *to_list*]

group_name — The name of the path group.

weight — Not supported.

range — Not supported.

from_list — A list of clocks, instances, ports or pins.

through_list A list of instances, pins or nets.

to_list A list of clocks, instances, ports or pins.

The group_path command is used to group paths reported by the report_checks command. See set_false_path for a description of allowed *from_list*, *through_list* and *to_list* objects.

---

link [*cell_name*]

Alias for link_design.

---

link_design [*cell_name*]

Link (elaborate, flatten) the design with *cell_name* as the top level cell. The design must be linked after reading netlist and library files. The default value of *cell_name* is the current design.

The linker creates empty "block box" cells for instances the reference undefined cells when the variable link_create_black_boxes is true. When link_create_black_boxes is false an error is reported and the link fails.

The link_design command returns 1 if the link succeeds and 0 if it fails.

---

make_instance *inst_path*
lib_cell

inst_path A hierarchical instance name.

lib_cell The library cell of the new instance.

The make_instance command makes an instance of library cell *lib_cell*.

---

make_net *net_name_list*

net_name_list A list of net names.

Creates a net for each hierarchical net name.

read_liberty	[-corner *corner_name*] [-min] [-max] [-no_latch_infer] *filename*
corner_name	Use library for process corner *corner_name* delay calculation.
-min	Use library for min delay calculation.
-max	Use library for max delay calculation.
filename	The name of the liberty library file to read.

The read_liberty command reads a Liberty format library file. The first library that is read sets the units used by SDC/TCL commands and reporting. The `include_file` attribute is supported.

Cells that have a triad of timing arcs between three pins as shown below are inferred as latches:

```
cell (infered_latch) {
  pin(D) {
    direction : input ;
    timing () {
      related_pin : "E" ;
      timing_type : setup_falling ;
    }
    timing () {
      related_pin : "E" ;
      timing_type : hold_falling ;
    }
  }
  pin(E) {
    direction : input;
  }
  pin(Q) {
    direction : output ;
    timing () {
      related_pin : "D" ;
    }
    timing () {
      related_pin : "E" ;
      timing_type : rising_edge ;
    }
  }
}
```

In this example a positive level-sensitive latch is inferred.

When the `read_liberty -no_latch_infer` flag is used latches are not inferenced. If a cell has the `interface_timing true` attribute, no latches are inferred in the cell.

Files compressed with gzip are automatically uncompressed.

---

read_sdc	[-echo] *filename*
-echo	Print each command before evaluating it.
filename	SDC command file.

Read SDC commands from *filename*.

The read_sdc command stops and reports any errors encountered while reading a file unless sta_continue_on_error is 1.

Files compressed with gzip are automatically uncompressed.

---

read_sdf	[-corner *corner_name*] [-unescaped_dividers] *filename*
corner_name	Process corner delays to annotate.
-unescaped_dividers	With this option path names in the SDF do not have to escape hierarchy dividers when the path name is escaped. For example, the escaped Verilog name "\inst1/inst2 " can be referenced as "inst1/inst2". The correct SDF name is "inst1\/inst2", since the divider does not represent a change in hierarchy in this case.
filename	The name of the SDF file to read.

Read SDF delay values from a file. The min and max values in the SDF tuples are used to annotate the delays for *corner*. The typical values in the SDF tuples are ignored. If multiple corners are defined -corner must be specified.

Files compressed with gzip are automatically uncompressed.

INCREMENT is supported as an alias for INCREMENTAL.

The following SDF statements are not supported.

```
PORT
INSTANCE wildcards
```

read_spef	[-min] [-max] [-path path] [-corner *corner*] [-keep_capacitive_coupling] [-coupling_reduction_factor *factor*] [-reduce_to pi_elmore|pi_pole_residue2] [-delete_after_reduce] [-quiet] *filenames*
corner	The name of a process corner.
path	Hierarchical instance path to annotate with parasitics.
-keep_capacitive_coupling	Keep coupling capacitors in parasitic networks rather than converting them to grounded capacitors.
factor	Factor to multiply coupling capacitance by when reducing parasitic networks.
-reduce_to	Reduce detailed parasitics to a PI/Elmore or PI/Pole residue model as each net is read.
-delete_after_reduce	Delete the detailed parasitic network after reducing it.
-quiet	Do not print error or warning messages.
-save	Save the parasitics database after reading it (OpenAccess only).
filename	The name of the parasitics file to read.

The read_spef command reads a file of net parasitics in SPEF format. Use the -report_parasitic_annotation command to check for nets that are not annotated.

Files compressed with gzip are automatically uncompressed.

Separate parasitics can be annotated for corners and min and max paths using the -corner, –min and -max arguments.

With the -reduce_to and -delete_after_reduce options, parasitic networks are reduced after each net is read, substantially reducing the memory footprint required to store the parasitics.

Coupling capacitors are multiplied by the `-coupling_reduction_factor` when a parasitic network is reduced.

The following SPEF constructs are ignored.

```
*DESIGN_FLOW (all values are ignored)
*S slews
*D driving cell
*I pin capacitances (library cell capacitances are used instead)
*Q r_net load poles
*K r_net load residues
```

If the SPEF file contains triplet values the first value is used.

Parasitic networks (DSPEF) can be annotated on hierarchical blocks using the -path argument to specify the instance path to the block. Parasitic networks in the higher level netlist are stitched together at the hierarchcal pins of the blocks.

---

read_power_activities	[-scope *scope*] -vcd *filename*
scope	The VCD scope of the current design to extract simulation data. Typically the test bench name and design under test instance name. Scope levels are separated with '/'.
filename	The name of the VCD file to read.

The `read_power_activities` command reads a VCD (Value Change Dump) file from a Verilog simulator and extracts pin activities and duty cycles for use in power estimation. Files compressed with gzip are supported.

---

read_verilog	*filename*
filename	The name of the verilog file to read.

The `read_verilog` command reads a gate level verilog netlist. After all verilog netlist and Liberty libraries are read the design must be linked with the `link_design` command.

Verilog 2001 module port declaratation support has been added. An example is shown below.

```
module top (input in1, in2, clk1, clk2, clk3,
            output out);
```

Files compressed with gzip are automatically uncompressed.

replace_cell	*instance_list* *replacement_cell*
instance_list	A list of instances to swap the cell.
replacement_cell	The replacement lib cell.

The replace_cell command changes the cell of an instance. The replacement cell must have the same port list (number, name, and order) as the instance's existing cell for the replacement to be successful.

report_annotated_check	[-setup] [-hold] [-recovery] [-removal] [-nochange] [-width] [-period] [-max_skew] [-max_line *lines*] [-list_annotated] [-list_not_annotated] [-constant_arcs]
-setup	Report annotated setup checks.
-hold	Report annotated hold checks.
-recovery	Report annotated recovery checks.
-removal	Report annotated removal checks.
-nochange	Report annotated nochange checks.
-width	Report annotated width checks.
-period	Report annotated period checks.
-max_skew	Report annotated max skew checks.
lines	Maximum number of lines listed by the –list_annotated and –list_not_annotated options.
-list_annotated	List annotated timing arcs.
-list_not_annotated	List unannotated timing arcs.

-constant_arcs	Report separate annotation counts for arcs disabled by logic constants (set_logic_one, set_logic_zero).

The report_annotated_check command reports a summary of SDF timing check annotation. The -list_annotated and –list_not_annotated options can be used to list arcs that are annotated or not annotated.

---

report_annotated_delay	[-cell] [-net] [-from_in_ports] [-to_out_ports] [-max_line *lines*] [-list_annotated] [-list_not_annotated] [-constant_arcs]
-cell	Report annotated cell delays.
-net	Report annotated internal net delays.
-from_in_ports	Report annotated delays from input ports.
-to_out_ports	Report annotated delays to output ports.
lines	Maximum number of lines listed by the –list_annotated and –list_not_annotated options.
-list_annotated	List annotated timing arcs.
-list_not_annotated	List unannotated timing arcs.
-constant_arcs	Report separate annotation counts for arcs disabled by logic constants (set_logic_one, set_logic_zero).

The report_annotated_delay command reports a summary of SDF delay annotation. Without the –from_in_ports and –to_out_ports options annotation of arcs to and from top level ports is not reported. The -list_annotated and –list_not_annotated options can be used to list arcs that are annotated or not annotated.

---

report_cell	[-connections] [-verbose] *instance_path* [> *filename*] [>> *filename*]
-connections	Report the instance pins.

-verbose	With -connections also report all pins connected to each instance pin net.
instance_path	Hierarchical path to the instance.

The report_cell command is an alias for report_instance.

---

report_checks

```
[-from from_list
  |-rise_from from_list
  |-fall_from from_list]
[-through through_list
  |-rise_through through_list
  |-fall_through through_list]
[-to to_list
  |-rise_to to_list
  |-fall_to to_list]
[-unconstrained]
[-path_delay min|min_rise|min_fall
             |max|max_rise|max_fall
             |min_max]
[-group_count path_count]
[-endpoint_count endpoint_path_count]
[-unique_paths_to_endpoint]
[-corner corner_name]
[-slack_max max_slack]
[-slack_min min_slack]
[-sort_by_slack]
[-path_group group_names]
[-format end|full|short|summary
             |full_clock|full_clock_expanded]
[-fields fields]
[-digits digits]
[-no_line_split]
[> filename]
[>> filename]
```

from_list	A list of clocks, instances, ports or pins.
through_list	A list of instances, pins or nets.
to_list	A list of clocks, instances, ports or pins.

-unconstrained	Report unconstrained paths. By default unconstrained paths are not reported.
-path_delay min	Report min path (hold) checks.
-path_delay min_rise	Report min path (hold) checks for rising endpoints.
-path_delay min_fall	Report min path (hold) checks for falling endpoints.
-path_delay max	Report max path (setup) checks.
-path_delay max_rise	Report max path (setup) checks for rising endpoints.
-path_delay max_fall	Report max path (setup) checks for falling endpoints.
-path_delay min_max	Report max and max path (setup and hold) checks.
path_count	The number of paths to report in each path group. The default is 1.
endpoint_path_count	The number of paths to report for each endpoint. The default is 1.
-unique_paths_to_endpoint	When multiple paths to and endpoint are specified with -endpoint_count many of the paths may differ only in the rise/fall edges of the pins in the paths. With this option only the worst path through the set of pis is reported.
corner_name	Only report paths for this process corner. But default the paths across all process corners are reported.
max_slack	Only report paths with less slack than *max_slack*.
min_slack	Only report paths with more slack than *min_slack*.
-sort_by_slack	Sort paths by slack rather than slack grouped by path group.
group_names	List of path group names to report. All path groups are reported if this option is not specified.
-format end	Report path ends in one line with delay, required time and slack.
-format full	Report path start and end points and the path. This is the default path type.

-format full_clock	Report path start and end points, the path, and the source and and target clock paths.
-format full_clock_expanded	Report path start and end points, the path, and the source and and target clock paths. If the clock is generated and propagated, the path from the clock source pin is also reported.
-format short	Report only path start and end points.
-format summary	Report only path ends with delay.
fields	List of capacitance|slew|input_pins|nets|fanout
digits	The number of digits after the decimal point to report. The default value is the variable default_significant_digits.
-no_line_splits	Do not split long lines into multiple lines.

The report_checks command reports paths in the design. See set_false_path for a description of allowed *from_list*, *through_list* and *to_list* objects.

Use the -corner keyword to report timing for one process corner. With no -corner keyword report timing reports the corner with the smallest slack for each timing check.

---

report_check_types

[-violators]
[-verbose]
[-format slack_only|end]
[-max_delay]
[-min_delay]
[-recovery]
[-removal]
[-clock_gating_setup]
[-clock_gating_hold]
[-max_slew]
[-min_slew]
[-min_pulse_width]
[-min_period]
[-digits *digits*]
[-no_split_lines]
[> *filename*]
[>> *filename*]

-violators	Report all violated timing and design rule constraints.
-verbose	Use a verbose output format.
-format	slack_only reports the minumum slack for each check. end reports the endpoint for each check.
-max_delay	Report setup and max delay path delay constraints.
-min_delay	Report hold and min delay path delay constraints.
-recovery	Report asynchronous recovery checks.
-removal	Report asynchronous removal checks.
-clock_gating_setup	Report gated clock enable setup checks.
-clock_gating_hold	Report gated clock hold setup checks.
-max_slew	Report max transition design rule checks.
-max_skew	Report max skew design rule checks.
-min_pulse_width	Report min pulse width design rule checks.
-min_period	Report min period design rule checks.
-min_slew	Report min slew design rule checks.
digits	The number of digits after the decimal point to report. The default is default_significant_digits.
-no_split_lines	

The report_check_types command reports the slack for each type of timing and design rule constraint. The keyword options allow a subset of the constraint types to be reported.

---

report_clock_min_period	[-clocks *clocks*] [-include_port_paths]
clocks	The clocks to report.

-include_port_paths	Include paths from input port and to output ports.

Report the minimum period and maximum frequency for *clocks*. If the -clocks argument is not specified all clock are reported. The minimum period is determined by examining the smallest slack paths between registers the rising edges of the clock or between falling edges of the clock. Paths between different clocks, different clock edges of the same clock, level sensitive latches, or paths constrained by set_multicycle_path, set_max_path are not considered.

report_clock_properties	[*clock_names*]

clock_names	List of clock names to report.

The report_clock_properties command reports the period and rise/fall edge times for each clock that has been defined.

report_clock_skew	[-setup|-hold] [-clock *clocks*] [-digits *digits*]
-setup	Report skew for setup checks.
-hold	Report skew for hold checks.
clocks	The target clocks to report.
digits	The number of digits to report for delays.

Report the clock skew between the source and target clocks for setup or hold timing checks.

report_dcalc	[-from *from_pin*] [-to *to_pin*] [-corner *corner_name*] [-min] [-max] [-digits *digits*] [> *filename*] [>> *filename*]
from_pin	An instance pin.
to_pin	An instance pin.

digits	The number of digits after the decimal point to report. The default is default_significant_digits.

The report_dcalc command shows how the delays between instance pins are calculated. It is useful for debugging problems with delay calculation.

Use the -corner keyword to specify a process corner. The -corner keyword is required if more than one process corner is defined.

---

report_disabled_edges

The report_disabled_edges command reports disabled timing arcs along with the reason they are disabled. Each disabled timing arc is reported as the instance name along with the from and to ports of the arc. The disable reason is shown next. Arcs that are disabled with set_disable_timing are reported with constraint as the reason. Arcs that are disabled by constants are reported with constant as the reason along with the constant instance pin and value. Arcs that are disabled to break combinational feedback loops are reported with loop as the reason.

```
> report_disabled_edges
u1 A B constant B=0
```

---

report_instance	[-connections] [-verbose] *instance_path* [> *filename*] [>> *filename*]
-connections	Report the pins connected to the net.
-verbose	Report the connections of each pin.
instance_path	Hierarchical path to a instance.

---

report_lib_cell	*cell_name* [> *filename*] [>> *filename*]
cell_name	The name of a library cell.

Describe the liberty library cell *cell_name.*

---

report_net [-connections]
[-verbose]
[-significant_digits *digits*]
net_path
[> *filename*]
[>> *filename*]

-connections — Report the net pins.

-verbose — With -connections also report all pins connected to each instance pin net.

net_path — Hierarchical path to a net.

---

report_parasitic_annotation [-report_unannotated]
[> *filename*]
[>> *filename*]

Report SPEF parasitic annotation completeness. When the -report_unannotated option is used unannotated and partially annotated nets and pins are reported.

---

report_pin *pin_path*
[> *filename*]
[>> *filename*]

pin_path — Hierarchical path to a pin.

---

report_power [-instances *instances*]
[-digits *digits*]
[> *filename*]
[>> *filename*]

instances — Report the power for *instances*.

digits — The number of digits after the decimal point to report. The default is default_significant_digits.

The `report_power` command uses static power analysis based on propagated or annotated pin activities in the circuit using Liberty power models. The internal, switching, leakage and total power are reported. Design power is reported separately for combinational, sequential, macro and pad groups. Power values are reported in watts.

Use `-instances` to report power for specific instances.

The `read_power_activities` command can be used to read activities from a file based on simulation. If no simulation activities are available, the `set_power_activity` command should be used to set the activity of input ports or pins in the design. The default input activity and duty for inputs are 0.1 and 0.5 respectively. The activities are propagated from annotated input ports or pins through gates and used in the power calculations.

```
Group                  Internal  Switching    Leakage      Total
                          Power      Power      Power      Power
------------------------------------------------------------------
Sequential             3.29e-06   3.41e-08   2.37e-07   3.56e-06  92.4%
Combinational          1.86e-07   3.31e-08   7.51e-08   2.94e-07   7.6%
Macro                  0.00e+00   0.00e+00   0.00e+00   0.00e+00   0.0%
Pad                    0.00e+00   0.00e+00   0.00e+00   0.00e+00   0.0%
------------------------------------------------------------------
Total                  3.48e-06   6.72e-08   3.12e-07   3.86e-06 100.0%
                          90.2%       1.7%       8.1%
```

---

report_pulse_width_checks [-verbose]
[-digits *digits*]
[-no_line_splits]
[*pins*]
[> *filename*]
[>> *filename*]

`-verbose` Use a verbose output format.

digits The number of digits after the decimal point to report. The default is default_significant_digits.

`-no_line_splits`

pins List of pins or ports to report.

The `report_pulse_width_checks` command reports min pulse width checks for pins in the clock network. If *pins* is not specified all clock network pins are reported.

---

report_units

Report the command units.

```
report_units
 time 1ns
 capacitance 1pF
```

```
resistance 1kohm
voltage 1v
current 1A
power 1pW
distance 1um
```

---

report_worst_slack	[-min] [-max] [-digits *digits*]
-max	Report the worst max/setup slack.
-min	Report the worst min/hold slack.
digits	The number of digits after the decimal point to report. The default is default_significant_digits.

---

set_assigned_check	-setup|-hold|-recovery|-removal [-rise] [-fall] [-corner *corner_name*] [-min] [-max] [-from *from_pins*] [-to *to_pins*] [-clock rise|fall] [-cond sdf_cond] [-worst] *margin*
-setup|-hold|-recovery|-removal	The timing check type to annotate.
-rise	Annotate the rising delays.
-fall	Annotate the falling delays.
corner_name	The name of a process corner.
-min	Annotate the minimum value of the process corner.
-max	Annotate the maximum value of the process corner.
from_pins	A list of pins for the clock.

to_pins	A list of pins for the data.
-clock	The clock pin transition.
-worst	Ignored.
margin	The timing check value.

The set_assigned_check command is used to annotate the timing checks between two pins on an instance. The annotated delay overrides the calculated delay. This command is a interactive way to back-annotate delays like an SDF file.

Use the -corner keyword to specify a process corner. The -corner keyword is required if more than one process corner is defined.

---

set_assigned_delay	-cell|-net [-rise] [-fall] [-corner *corner_name*] [-min] [-max] [-from *from_pins*] [-to *to_pins*] *delay*
-cell	Annotate the delays between two pins on an instance.
-net	Annotate the delays between two pins on a net.
-rise	Annotate the rising delays.
-fall	Annotate the falling delays.
-min	Annotate the minimum delays.
-max	Annotate the maximum delays.
from_pins	A list of pins.
to_pins	A list of pins.
delay	The delay between *from_pins* and *to_pins*.

The set_assigned_delay command is used to annotate the delays between two pins on an instance or net. The annotated delay overrides the calculated delay. This command is a interactive way to back-annotate delays like an SDF file.

Use the -corner keyword to specify a process corner. The -corner keyword is required if more than one process corner is defined.

set_assigned_transition	[-rise] [-fall] [-corner *corner_name*] [-min] [-max] *slew* *pin_list*
-rise	Annotate the rising transition.
-fall	Annotate the falling transition.
-min	Annotate the minimum transition time.
-max	Annotate the maximum transition time.
slew	The pin transition time.
pin_list	A list of pins.

The set_assigned_transition command is used to annotate the transition time (slew) of a pin. The annotated transition time overrides the calculated transition time.

Use the -corner keyword to specify a process corner. The -corner keyword is required if more than one process corner is defined.

set_case_analysis	0|1|zero|one|rise|rising|fall|falling *port_or_pin_list*
port_or_pin_list	A list of ports or pins.

The set_case_analysis command sets the signal on a port or pin to a constant logic value. No paths are propagated from constant pins. Constant values set with the set_case_analysis command are propagated through downstream gates.

set_clock_gating_check	[-setup *setup_time*] [-hold *hold_time*] [-rise] [-fall] [-high] [-low] [*objects*]
setup_time	Clock enable setup margin.
hold_time	Clock enable hold margin.
-rise	The setup/hold margin is for the rising edge of the clock enable.
-fall	The setup/hold margin is for the falling edge of the clock enable.
-high	The gating clock is active high (pin and instance objects only).
-low	The gating clock is active low (pin and instance objects only).
objects	A list of clocks, instances, pins or ports.

The set_clock_gating_check command is used to add setup or hold timing checks for data signals used to gate clocks.

If no objects are specified the setup/hold margin is global and applies to all clock gating circuits in the design. If neither of the -rise and -fall options are used the setup/hold margin applies to the rising and falling edges of the clock gating signal.

Normally the library cell function is used to determine the active state of the clock. The clock is active high for AND/NAND functions and active low for OR/NOR functions. The -high and -low options are used to specify the active state of the clock for other cells, such as a MUX.

If multiple set_clock_gating_check commands apply to a clock gating instance he priority of the commands is shown below (highest to lowest priority).

```
clock enable pin
instance
clock pin
clock
global
```

set_clock_groups	[-name *name*] [-logically_exclusive] [-physically_exclusive] [-asynchronous] [-allow_paths] -group *clocks*
name	The clock group name.
-logically_exclusive	The clocks in different groups do not interact logically but can be physically present on the same chip. Paths between clock groups are considered for noise analysis.
-physically_exclusive	The clocks in different groups cannot be present at the same time on a chip. Paths between clock groups are *not* considered for noise analysis.
-asynchronous	The clock groups are asynchronous. Paths between clock groups are considered for noise analysis.
-allow_paths	
clocks	A list of clocks in the group.

The set_clock_groups command is used to deifine groups of clocks that interact with each other. Clocks in different groups do not interact and paths between them are not reported. Use a –group argument for each clock group.

set_clock_latency	[-source] [-clock *clock*] [-rise] [-fall] [-min] [-max] *delay* *objects*
-source	The latency is at the clock source.
clock	If multiple clocks are defined at a pin this use this option to specify the latency for a specific clock.
-rise	The latency is for the rising edge of the clock.

`-fall`	The latency is for the falling edge of the clock.
`-min`	*delay* is the minimum latency.
`-max`	*delay* is the maximum latency.
delay	Clock source or insertion delay.
objects	A list of clocks, pins or ports.

The `set_clock_latency` command describes expected delays of the clock tree when analyzing a design using ideal clocks. Use the `-source` option to specify latency at the clock source, also known as insertion delay. Source latency is delay in the clock tree that is external to the design or a clock tree internal to an instance that implements a complex logic function.

---

set_clock_transition	[-rise] [-fall] [-min] [-max] *transition* *clocks*
-rise	The transition time is for the rising edge of the clock.
-fall	The transition time is for the falling edge of the clock.
-min	*transition* is the minimum transition time.
-max	*transition* is the maximum transition time.
transition	Clock transition time (slew).
clocks	A list of clocks.

The `set_clock_transition` command describes expected transition times of the clock tree when analzying a design using ideal clocks.

set_clock_uncertainty	[-from|-rise_from|-fall_from *from_clock*] [-to|-rise_to|-fall_to *to_clock*] [-rise] [-fall] [-setup] [-hold] *uncertainty* [*objects*]
from_clock	Inter-clock uncertainty source clock.
to_clock	Inter-clock uncertainty target clock.
-rise	Inter-clock target clock rise edge, alternative to -rise_to.Inter-clock target clock rise edge, alternative to -rise_to.
-fall	Inter-clock target clock rise edge, alternative to -fall_to.
-setup	*uncertainty* is for setup checks.
-hold	*uncertainty* is for hold checks.
uncertainty	Clock uncertainty.
objects	A list of clocks, ports or pins.

The set_clock_uncertainty command specifies the uncertainty or jitter in a clock. The uncertainty for a clock can be specified on its source pin or port, or the clock itself.

```
set_clock_uncertainty .1 [get_clock clk1]
```

Inter-clock uncertainty between the source and target clocks of timing checks is specified with the -from|-rise_from|-fall_from and -to|-rise_to|-fall_to arguments .

```
set_clock_uncertainty -from [get_clock clk1] -to [get_clocks clk2] .1
```

The following commands are equivalent.

```
set_clock_uncertainty -from [get_clock clk1] -rise_to [get_clocks clk2] .1
set_clock_uncertainty -from [get_clock clk1] -to [get_clocks clk2] -rise .1
```

set_cmd_units	[-capacitance *cap_unit*] [-resistance *res_unit*] [-time *time_unit*] [-voltage *voltage_unit*] [-current *current_unit*] [-power *power_unit*] [-distance *distance_unit*]
cap_unit	The capacitance scale factor followed by 'f'.
res_unit	The resistance scale factor followed by 'ohm'.
time_unit	The time scale factor followed by 's'.
voltage_unit	The voltage scale factor followed by 'v'.
current_unit	The current scale factor followed by 'A'.
power_unit	The power scale factor followed by 'w'.
distance_unit	The distance scale factor followed by 'm'.

The `set_cmd_units` command is used to change the units used by the STA command interpreter when parsing commands and reporting results. The default units are the units specified in the first Liberty library file that is read.

Units are specified as a scale factor followed by a unit name. The scale factors are as follows.

```
M 1E+6
k 1E+3
m 1E-3
u 1E-6
n 1E-9
p 1E-12
f 1E-15
```

An example of the `set_units` command is shown below.

```
set_cmd_units -time ns -capacitance pF -current mA -voltage V
              -resistance kOhm -distance um
```

set_data_check

[-from *from_object*]
[-rise_from *from_object*]
[-fall_from *from_object*]
[-to *to_object*]
[-rise_to *to_object*]
[-fall_to *to_object*]
[-setup]
[-hold]
[-clock *clock*]
margin

from_object A pin used as the timing check reference.

to_object A pin that the setup/hold check is applied to.

-setup Add a setup timing check.

-hold Add a hold timing check.

clock The setup/hold check clock.

margin The setup or hold time margin.

The set_data_check command is used to add a setup or hold timing check between two pins.

---

set_disable_inferred_clock_gating *objects*

objects A list of clock gating instances, clock gating pins, or clock enable pins.

The set_disable_inferred_clock_gating command disables clock gating checks on a clock gating instance, clock gating pin, or clock gating enable pin.

---

set_disable_timing

[-from *from_port*]
[-to *to_port*]
objects

from_port

to_port

objects	A list of instances, ports, pins, cells or [library/]cell/port.

The set_disable_timing command is used to disable paths though pins in the design. There are many different forms of the command depending on the objects specified in *objects*.

All timing paths though an instance are disabled when *objects* contains an instance. Timing checks in the instance are *not* disabled.

```
set_disable_timing u2
```

The -from and -to options can be used to restrict the disabled path to those from, to or between specific pins on the instance.

```
set_disable_timing -from A u2
set_disable_timing -to Z u2
set_disable_timing -from A -to Z u2
```

A list of top level ports or instance pins can also be disabled.

```
set_disable_timing u2/Z
set_disable_timing in1
```

Timing paths though all instances of a library cell in the design can be disabled by naming the cell using a hierarchy separator between the library and cell name. Paths from or to a cell port can be disabled with the -from and -to options or a port name after library and cell names.

```
set_disable_timing liberty1/snl_bufx2
set_disable_timing -from A liberty1/snl_bufx
set_disable_timing -to Z liberty1/snl_bufx
set_disable_timing liberty1/snl_bufx2/A
```

---

set_drive	[-rise] [-fall] [-max] [-min] *resistance* *port_list*
-rise	This is the drive resistance of the rising edge of the input.
-fall	This is the drive resistance of the falling edge of the input.
-max	This is the drive resistance for maximum path delays.
-min	This is the drive resistance for minimum path delays.

resistance	The external drive resistance.
port_list	A list of ports.

The set_drive command describes the resistance of an input port external driver.

---

set_driving_cell	[-lib_cell *cell_name*] [-library *library*] [-rise] [-fall] [-min] [-max] [-pin *pin*] [-from_pin *from_pin*] [-input_transition_rise *trans_rise*] [-input_transition_fall *trans_fall*] *port_list*
cell_name	The cell of driver.
library	The library of the driving cell.
-rise	This is the driving cell for the rising edge of the input.
-fall	This is the driving cell for the falling edge of the input.
-max	This is the driving cell for maximum path delays.
-min	This is the driving cell for minimum path delays.
pin	The output port of the driving cell.
from_pin	Use paths through the driving cell from this pin to the output pin.
trans_rise	The transition time for a rising input at *from_pin*.
trans_fall	The transition time for a falling input at *from_pin*.
port_list	A list of ports.

The set_driving_cell command describes an input port external driver.

set_false_path	[-setup] [-hold] [-rise] [-fall] [-from *from_list*] [-rise_from *from_list*] [-fall_from *from_list*] [-through *through_list*] [-rise_through *through_list*] [-fall_through *through_list*] [-to *to_list*] [-rise_to *to_list*] [-fall_to *to_list*] [-reset_path]
-setup	Only apply to setup checks.
-hold	Only apply to hold checks.
-rise	Only apply to rising path edges.
-fall	Only apply to falling path edges.
-reset_path	Remove any matching set_false_path, set_multicycle_path, set_max_delay, set_min_delay exceptions first.
from_list	A list of clocks, instances, ports or pins.
through_list	A list of instances, pins or nets.
to_list	A list of clocks, instances, ports or pins.

The set_false_path command disables timing along a path from, through and to a group of design objects.

Objects in *from_list* can be clocks, register/latch instances, or register/latch clock pins. The -rise_from and -fall_from keywords restrict the false paths to a specific clock edge.

Objects in *through_list* can be nets, instances, instance pins, or hierarchical pins,. The -rise_through and -fall_through keywords restrict the false paths to a specific path edge that traverses through the object.

Objects in *to_list* can be clocks, register/latch instances, or register/latch clock pins. The -rise_to and -fall_to keywords restrict the false paths to a specific transition at the path end.

set_fanout_load *fanout*
port_list

This command is ignored.

set_hierarchy_separator *separator*

separator Character used to separate hierarchical names.

Set the character used to separate names in a hierarchical instance, net or pin name. This separator is used by the command interpreter to read arguments and print results. The default separator is '/'.

set_ideal_latency [-rise] [-fall] [-min] [-max] *delay objects*

The set_ideal_latency command is parsed but ignored.

set_ideal_network [-no_propagation] *objects*

The set_ideal_network command is parsed but ignored.

set_ideal_transition [-rise] [-fall] [-min] [-max] *transition_time objects*

The set_ideal_transition command is parsed but ignored.

set_input_delay
[-rise]
[-fall]
[-max]
[-min]
[-clock *clock*]
[-clock_fall]
[-reference_pin *ref_pin*]
[-source_latency_included]
[-network_latency_included]
[-add_delay]
delay
port_pin_list

-rise This is the arrival time for the rising edge of the input.

-fall	This is the arrival time for the falling edge of the input.
-max	This is the minimum arrival time.
-min	This is the maximum arrival time.
clock	The arrival time is from this clock.
-clock_fall	The arrival time is from the falling edge of *clock*
ref_pin	The arrival time is with respect to the clock that arrives at *ref_pin*.
-source_latency_included	If -source_latency_included is not specified the clock source latency (insertion delay) is added to the delay value.
-network_latency_included	If -network_latency_included is not specified and the clock is ideal the clock latency is added to the delay value.
-add_delay	Add this arrival to any existing arrivals on *port_pin_list*.
delay	The arrival time after *clock*.
pin_port_list	A list of pins or ports.

The set_input_delay command is used to specify the arrival time of an input signal. Unless the -add_delay option is specified, any existing arrival time is replaced.

The –reference_pin option is used to specify an arrival time with respect to the arrival on a pin in the clock network. For propagated clocks, the input arrival time is relative to the clock arrival time at the reference pin (the clock source latency and network latency from the clock source to the reference pin). For ideal clocks, input arrival time is relative to the reference pin clock source latency. With the -clock_fall flag the arrival time is relative to the falling transition at the reference pin. If no clocks arrive at the reference pin the set_input_delay command is ignored. If no -clock is specified the arrival time is with respect to all clocks that arrive at the reference pin. The -source_latency_included and -network_latency_included options cannot be used with -reference_pin.

Paths from inputs that do not have an arrival time defined by set_input_delay are not reported. Set the sta_input_port_default_clock variable to 1 to report paths from inputs without a set_input_delay.

set_input_transition	[-rise] [-fall] [-max] [-min] *transition* *port_list*
-rise	This is the transition time for the rising edge of the input.
-fall	This is the transition time for the falling edge of the input.
-max	This is the minimum transition time.
-min	This is the maximum transition time.
transition	The transition time (slew).
port_list	A list of ports.

The set_input_transition command is used to specify the transition time (slew) of an input signal.

set_level_shifter_strategy	[-rule *rule_type*]

This command is parsed and ignored by timing analysis.

set_level_shifter_threshold	[-voltage *voltage*]

This command is parsed and ignored by timing analysis.

set_load	[-rise] [-fall] [-max] [-min] [-subtract_pin_load] [-pin_load] [-wire_load] *capacitance* *objects*
-rise	The capacitance is for rising edge delays.

`-fall`	The capacitance is for falling edge delays.
`-max`	The capacitance is for maximum path delays.
`-min`	The capacitance is for minimum path delays.
`-subtract_pin_load`	Subtract the capacitance of all instance pins connected to the net from wire *capacitance*.
`-pin_load`	*capcitance* is external instance pin capacitance (ports only).
`-wire_load`	*capcitance* is external wire capacitance (ports only).
capacitance	The capacitance, in library capacitance units.
objects	A list of nets or ports.

The `set_load` command annotates capacitance on a net or port.

Ports can have external wire or pin capacitance that is annotated separately with the `-pin_load` and `-wire_load` options. Without the `-pin_load` and `-wire_load` options pin capacitance is annotated. External capacitances are used by delay calculator to find output driver delays and transition times.

Net wire capacitance can also be annotated with the `set_load` command. If the *-subtract_pin_load* option is specified the capacitance of all instance pins connected to the net is subtracted from *capacitance*. `set_load` command annotates capacitance has precidence over RC SPEF parasitics.

---

set_logic_dc	*port_list*
port_pin_list	List of ports or pins.

Set a port or pin to a constant unknown logic value. No paths are propagated from constant pins.

---

set_logic_one	*port_list*
port_pin_list	List of ports or pins.

Set a port or pin to a constant logic one value. No paths are propagated from constant pins. Constant values set with the `set_logic_one` command are **not** propagated through downstream gates.

---

set_logic_zero	*port_list*
port_pin_list	List of ports or pins.

Set a port or pin to a constant logic zero value. No paths are propagated from constant pins. Constant values set with the `set_logic_zero` command are **not** propagated through downstream gates.

---

set_max_area *area*

area

The `set_max_area` command is ignored during timing but is included in SDC files that are written.

---

set_max_capacitance *capacitance*
objects

capacitance

objects List of ports or cells.

The `set_max_capacitance` command is ignored during timing but is included in SDC files that are written.

---

set_max_delay [-rise]
[-fall]
[-from *from_list*]
[-rise_from *from_list*]
[-fall_from *from_list*]
[-through *through_list*]
[-rise_through *through_list*]
[-fall_through *through_list*]
[-to *to_list*]
[-rise_to *to_list*]
[-fall_to *to_list*]
[-reset_path]
[-ignore_clock_latency]
delay

-rise Only constrain paths to rising edges.

-fall Only constrain paths to falling edges.

from_list A list of clocks, instances, ports or pins.

through_list A list of instances, pins or nets.

to_list A list of clocks, instances, ports or pins.

-ignore_clock_latency	Ignore clock latency at the source and target registers.
-reset_path	Remove any matching set_false_path, set_multicycle_path, set_max_delay, set_min_delay exceptions first.
delay	The maximum delay.

The set_max_delay command constrains the maximum delay through combinational logic paths. See set_false_path for a description of allowed *from_list*, *through_list* and *to_list* objects. If the *to_list* ends at a timing check the setup/hold time is included in the path delay.

When the -ignore_clock_latency option is used clock latency at the source and destination of the path delay is ignored. The constraint is reported in the default path group (**default**) rather than the clock path group when the path ends at a timing check.

---

set_max_dynamic_power *power* [*unit*]

The set_max_dynamic_power command is ignored.

---

set_max_fanout *fanout*
objects

fanout

objects	List of ports or cells.

The set_max_fanout command is ignored during timing but is included in SDC files that are written.

---

set_max_leakage_power *power* [*unit*]

The set_max_leakage_power command is ignored.

---

set_max_time_borrow *delay*
objects

delay	The maximum time the latches can borrow.
objects	List of clocks, instances or pins.

The set_max_time_borrow command specifies the maximum amount of time that latches can borrow. Time borrowing is the time that a data input to a transparent latch arrives after the latch opens.

---

set_max_transition [-data_path]
[-clock_path]
[-rise]
[-fall]
transition
objects

transition The maximum transition time (slew).

objects List of clocks, ports or designs.

The set_max_transition command is specifies the maximum transition time (slew) design rule checked by the report_constraint –max_transition command.

If specified for a design, the default maximum transition is set for the design.

If specified for a clock, the maximum transition is applied to all pins in the clock domain. The –clock_path option restricts the maximum transition to clocks in clock paths. The -data_path option restricts the maximum transition to clocks data paths. The –clock_path, -data_path, -rise and –fall options only apply to clock objects.

---

set_min_capacitance *capacitance*
objects

capacitance

objects List of ports or cells.

The set_min_capacitance command is ignored during timing but is included in SDC files that are written.

---

set_min_delay [-rise]
[-fall]
[-from *from_list*]
[-rise_from *from_list*]
[-fall_from *from_list*]
[-through *through_list*]
[-rise_through *through_list*]
[-fall_through *through_list*]
[-to *to_list*]
[-rise_to *to_list*]
[-fall_to *to_list*]
[-ignore_clock_latency]
[-reset_path]
delay

-rise	Only constrain paths to rising edges.
-fall	Only constrain paths to falling edges.
from_list	A list of clocks, instances, ports or pins.
through_list	A list of instances, pins or nets.
to_list	A list of clocks, instances, ports or pins.
-ignore_clock_latency	Ignore clock latency at the source and target registers.
-reset_path	Remove any matching set_false_path, set_multicycle_path, set_max_delay, set_min_delay exceptions first.
delay	The minimum delay.

The set_min_delay command constrains the minimum delay through combinational logic. See set_false_path for a description of allowed *from_list*, *through_list* and *to_list* objects. If the *to_list* ends at a timing check the setup/hold time is included in the path delay.

When the -ignore_clock_latency option is used clock latency at the source and destination of the path delay is ignored. The constraint is reported in the default path group (**default**) rather than the clock path group when the path ends at a timing check.

---

set_min_pulse_width	[-high] [-low] *min_width* *objects*
-high	Set the minimum high pulse width.
-low	Set the minimum low pulse width.
min_width	
objects	List of pins, instances or clocks.

If -low and -high are not specified the minimum width applies to both high and low pulses.

set_multicycle_path	[-setup] [-hold] [-rise] [-fall] [-start] [-end] [-from *from_list*] [-rise_from *from_list*] [-fall_from *from_list*] [-through *through_list*] [-rise_through *through_list*] [-fall_through *through_list*] [-to *to_list*] [-rise_to *to_list*] [-fall_to *to_list*] [-reset_path] *path_multiplier*
-setup	Only apply to setup checks.
-hold	Only apply to hold checks.
-rise	Only apply to rising path edges.
-fall	Only apply to falling path edges.
-start	Multiply the source clock period by *period_multiplier*.
-end	Multiply the target clock period by *period_multiplier*.
-reset_path	Remove any matching set_false_path, set_multicycle_path, set_max_delay, set_min_delay exceptions first.
from_list	A list of clocks, instances, ports or pins.
through_list	A list of instances, pins or nets.
to_list	A list of clocks, instances, ports or pins.
path_multiplier	The number of clock periods to add to the path required time.

Normally the path between two registers or latches is assumed to take one clock cycle. The set_multicycle_path command overrides this assumption and allows multiple clock cycles for a timing check. See set_false_path for a description of allowed *from_list*, *through_list* and *to_list* objects.

set_operating_conditions	[-analysis_type single|bc_wc|on_chip_variation] [-library *lib*] [*condition*] [-min *min_condition*] [-max *max_condition*] [-min_library *min_lib*] [-max_library *max_lib*]
single	Use one operating condition for min and max paths.
bc_wc	Best case, worst case analysis. Setup checks use *max_condition* for clock and data paths. Hold checks use the *min_condition* for clock and data paths.
on_chip_variation	The min and max operating conditions represent variations on the chip that can occur simultaineously. Setup checks use *max_condition* for data paths and *min_condition* for clock paths. Hold checks use *min_condition* for data paths and *max_condition* for clock paths. This is the default analysis type.
lib	The name of the library that contains *condition*.
condition	The operating condition for analysis type single.
min_condition	The operating condition to use for min paths and hold checks.
max_condition	The operating condition to use for max paths and setup checks.
min_lib	The name of the library that contains *min_condition*.
max_lib	The name of the library that contains *max_condition*.

The set_operating_conditions command is used to specify the type of analysis performed and the operating conditions used to derate library data.

set_output_delay	[-rise] [-fall] [-max] [-min] [-clock *clock*] [-clock_fall] [-reference_pin *ref_pin*] [-source_latency_included] [-network_latency_included] [-add_delay] *delay* *port_pin_list*
-rise	This is the arrival time for the rising edge of the input.
-fall	This is the arrival time for the falling edge of the input.
-max	This is the minimum arrival time.
-min	This is the maximum arrival time.
clock	The departure time is from this clock.
-clock_fall	The departure time is from the falling edge of *clock*.
ref_pin	The departure time is with respect to the clock that arrives at *ref_pin*.
-add_delay	Add this departure to any existing arrivals on *port_pin_list*.
delay	The departure time after *clock*.
pin_port_list	A list of pins or ports.

The set_output_delay command is used to specify the departure time of an output signal. Unless the -add_delay option is specified any existing departure time is replaced.

The –reference_pin option is used to specify a departure time with respect to the arrival on a pin in the clock network. For propagated clocks, the output departure time is relative to the clock arrival time at the reference pin (the clock source latency and network latency from the clock source to the reference pin). For ideal clocks, output departure time is relative to the reference pin clock source latency. With the -clock_fall flag the departure time is relative to the falling transition at the reference pin. If no clocks arrive at the reference pin the set_output_delay command is ignored. If no -clock is specified the departure time is with respect to all

clocks that arrive at the reference pin. The -source_latency_included and -network_latency_included options cannot be used with -reference_pin.

set_port_fanout_number	[-min] [-max] *fanout* *port_list*
-min	The fanout for minimum path delay calculation.
-max	The fanout for maximum path delay calculation.
fanout	The external fanout of the ports.
port_list	A list of ports.

set_power_activity	[-global] [-input] [-input_ports *ports*] [-pins *pins*] [-activity *activity*] [-duty *duty*]
-global	Set the activity/duty for all non-clock pins.
-input	Set the default input port activity/duty.
input_ports	Set the input port activity/duty.
pins	Set the pin activity/duty.
activity	The activity, or number of transitions per clock cycle.
duty	The duty, or probability the signal is high. Defaults to 0.5.

The set_power_activity command is used to set the activity and duty used for power analysis globally or for input ports or pins in the design.

The default input activity and duty for inputs are 0.1 and 0.5 respectively, which is equivalent to the following command:

```
set_power_activity -input -activity 0.1 -duty 0.5
```

---

set_propagated_clock *objects*

objects A list of clocks, ports or pins.

The set_propagated_clock command changes a clock tree from an ideal network that has no delay one that uses calculated or back-annotated gate and interconnect delays. When *objects* is a port or pin, clock delays downstream of the object are used.

---

set_pvt *instances*
[-min]
[-max]
[-process *process*]
[-voltage *voltage*]
[-temperature *temperature*]

instances A list instances.

-min Only set the PVT values for max delay paths.

-max Only set the PVT values for min delay paths.

process A process value (float).

voltage A voltage value (float).

temperature A temperature value (float).

The set_pvt command sets the process, voltage and temperature values used during delay calculation for a specific instance in the design.

---

set_sense [-type clock|data]
[-positive]
[-negative]
[-pulse *pulse_type*]
[-stop_propagation]
[-clock *clocks*]
pins

-positive The clock sense is positive unite.

-negative	The clock sense is negative unite.
pulse_type	rise_triggered_high_pulse rise_triggered_low_pulse fall_triggered_high_pulse fall_triggered_low_pulse
-stop_propagation	Stop propagating *clocks* clocks at *pins*.
clocks	A list of clocks to apply the sense
pins	A list of pins.

The set_sense command is used to modify the propagation of a clock signal. Note that -type data is not supported. The clock sense is set with the –positive and –negative flags. Use the –stop_propagation flag to stop the clock from propagating beyond a pin. The –positive, -negative, -stop_propagation, and –pulse options are mutually exclusive. If the –clock option is not used the command applies to all clocks that traverse *pins*. The –pulse option is currently not supported.

set_timing_derate	[-rise] [-fall] [-early] [-late] [-clock] [-data] [-net_delay] [-cell_delay] [-cell_check] *derate* [*objects*]
-early	Derate early (min) paths.
-late	Derate late (max) paths.
-clock	Derate paths in the clock network.
-data	Derate data paths.
-net_delay	Derate net (interconnect) delays.
-cell_delay	Derate cell delays.
-cell_check	Derate cell timing check margins.

derate	The derating factor to apply to delays.
objects	A list of instances, library cells, or nets.

The set_timing_derate command is used to derate delay calculation results used by the STA. If the -early and -late flags are omitted the both min and max paths are derated. If the -clock and -data flags are not used the derating both clock and data paths are derated.

Use the unset_timing_derate command to remove all derating factors.

---

set_resistance [-max]
[-min]
resistance
net_list

-min	The resistance for minimum path delay calculation.
-max	The resistance for maximum path delay calculation.
resistance	The net resistance.
net_list	A list of nets.

---

set_units [-capacitance *cap_unit*]
[-resistance *res_unit*]
[-time *time_unit*]
[-voltage *voltage_unit*]
[-current *current_unit*]
[-power *power_unit*]
[-distance *distance_unit*]

cap_unit	The capacitance scale factor followed by 'f'.
res_unit	The resistance scale factor followed by 'ohm'.
time_unit	The time scale factor followed by 's'.
voltage_unit	The voltage scale factor followed by 'v'.
current_unit	The current scale factor followed by 'A'.

power_unit The power scale factor followed by 'w'.

The set_units command is used to **check** the units used by the STA command interpreter when parsing commands and reporting results. If the current units differ from the set_unit value a warning is printed. Use the set_cmd_units command to change the command units.

Units are specified as a scale factor followed by a unit name. The scale factors are as follows.

```
M 1E+6
k 1E+3
m 1E-3
u 1E-6
n 1E-9
p 1E-12
f 1E-15
```

An example of the set_units command is shown below.

```
set_units -time ns -capacitance pF -current mA -voltage V -resistance kOhm
```

---

set_wire_load_min_block_size *size*

The set_wire_load_min_block_size command is not supported.

---

set_wire_load_mode top|enclosed|segmented

top

enclosed

segmented

The set_wire_load_mode command is ignored during timing but is included in SDC files that are written.

---

set_wire_load_model -name *model_name*
[-library *library*]
[-max]
[-min]
[*objects*]

model_name The name of a wire load model.

library Library to look for *model_name*.

-max	The wire load model is for maximum path delays.
-min	The wire load model is for minimum path delays.
objects	Not supported.

---

set_wire_load_selection_group [-library *library*]
[-max]
[-min]
group_name
[*objects*]

library	Library to look for *group_name*.
-max	The wire load selection is for maximum path delays.
-min	The wire load selection is for minimum path delays.
group_name	A wire load selection group name.
objects	Not supported.

The set_wire_load_selection_group command is parsed but not supported.

---

source [-echo]
[-verbose]
filename
> *filename*]
>> *filename*]

-echo	Print each command before evaluating it.
-verbose	Print each command before evaluating it as well as the result it returns.
filename	The name of the file containing commands to read.

Read STA/SDC/Tcl commands from *filename*.

The source command stops and reports any errors encountered while reading a file unless sta_continue_on_error is 1.

unset_case_analysis	*port_or_pin_list*
port_or_pin_list	A list of ports or pins.

The unset_case_analysis command removes the constant values defined by the set_case_analysis command.

unset_clock_latency	[-source] *objects*
-source	Specifies source clock latency (clock insertion delay).
objects	A list of clocks, pins or ports.

The unset_clock_latency command removes the clock latency set with the set_clock_latency command.

unset_clock_transition	*clocks*
clocks	A list of clocks.

The unset_clock_transition command removes the clock transition set with the set_clock_transition command.

unset_clock_uncertainty	[-from|-rise_from|-fall_from *from_clock*] [-to|-rise_to|-fall_to *to_clock*] [-rise] [-fall] [-setup] [-hold] [*objects*]
from_clock	
to_clock	
-rise	The uncertainty is for the rising edge of the clock.
-fall	The uncertainty is for the falling edge of the clock.
-setup	*uncertainty* is the setup check uncertainty.
-hold	*uncertainty* is the hold uncertainty.

uncertainty	Clock uncertainty.
objects	A list of clocks, ports or pins.

The unset_clock_uncertainty command removes clock uncertainty defined with the set_clock_uncertainty command.

---

unset_data_check	[-from *from_object*] [-rise_from *from_object*] [-fall_from *from_object*] [-to *to_object*] [-rise_to *to_object*] [-fall_to *to_object*] [-setup] [-hold] [-clock *clock*]
from_object	A pin used as the timing check reference.
to_object	A pin that the setup/hold check is applied to.
-setup	Add a setup timing check.
-hold	Add a hold timing check.
clock	The setup/hold check clock.

The unset_clock_transition command removes a setup or hold check defined by the set_data_check command.

---

unset_disable_inferred_clock_g*objects*
ating

objects	A list of clock gating instances, clock gating pins, or clock enable pins.

The unset_disable_inferred_clock_gating command removes a previous set_disable_inferred_clock_gating command.

---

unset_disable_timing	[-from *from_port*] [-to *to_port*] *objects*

from_port

to_port

objects — A list of instances, ports, pins, cells or [library/]cell/port.

The unset_disable_timing command is used to remove the effect of previous set_disable_timing commands.

---

unset_input_delay [-rise] [-fall] [-max] [-min] [-clock *clock*] [-clock_fall] *port_pin_list*

-rise — This is the arrival time for the rising edge of the input.

-fall — This is the arrival time for the falling edge of the input.

-max — This is the minimum arrival time.

-min — This is the maximum arrival time.

clock — The arrival time is from this clock.

-clock_fall — The arrival time is from the falling edge of *clock*

pin_port_list — A list of pins or ports.

The unset_input_delay command removes a previously defined set_input_delay.

---

unset_output_delay [-rise] [-fall] [-max] [-min] [-clock *clock*] [-clock_fall] *port_pin_list*

-rise	This is the arrival time for the rising edge of the input.
-fall	This is the arrival time for the falling edge of the input.
-max	This is the minimum arrival time.
-min	This is the maximum arrival time.
clock	The arrival time is from this clock.
-clock_fall	The arrival time is from the falling edge of *clock*
pin_port_list	A list of pins or ports.

The unset_output_delay command a previously defined set_output_delay.

---

unset_path_exceptions	[-setup] [-hold] [-rise] [-fall] [-from *from_list*] [-rise_from *from_list*] [-fall_from *from_list*] [-through *through_list*] [-rise_through *through_list*] [-fall_through *through_list*] [-to *to_list*] [-rise_to *to_list*] [-fall_to *to_list*]
-setup	Only apply to setup checks.
-hold	Only apply to hold checks.
-rise	Only apply to rising path edges.
-fall	Only apply to falling path edges.
from_list	A list of clocks, instances, ports or pins.

through_list	A list of instances, pins or nets.
to_list	A list of clocks, instances, ports or pins.

The unset_path_exceptions command removes any matching set_false_path, set_multicycle_path, set_max_delay, and set_min_delay exceptions.

---

unset_propagated_clock *objects*

objects	A list of clocks, ports or pins.

Remove a previous set_propagated_clock command.

---

unset_timing_derate

Remove all derating factors set with the set_timing_derate command.

---

user_run_time

Returns the total user cpu run time in seconds as a float.

---

with_output_to_variable *var* { *commands* }

var	The name of a variable to save the output of *commands* to.
commands	TCL commands that the output will be redirected from.

The with_output_to_variable command redirects the output of TCL commands to a variable.

---

write_path_spice
-path_args *path_args*
-spice_directory *spice_directory*
-lib_subckt_file *lib_subckts_file*
-model_file *model_file*
-power *power*
-ground *ground*

path_args	-from|-through|-to arguments as in report_checks.
spice_directory	Spice output directory.

lib_subckts_file	Cell transistor level subckts.
model_file	Transistor model definitions .included by *spice_file*.
power	Voltage supply name in voltage_map of the default liberty library.
ground	Ground supply name in voltage_map of the default liberty library.

The write_path_spice command writes a spice netlist for timing paths. Use *path_args* to specify -from/-through/-to as arguments to the find_timing_paths command. For each path, a spice netlist and the subckts referenced by the path are written in *spice_directory*. The spice netlist is written in path_<id>.sp and subckt file is path_<id>.subckt.

The spice netlists used by the path are written to *subckt_file*, which spice_file .includes. The device models used by the spice subckt netlists in *model_file* are also .included in spice_file. Power and ground names are specified with the -power and -ground arguments. The spice netlist includes a piecewise linear voltage source at the input and .measure statement for each gate delay and pin slew.

Example command:

```
write_path_spice -path_args {-from "in0" -to "out1" -unconstrained} \
  -spice_directory $result_dir \
  -lib_subckt_file "write_spice1.subckt" \
  -model_file "write_spice1.models" \
  -power VDD -ground VSS
```

---

write_sdc	[-digits *digits*] [-gzip] [-no_timestamp] *filename*
digits	The number of digits after the decimal point to report. The default is 4.
-gzip	Write a gzip compressed file.
-no_timestamp	Do not include a time and date in the SDC file.
filename	The name of the file to write the constraints to.

Write the constraints for the design in SDC format to *filename*.

write_sdf	[-corner *corner_name*] [-divider /|.] [-include_typ] [-digits *digits*] [-gzip] [-no_timestamp] [-no_version] *filename*
corner_name	Process corner delays to write.
-divider	Divider to use between hierarchy levels in pin and instance names. Must be '/' or '.'.
-include_typ	Include a 'typ' value in the SDF triple that is the average of min and max delays to satisfy some Verilog simulators that require three values in the delay triples.
digits	The number of digits after the decimal point to report. The default is 4.
-gzip	Write a gzip compressed file.
-no_timestamp	Do not write a DATE statement.
-no_version	Do not write a VERSION statement.
filename	The name of the file to write the constraints to.
digits	The number of digits after the decimal point to report. The default is 4.

Write the delay calculation delays for the design in SDF format to *filename*. The SDF TIMESCALE is same as the time_unit in the first liberty file read.

write_timing_model	[-library_name *lib_name*] [-cell_name *cell_name*] [-corner *corner*] *filename*
lib_name	The name to use for the liberty library. Defaults to *cell_name*.
cell_name	The name to use for the liberty cell. Defaults to the top level module name.

corner	The process corner to use for extracting the model.
filename	Filename to write.

The `write_timing_model` command constructs a liberty timing model for the current design and writes it to *filename*. *`cell_name`* defaults to the cell name of the top level block in the design.

The SDC used to extract the block should include the clock definitions, but not `set_input_delay` or `set_output_delay` commands. Using `set_input_transition` with the slew from the block context will be used will improve the match between the timing model and the block netlist. Clock delays are propagated through the clock network to include clock latencies in the timing model (implicit `set_propagated_clock`). Paths defined on clocks that are defined on internal pins are ignored because the model has no way to include the clock definition.

The resulting timing model can be used in a hierarchical timing flow as a replacement for the block to speed up timing analysis. This hierarchical timing methodology does not handle timing exceptions that originate or terminate inside the block. The timing model includes:

```
combinational paths between inputs and outputs
setup and hold timing constraints on inputs
clock to output timing paths
```

Resistance of long wires on inputs and outputs of the block cannot be modeled in Liberty. To reduce inaccuracies from wire resistance in technologies with resistive wires place buffers on inputs and ouputs.

The extracted timing model setup/hold checks are scalar (no input slew dependence). Delay timing arcs are load dependent but do not include input slew dependency.

---

write_verilog	[-sort] [-include_pwr_gnd] [-remove_cells *lib_cells*] *filename*
-sort	Sort the instances in the netlist.
-include_pwr_gnd	Incluce power and ground pins on instances.
lib_cells	Liberty cells to remove. Use get_lib_cells, a list of cells names, or a cell name with wildcards.
filename	Filename to write.

The `write_verilog` command writes a verilog netlist to *`filename`*. Use `-sort` to sort the instances so the results are reproducible across operating systems. Use `-remove_cells` to remove instances of *`lib_cells`* from the netlist.

Variables

hierarchy_separator Any character.

The hierarchy_separator separates instance names in a hierarchical instance, net, or pin name. The default value is '/'.

link_make_black_boxes 0|1

When link_make_black_boxes is 1 the link_design command will make empty “black box” cells for instances that reference undefined cells. The default value is 1.

sta_bidirect_net_paths_enabled 0|1

When set to 0, paths from bidirectional (inout) ports back through nets are disabled. When set to 1, paths from bidirectional paths from the net back into the instance are enabled. The default value is 0.

sta_continue_on_error 0|1

The source and read_sdc commands stop and report any errors encountered while reading a file unless sta_continue_on_error is 1. The default value is 0.

sta_crpr_mode same_pin|same_transition

When the data and clock paths of a timing check overlap (see sta_crpr_enabled), pessimism is removed independent of whether of the path rise/fall transitions. When sta_crpr_mode is same_transition, the pessimism is only removed if the path rise/fall transitions are the same. The default value is same_pin.

sta_cond_default_arcs_enabled 0|1

When set to 0, default timing arcs with no condition (Liberty timing arcs with no “when” expression) are disabled if there are other conditional timing arcs between the same pins. The default value is 1.

sta_crpr_enabled 0|1

During min/max timing analysis for on_chip_variation the data and clock paths may overlap. For a setup check the maximum path delays are used for the data and the minimum path delays are used for the clock. Because the gates cannot simultaneously have minimum and maximum delays the timing check slack is pessimistic. This pessimism is known as Common Reconvergent Pesssimism Removal, or “CRPR”. Enabling CRPR slows down the analysis. The default value is 1.

sta_dynamic_loop_breaking 0|1

When sta_dynamic_loop_breaking is 0, combinational logic loops are disabled by disabling a timing arc that closes the loop. When sta_dynamic_loop_breaking is 1, all paths around the loop are reported. The default value is 0.

---

sta_gated_clock_checks_enabled 0|1

When sta_gated_clock_checks_enabled is 1, clock gating setup and hold timing checks are checked. The default value is 1.

---

sta_input_port_default_clock 0|1

When sta_input_port_default_clock is 1 a default input arrival is added for input ports that do not have an arrival time specified with the set_input_delay command. The default value is 0.

---

sta_internal_bidirect_instance_paths_enabled 0|1

When set to 0, paths from bidirectional (inout) ports back into the instance are disabled. When set to 1, paths from bidirectional ports back into the instance are enabled. The default value is 0.

---

sta_pocv_enabled 0|1

Enable parametric on chip variation using statistical timing analysis. The default value is 0.

---

sta_propagate_all_clocks 0|1

All clocks defined after sta_propagate_all_clocks is set to 1 are propagated. If it is set before any clocks are defined it has the same effect as

```
set_propagated_clock [all_clocks]
```

after all clocks have been defined. The default value is 0.

---

sta_propagate_gated_clock_enable 0|1

When set to 1, paths of gated clock enables are propagated through the clock gating instances. If the gated clock controls sequential elements setting sta_propagate_gated_clock_enable to 0 prevents spurious paths from the clock enable. The default value is 1.

---

sta_recovery_removal_checks_enabled 0|1

When sta_recovery_removal_checks_enabled is 0, recovery and removal timing checks are disabled. The default value is 1.

sta_report_default_digits integer

The number of digits to print after a decimal point. The default value is 2.

sta_preset_clear_arcs_enabled 0|1

When set to 1, paths through asynchronous preset and clear timing arcs are searched. The default value is 0.

Version 2.0.15, Apr 29, 2019