

# OpenSTA

## Command Line Arguments

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

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

When the STA starts up commands are first read from the user initialization file `~/.sta` if it exists. Next a command passed with the `-x` option is evaluated. Multiple commands can be specified in a quoted command string by separating them with semicolons. If a TCL command file is specified with `-f` on the command line, commands are read from the file and executed. Finally, a TCL command interpreter reads commands from standard input until the `exit` command is executed. 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 liberty.lib
read_verilog design.v
link_design top
create_clock -name clk -period 10 {clk1 clk2 clk3}
set_input_delay -clock clk 0 {in1 in2}
report_checks
```

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 two process corners (`ff` and `ss`) is shown below.

```
read_liberty liberty.lib
read_verilog design.v
link_design top
define_corners ff ss
read_sdf -corner ff reg1.sdf
read_sdf -corner ss reg1_ss.sdf
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 -path_delay min_max -corner ss
report_checks -path_delay min_max -corner ff
```

## 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.

---

|                      |  |
|----------------------|--|
| <b>all_registers</b> | <code>[-clock <i>clock_names</i>]</code><br><code>[-cells]</code><br><code>[-data_pins]</code><br><code>[-clock_pins]</code><br><code>[-async_pins]</code><br><code>[-output_pins]</code><br><code>[-level_sensitive]</code><br><code>[-edge_triggered]</code> |
|----------------------|--|

|                    |   |
|--------------------|---|
| <i>clock_names</i> | A list of clock names. Only registers clocked by these clocks are returned. |
|--------------------|---|

|                     |                                      |
|---------------------|--------------------------------------|
| <code>-cells</code> | Return a list of register instances. |
|---------------------|--------------------------------------|

|                         |                                |
|-------------------------|--------------------------------|
| <code>-data_pins</code> | Return the register data pins. |
|-------------------------|--------------------------------|

|                          |                                 |
|--------------------------|---------------------------------|
| <code>-clock_pins</code> | Return the register clock pins. |
|--------------------------|---------------------------------|

|                          |                                     |
|--------------------------|-------------------------------------|
| <code>-async_pins</code> | Return the register set/clear pins. |
|--------------------------|-------------------------------------|

|                           |                                  |
|---------------------------|----------------------------------|
| <code>-output_pins</code> | Return the register output pins. |
|---------------------------|----------------------------------|

|                               |                                 |
|-------------------------------|---------------------------------|
| <code>-level_sensitive</code> | Return level-sensitive latches. |
|-------------------------------|---------------------------------|

|                              |                                  |
|------------------------------|----------------------------------|
| <code>-edge_triggered</code> | 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 restricts 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.

---

|                                       |   |
|---------------------------------------|---|
| <b>check_setup</b>                    | <code>[-verbose]</code><br><code>[-unconstrained_endpoints]</code><br><code>[-multiple_clock]</code><br><code>[-no_clock]</code><br><code>[-no_input_delay]</code><br><code>[-loops]</code><br><code>[-generated_clocks]</code><br><code>[&gt; filename]</code><br><code>[&gt;&gt; filename]</code> |
| <code>-verbose</code>                 | Show offending objects rather than just error counts.   |
| <code>-unconstrained_endpoints</code> | Check path endpoints for timing constraints (timing check or set_output_delay).   |
| <code>-multiple_clock</code>          | Check register/latch clock pins for multiple clocks.  |
| <code>-no_clock</code>                | Check register/latch clock pins for a clock.  |
| <code>-no_input_delay</code>          | Check for inputs that do not have a set_input_delay command.  |
| <code>-loops</code>                   | Check for combinational logic loops.  |
| <code>-generated_clocks</code>        | 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.

---

|                            |   |
|----------------------------|---|
| <b>connect_pins</b>        | <code>net</code><br><code>port_pin_list</code>            |
| <code>net</code>           | A net to add connections to.                              |
| <code>port_pin_list</code> | A list of ports or pins to connect to <code>net</code> .. |

The `connect_pins` command connects ports or instance pins to a net.

---

|                     |   |
|---------------------|---|
| <b>create_clock</b> | -period period<br>[-name clock_name]<br>[-waveform edge_list]<br>[-add]<br>[pin_list] |
| <i>period</i>       | The clock period.   |
| <i>clock_name</i>   | The name of the clock.  |
| <i>edge_list</i>    | A list of edge rise and fall time.  |
| -add                | Add this clock to the clocks on <i>pin_list</i> .                                     |
| <i>pin_list</i>     | 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 referred 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 waveform 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}
```

---

|                               |  |
|-------------------------------|--|
| <b>create_generated_clock</b> | [-name <i>clock_name</i> ]<br>-source <i>master_pin</i><br>[-master_clock <i>master_clock</i> ]<br>[-pll_out <i>pll_out_pin</i> ]<br>[-pll_feedback <i>pll_fdbk_pin</i> ]<br>[-divide_by <i>divisor</i> ]<br>[-multiply_by <i>multiplier</i> ]<br>[-duty_cycle <i>duty_cycle</i> ]<br>[-invert]<br>[-edges <i>edge_list</i> ]<br>[-edge_shift <i>shift_list</i> ]<br>[-add]<br><i>pin_list</i> |
| <i>clock_name</i>             | The name of the generated clock.   |
| <i>master_pin</i>             | A pin the the fanout of the master clock that is the source of the generated clock.  |
| <i>master_clock</i>           | The master clock used to generate the clock waveform.  |
| <i>pll_out_pin</i>            | The pin from <i>pin_list</i> that is the phase locked loop output pin.   |
| <i>pll_fdbk_pin</i>           | A pin in the fanout of the <i>pll_out_pin</i> that the phased locked loop phase locks to the master clock pin.   |
| <i>divisor</i>                | Divide the master clock period by <i>divisor</i> .   |
| <i>multiplier</i>             | Multiply the master clock period by <i>multiplier</i> .  |
| <i>duty_cycle</i>             | The percent of the period that the generated clock is high (between 0 and 100).  |
| -invert                       | Invert the master clock.   |
| <i>edge_list</i>              |  |
| <i>shift_list</i>             | Not supported.   |
| -add                          | Add this clock to the clocks on <i>pin_list</i> .  |
| <i>pin_list</i>               | A list of pins driven by the 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  $\text{period} * \text{duty\_cycle} / 100$ . 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.

---

|                            |  |
|----------------------------|--|
| <b>create_voltage_area</b> | <code>[-name name]</code>              |
|                            | <code>[-coordinate coordinates]</code> |
|                            | <code>[-guard_band_x guard_x]</code>   |
|                            | <code>[-guard_band_y guard_y]</code>   |
|                            | <code>cells</code>                     |

This command is parsed and ignored by timing analysis.

---

|                       |                   |
|-----------------------|-------------------|
| <b>current_design</b> | [ <i>design</i> ] |
|-----------------------|-------------------|

---

|                         |                     |
|-------------------------|---------------------|
| <b>current_instance</b> | [ <i>instance</i> ] |
|-------------------------|---------------------|

|                 |                |
|-----------------|----------------|
| <i>instance</i> | Not supported. |
|-----------------|----------------|

---

|                       |  |
|-----------------------|--|
| <b>define_corners</b> | <i>corner_name1</i> [ <i>corner_name2</i> ]... |
|-----------------------|--|

|                    |   |
|--------------------|---|
| <i>corner_name</i> | 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` 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.

---

|                        |                      |
|------------------------|----------------------|
| <b>delete_instance</b> | <i>instance_list</i> |
|------------------------|----------------------|

|                      |                                |
|----------------------|--------------------------------|
| <i>instance_list</i> | A list of instances to remove. |
|----------------------|--------------------------------|

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

---

|                   |                 |
|-------------------|-----------------|
| <b>delete_net</b> | <i>net_list</i> |
|-------------------|-----------------|

|                 |                           |
|-----------------|---------------------------|
| <i>net_list</i> | A list of nets to remove. |
|-----------------|---------------------------|

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

---

|                        |  |
|------------------------|--|
| <b>disconnect_pins</b> | <i>net</i><br><i>port_pin_list</i> /-all |
|------------------------|--|

|            |                                  |
|------------|----------------------------------|
| <i>net</i> | The net to disconnect pins from. |
|------------|----------------------------------|

|                      |   |
|----------------------|---|
| <i>port_pin_list</i> | A list of ports or pins to disconnect from <i>net</i> . |
|----------------------|---|

`-all` Disconnect all pins from the net.

Disconnects pins from a net. Parasitics connected to the pins 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.



|  |  |
|--|--|
| <code>-path_delay max</code>           | Report max path (setup) checks.  |
| <code>-path_delay max_rise</code>      | Report max path (setup) checks for rising endpoints.   |
| <code>-path_delay max_fall</code>      | Report max path (setup) checks for falling endpoints.  |
| <code>-path_delay min_max</code>       | Report max and max path (setup and hold) checks.   |
| <code>path_count</code>                | The number of paths to report in each path group.  |
| <code>endpoint_path_count</code>       | The number of paths to report for each endpoint.   |
| <code>-unique_paths_to_endpoint</code> | Report multiple paths to an endpoint that traverse different pins without showing multiple paths with different rise/fall transitions. |
| <code>corner_name</code>               | Only report paths for one process corner.  |
| <code>max_slack</code>                 | Only report paths with less slack than <code>max_slack</code> .  |
| <code>min_slack</code>                 | Only report paths with more slack than <code>min_slack</code> .  |
| <code>-sort_by_slack</code>            | Sort paths by slack rather than slack grouped by path group.   |
| <code>group_names</code>               | 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.

---

|                               |  |
|-------------------------------|--|
| <b><code>get_cells</code></b> | <code>[-hierarchical]</code><br><code>[-hsc separator]</code><br><code>[-filter expr]</code><br><code>[-regexp]</code><br><code>[-nocase]</code><br><code>[-quiet]</code><br><code>[-of_objects objects]</code><br><code>[patterns]</code> |
| <code>-hierarchical</code>    | Searches hierarchy levels below the current instance for matches.  |
| <code>separator</code>        | Character to use to separate hierarchical instance names in <code>patterns</code> .  |

|                 |   |
|-----------------|---|
| <i>expr</i>     | A filter expression of the form<br><code>attribute == ~= pattern</code><br>where <i>attribute</i> is an attribute supported by the <code>get_property</code> command.   |
| <i>objects</i>  | The name of a pin or net, a list of pins returned by <code>get_pins</code> , or a list of nets returned by <code>get_nets</code> . The <code>-hierarchical</code> option cannot be used with <code>-of_objects</code> . |
| <i>patterns</i> | 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`.

---

|                     |  |
|---------------------|--|
| <b>get_clocks</b>   | <code>[-regexp]</code><br><code>[-nocase]</code><br><code>[-quiet]</code><br><i>patterns</i> |
| <code>-quiet</code> | Do not report an error if <i>patterns</i> do not match anything.                             |
| <i>patterns</i>     | 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`.

---

|                  |   |
|------------------|---|
| <b>get_fanin</b> | <code>-to sink_list</code><br><code>[-flat]</code><br><code>[-only_cells]</code><br><code>[-startpoints_only]</code><br><code>[-levels level_count]</code><br><code>[-pin_levels pin_count]</code><br><code>[-trace_arcs timing enabled all]</code> |
|------------------|---|

List of pins, ports, or nets to find the fanin of.  
For nets, the fanin of driver pins on the nets are  
returned.

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.

Return the instances connected to the pins in the fanin.

Only return pins that are startpoints.

Only return pins within `level_count` instance traversals.

Only return pins within *pin\_count* pin traversals.

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.

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

|                    |   |
|--------------------|---|
| <i>source_list</i> | 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.  |
| <i>level_count</i> | Only return pins within <i>level_count</i> instance traversals.   |
| <i>pin_count</i>   | Only return pins within <i>pin_count</i> 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.

---

|                      |  |
|----------------------|--|
| <b>get_lib_cells</b> | [-of_objects <i>objects</i> ]<br>[-hsc <i>separator</i> ]<br>[-regexp]<br>[-nocase]<br>[-quiet]<br><i>patterns</i> |
|----------------------|--|

---

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

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`.

---

|                     |  |
|---------------------|--|
| <b>get_lib_pins</b> | [-hsc <i>separator</i> ]<br>[-regexp]<br>[-nocase]<br>[-quiet]<br><i>patterns</i>      |
| <i>separator</i>    | Character that separates the library name, cell name and port name in <i>pattern</i> . |
| -quiet              | Do not report an error if <i>patterns</i> do not match anything.                       |
| <i>patterns</i>     | 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`.

---

|                 |  |
|-----------------|--|
| <b>get_libs</b> | [-regexp]<br>[-nocase]<br>[-quiet]<br><i>patterns</i>            |
| -quiet          | Do not report an error if <i>patterns</i> do not match anything. |
| <i>patterns</i> | 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`.

---

|                 |   |
|-----------------|---|
| <b>get_nets</b> | [-hierarchical]<br>[-hsc <i>separator</i> ]<br>[-regexp]<br>[-nocase]<br>[-quiet]<br>[-of_objects <i>objects</i> ]<br>[ <i>patterns</i> ] |
|-----------------|---|

|                            |  |
|----------------------------|--|
| <code>-hierarchical</code> | Searches hierarchy levels below the current instance for matches.  |
| <code>separator</code>     | Character that separates hierarchical instance names and the net name in <i>pattern</i> .  |
| <code>-quiet</code>        | Do not report an error if <i>patterns</i> do not match anything.   |
| <code>objects</code>       | The name of a pin or instance, a list of pins returned by <code>get_pins</code> , or a list of instances returned by <code>get_cells</code> . The <code>-hierarchical</code> option cannot be used with <code>-of_objects</code> . |
| <code>patterns</code>      | 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`.

---

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

*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`.

---

|                     |  |
|---------------------|--|
| <b>get_ports</b>    | <code>[-filter <i>expr</i>]</code><br><code>[-regexp]</code><br><code>[-nocase]</code><br><code>[-quiet]</code><br><code>[-of_objects <i>objects</i>]</code><br><code>[<i>patterns</i>]</code> |
| <code>-quiet</code> | Do not report an error if <i>patterns</i> do not match any clocks.   |
| <i>expr</i>         | A filter expression of the form<br><code>attribute == ~= pattern</code><br>where <i>attribute</i> is an attribute supported by the <code>get_property</code> command.                          |
| <i>objects</i>      | The name of a net, or a list of nets returned by <code>get_nets</code> .   |
| <i>patterns</i>     | 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`.

---

|                     |  |
|---------------------|--|
| <b>get_property</b> | <code>[-object_type <i>object_type</i>]</code><br><i>object</i><br><i>property</i> |
| <i>object_type</i>  | <code>cell pin net port clock timing_arc</code>                                    |
| <i>object</i>       | An object or object name if <code>-object_type</code> is specified.                |
| <i>property</i>     | A property name.   |

The properties for different objects types are shown below.

instance (cell)

ref\_name  
full\_name

## pin

direction  
full\_name  
lib\_pin\_name  
clocks  
actual\_fall\_transition\_min  
actual\_fall\_transition\_max  
actual\_rise\_transition\_min  
actual\_rise\_transition\_max  
min\_fall\_slack  
max\_fall\_slack  
min\_rise\_slack  
max\_rise\_slack

## port

direction  
full\_name  
actual\_fall\_transition\_min  
actual\_fall\_transition\_max  
actual\_rise\_transition\_min  
actual\_rise\_transition\_max  
min\_fall\_slack  
max\_fall\_slack  
min\_rise\_slack  
max\_rise\_slack

## net

full\_name

## liberty\_port

direction  
full\_name

## liberty\_cell

base\_name  
full\_name

## library

name

## clock

name  
period  
sources

## edge



delay\_min\_fall  
delay\_max\_fall  
delay\_min\_rise  
delay\_max\_rise

path(PathEnd)

endpoint  
endpoint\_clock  
endpoint\_clock\_pin  
slack  
startpoint  
startpoint\_clock  
points

point (PathRef)

pin  
arrival  
required  
slack

---

|                         |   |
|-------------------------|---|
| <b>get_timing_edges</b> | <code>[-from <i>from_pins</i>]</code><br><code>[-to <i>to_pins</i>]</code><br><code>[-of_objects <i>objects</i>]</code><br><code>[-filter <i>expr</i>]</code><br><code>[<i>patterns</i>]</code> |
| <i>from_pin</i>         | A collection of pins.   |
| <i>to_pin</i>           | A collection of pins.   |
| <i>objects</i>          | A collection of instances or library cells. The <code>-from</code> and <code>-to</code> options cannot be used with <code>-of_objects</code> .  |
| <i>expr</i>             | A filter expression of the form<br><code>attribute == ~= pattern</code><br>where <i>attribute</i> is an attribute supported by the <code>get_property</code> 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`.

---

|                     |   |
|---------------------|---|
| <b>group_path</b>   | -name <i>group_name</i><br>[-weight <i>weight</i> ]<br>[-critical_range <i>range</i> ]<br>[-from <i>from_list</i> ]<br>[-rise_from <i>from_list</i> ]<br>[-fall_from <i>from_list</i> ]<br>[-through <i>through_list</i> ]<br>[-rise_through <i>through_list</i> ]<br>[-fall_through <i>through_list</i> ]<br>[-to <i>to_list</i> ]<br>[-rise_to <i>to_list</i> ]<br>[-fall_to <i>to_list</i> ]<br> |
| <i>group_name</i>   | The name of the path group.   |
| <i>weight</i>       | Not supported.  |
| <i>range</i>        | Not supported.  |
| <i>from_list</i>    | A list of clocks, instances, ports or pins.   |
| <i>through_list</i> | A list of instances, pins or nets.  |
| <i>to_list</i>      | 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.

---

|             |                      |
|-------------|----------------------|
| <b>link</b> | [ <i>cell_name</i> ] |
|-------------|----------------------|

Alias for `link_design`.

---

|                    |                      |
|--------------------|----------------------|
| <b>link_design</b> | [ <i>cell_name</i> ] |
|--------------------|----------------------|

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.

---

|                      |   |
|----------------------|---|
| <b>make_instance</b> | <i>inst_names</i><br><i>lib_cell_name</i> |
| <i>inst_names</i>    | A list of instance names.                 |
| <i>lib_cell_name</i> | The library cell of the new instances.    |

The `make_instance` command makes instances of library cell *lib\_cell\_name*.

---

|                      |                      |
|----------------------|----------------------|
| <b>make_net</b>      | <i>net_name_list</i> |
| <i>net_name_list</i> | A list of net names. |

Creates a net for each hierarchical net name.

---

|                     |  |
|---------------------|--|
| <b>read_liberty</b> | <code>[-corner <i>corner_name</i>]</code><br><code>[-min]</code><br><code>[-max]</code><br><code>[-no_latch_infer]</code><br><i>filename</i> |
| <i>corner_name</i>  | Use library for process corner <i>corner_name</i> delay calculation.   |
| <code>-min</code>   | Use library for min delay calculation.   |
| <code>-max</code>   | Use library for max delay calculation.   |
| <i>filename</i>     | 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 (inferred_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 inferred. If a cell has the `interface_timing true` attribute, no latches are inferred in the cell.

---

|                        |   |
|------------------------|---|
| <b>read_parasitics</b> | [-min]<br>[-max]<br>[-elmore]<br>[-path path]<br>[-analysis_point ap]<br>[-increment]<br>[-keep_capacitive_coupling]<br>[-coupling_reduction_factor factor]<br>[-reduce_to pi_elmore pi_pole_residue2]<br>[-delete_after_reduce]<br>[-quiet]<br>[-save]<br><i>filenames</i> |
| <b>-elmore</b>         | Ignored (for compatibility purposes only).  |
| <b>ap</b>              | The name of a parasitics analysis point.<br>min - annotate parasitics for min paths.<br>max - annotate parasitics for max paths.<br>min_max - annotate parasitics for min and max paths.<br>The default value is min_max.   |
| <b>path</b>            | Hierarchical instance path to annotate with parasitics.   |
| <b>-increment</b>      | Incrementally annotate parasitics (do not remove existing parasitics on annotated nets).  |

|  |  |
|--|--|
| <code>-keep_capacitive_coupling</code> | Keep coupling capacitors in parasitic networks rather than converting them to grounded capacitors. |
| <i>factor</i>                          | Factor to multiply coupling capacitance by when reducing parasitic networks.                       |
| <code>-reduce_to</code>                | Reduce detailed parasitics to a PI/Elmore or PI/Pole residue model as each net is read.            |
| <code>-delete_after_reduce</code>      | Delete the detailed parasitic network after reducing it.   |
| <code>-quiet</code>                    | Do not print error or warning messages.  |
| <code>-save</code>                     | Save the parasitics database after reading it (OpenAccess only).                                   |
| <i>filename</i>                        | The name of the parasitics file to read.   |

The `read_parasitics` command reads a file of net parasitics in RSPF, DSPF or SPEF format.

Files compressed with gzip are automatically uncompressed. Compression can reduce the file to less than 10% of its original size.

Separate parasitics can be annotated for min and max paths using the `-min` and `-max` arguments.

With the `-reduce_to` and `-delete_after_reduce` options, parasitic networks (DSPF) 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 SPF constructs are ignored.

```
*|I pin capacitances in instance pin declarations.
*|P external loads defined in pin declarations (use set_load instead).
X The instance section of an spf file is ignored.
```

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.

---

|                       |   |
|-----------------------|---|
| <b>read_sdc</b>       | <code>[-echo]</code><br><code>filename</code>             |
| <code>-echo</code>    | Print each command before evaluating it.SDC command file. |
| <code>filename</code> | 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.

---

|                                  |  |
|----------------------------------|--|
| <b>read_sdf</b>                  | <code>[-analysis_type single bc_wc on_chip_variation]</code><br><code>[-corner corner_name]</code><br><code>[-type sdf_min sdf_typ sdf_max]</code><br><code>[-min_type sdf_min sdf_typ sdf_max]</code><br><code>[-max_type sdf_min sdf_typ sdf_max]</code><br><code>[-unescaped_dividers]</code><br><code>filename</code>  |
| <code>-analysis_type</code>      | See <code>set_operating_conditions</code> for a description of the analysis types.   |
| <code>corner_name</code>         | Process corner delays to annotate.   |
| <code>-type</code>               | Which of the three values in the SDF tuples to read for analysis type single.  |
| <code>-min_type</code>           | Which of the three values in the SDF tuples to read for min paths for analysis type <code>bc_wc</code> or <code>on_chip_variation</code> .   |
| <code>-max_type</code>           | Which of the three values in the SDF tuples to read for max paths for analysis type <code>bc_wc</code> or <code>on_chip_variation</code> .   |
| <code>-unescaped_dividers</code> | 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 " <code>\inst1\inst2</code> " can be referenced as " <code>inst1/inst2</code> ". The correct SDF name is " <code>inst1\inst2</code> ", since the divider does not represent a change in hierarchy in this case. |
| <code>filename</code>            | The name of the SDF file to read.  |

Files compressed with gzip are automatically uncompressed. Compression can reduce the file to less than 10% of its original size.

INCREMENT is supported as an alias for INCREMENTAL.

The following SDF statements are not supported.

```
PORT
INSTANCE wildcards
```

---

|                     |                 |
|---------------------|-----------------|
| <b>read_verilog</b> | <i>filename</i> |
|---------------------|-----------------|

|                 |                                       |
|-----------------|---------------------------------------|
| <i>filename</i> | 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. *filename* can be compressed with gzip.

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

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

---

|                     |   |
|---------------------|---|
| <b>replace_cell</b> | <i>instance_list</i><br><i>replacement_cell</i> |
|---------------------|---|

|                      |                                       |
|----------------------|---------------------------------------|
| <i>instance_list</i> | A list of instances to swap the cell. |
|----------------------|---------------------------------------|

|                         |                           |
|-------------------------|---------------------------|
| <i>replacement_cell</i> | 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.

---

|                               |  |
|-------------------------------|--|
| <b>report_annotated_check</b> | <code>[-setup]</code><br><code>[-hold]</code><br><code>[-recovery]</code><br><code>[-removal]</code><br><code>[-nochange]</code><br><code>[-width]</code><br><code>[-period]</code><br><code>[-max_skew]</code><br><code>[-max_line lines]</code><br><code>[-list_annotated]</code><br><code>[-list_not_annotated]</code><br><code>[-constant_arcs]</code> |
|-------------------------------|--|

|                     |                                |
|---------------------|--------------------------------|
| <code>-setup</code> | Report annotated setup checks. |
|---------------------|--------------------------------|

|                    |                               |
|--------------------|-------------------------------|
| <code>-hold</code> | Report annotated hold checks. |
|--------------------|-------------------------------|

|                        |                                   |
|------------------------|-----------------------------------|
| <code>-recovery</code> | Report annotated recovery checks. |
|------------------------|-----------------------------------|

|                                  |  |
|----------------------------------|--|
| <code>-removal</code>            | Report annotated removal checks.   |
| <code>-nochange</code>           | Report annotated nochange checks.  |
| <code>-width</code>              | Report annotated width checks.   |
| <code>-period</code>             | Report annotated period checks.  |
| <code>-max_skew</code>           | Report annotated max skew checks.  |
| <i>lines</i>                     | Maximum number of lines listed by the <code>-list_annotated</code> and <code>-list_not_annotated</code> options.                     |
| <code>-list_annotated</code>     | List annotated timing arcs.  |
| <code>-list_not_annotated</code> | List unannotated timing arcs.  |
| <code>-constant_arcs</code>      | Report separate annotation counts for arcs disabled by logic constants ( <code>set_logic_one</code> , <code>set_logic_zero</code> ). |

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.

---

|  |   |
|--|---|
| <b><code>report_annotated_delay</code></b> | <code>[-cell]</code><br><code>[-net]</code><br><code>[-from_in_ports]</code><br><code>[-to_out_ports]</code><br><code>[-max_line lines]</code><br><code>[-list_annotated]</code><br><code>[-list_not_annotated]</code><br><code>[-constant_arcs]</code> |
| <code>-cell</code>                         | Report annotated cell delays.   |
| <code>-net</code>                          | Report annotated internal net delays.   |
| <code>-from_in_ports</code>                | Report annotated delays from input ports.   |
| <code>-to_out_ports</code>                 | Report annotated delays to output ports.  |
| <i>lines</i>                               | Maximum number of lines listed by the <code>-list_annotated</code> and <code>-list_not_annotated</code> options.  |
| <code>-list_annotated</code>               | List annotated timing arcs.   |
| <code>-list_not_annotated</code>           | 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`.

---

|                      |   |
|----------------------|---|
| <b>report_checks</b> | <pre> [-from <i>from_list</i>    -rise_from <i>from_list</i>    -fall_from <i>from_list</i>] [-through <i>through_list</i>    -rise_through <i>through_list</i>    -fall_through <i>through_list</i>] [-to <i>to_list</i>    -rise_to <i>to_list</i>    -fall_to <i>to_list</i>] [-unconstrained] [-path_delay min min_rise min_fall                          max max_rise max_fall                          min_max] [-group_count <i>path_count</i>] [-endpoint_count <i>endpoint_path_count</i>] [-unique_paths_to_endpoint] [-corner <i>corner_name</i>] [-slack_max <i>max_slack</i>] [-slack_min <i>min_slack</i>] [-sort_by_slack] [-path_group <i>group_names</i>] [-format end full short summary                          full_clock full_clock_expanded] [-fields <i>fields</i>] [-digits <i>digits</i>] [-no_line_split] [&gt; <i>filename</i>] [&gt;&gt; <i>filename</i>] </pre> |
| <i>from_list</i>     | A list of clocks, instances, ports or pins.   |
| <i>through_list</i>  | A list of instances, pins or nets.  |
| <i>to_list</i>       | A list of clocks, instances, ports or pins.   |
| -unconstrained       | Report 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.   |

|  |   |
|--|---|
| <code>-path_delay max_rise</code>        | Report max path (setup) checks for rising endpoints.  |
| <code>-path_delay max_fall</code>        | Report max path (setup) checks for falling endpoints.   |
| <code>-path_delay min_max</code>         | Report max and max path (setup and hold) checks.  |
| <code>path_count</code>                  | The number of paths to report in each path group.   |
| <code>endpoint_path_count</code>         | The number of paths to report for each endpoint.  |
| <code>-unique_paths_to_endpoint</code>   | Report multiple paths to an endpoint that traverse different pins without showing multiple paths with different rise/fall transitions.  |
| <code>corner_name</code>                 | Only report paths for one process corner.   |
| <code>max_slack</code>                   | Only report paths with less slack than <code>max_slack</code> .   |
| <code>min_slack</code>                   | Only report paths with more slack than <code>min_slack</code> .   |
| <code>-sort_by_slack</code>              | Sort paths by slack rather than slack grouped by path group.  |
| <code>group_names</code>                 | List of path group names to report. All path groups are reported if this option is not specified.   |
| <code>-format end</code>                 | Report path ends with delay, required time and slack.   |
| <code>-format full</code>                | Report path start and end points and the path. This is the default path type.   |
| <code>-format full_clock</code>          | Report path start and end points, the path, and the source and and target clock paths.  |
| <code>-format full_clock_expanded</code> | 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. |
| <code>-format short</code>               | Report path start and end points.   |
| <code>-format summary</code>             | Report path ends and delay.   |
| <code>fields</code>                      | List of capacitance transition_time input_pins nets fanout  |
| <code>digits</code>                      | The number of digits after the decimal point to report. The default value is the variable <code>default_significant_digits</code> .   |

`-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\_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.

---

|                           |   |
|---------------------------|---|
| <b>report_check_types</b> | [-all_violators]<br>[-verbose]<br>[-format slack_only end]<br>[-max_delay]<br>[-min_delay]<br>[-recovery]<br>[-removal]<br>[-clock_gating_setup]<br>[-clock_gating_hold]<br>[-max_transition]<br>[-min_transition]<br>[-min_pulse_width]<br>[-min_period]<br>[-digits <i>digits</i> ]<br>[-no_split_lines]<br>[> <i>filename</i> ]<br>[>> <i>filename</i> ] |
| -all_violators            | All violated timing and design rule constraints are reported.   |
| -verbose                  | Use a verbose output format.  |
| -format                   | slack_only reports the minimum slack for each check.<br>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_transition           | Report max transition design rule checks.   |
| -max_skew                 | Report max skew design rule checks.   |
| -min_pulse_width          | Report min pulse width design rule checks.  |

|                              |  |
|------------------------------|--|
| <code>-min_period</code>     | Report min period design rule checks.  |
| <code>-min_transition</code> | Report min transition design rule checks.  |
| <code>digits</code>          | The number of digits after the decimal point to report. The default is <code>default_significant_digits</code> . |
| <code>-no_split_lines</code> |  |

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.

---

|                                  |   |
|----------------------------------|---|
| <b><code>report_dcalc</code></b> | <code>[-from from_pin]</code><br><code>[-to to_pin]</code><br><code>[-corner corner_name]</code><br><code>[-min]</code><br><code>[-max]</code><br><code>[-digits digits]</code><br><code>[&gt; filename]</code><br><code>[&gt;&gt; filename]</code> |
| <code>from_pin</code>            | An instance pin.  |
| <code>to_pin</code>              | An instance pin.  |
| <code>digits</code>              | The number of digits after the decimal point to report. The default is <code>default_significant_digits</code> .  |

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_disable_timing
ul A B constant B=0
```

---

|                            |  |
|----------------------------|--|
| <b>report_instance</b>     | <code>[-connections]</code><br><code>[-verbose]</code><br><code>instance_path</code><br><code>[&gt; filename]</code><br><code>[&gt;&gt; filename]</code> |
| <code>-connections</code>  | Report the pins connected to the net.  |
| <code>-verbose</code>      | Report the capacitance of each pin.  |
| <code>instance_path</code> | Hierarchical path to a instance.   |

---

|                        |  |
|------------------------|--|
| <b>report_lib_cell</b> | <code>cell_name</code><br><code>[&gt; filename]</code><br><code>[&gt;&gt; filename]</code> |
| <code>cell_name</code> | The name of a library cell.  |

Describe the liberty library cell *cell\_name*.

---

|                           |  |
|---------------------------|--|
| <b>report_net</b>         | <code>[-connections]</code><br><code>[-verbose]</code><br><code>[-significant_digits digits]</code><br><code>net_path</code><br><code>[&gt; filename]</code><br><code>[&gt;&gt; filename]</code> |
| <code>-connections</code> | Report the net pins.   |
| <code>-verbose</code>     | With <code>-connections</code> also report all pins connected to each instance pin net.  |
| <code>net_path</code>     | Hierarchical path to a net.  |

---

|                   |   |
|-------------------|---|
| <b>report_pin</b> | <code>pin_path</code><br><code>[&gt; filename]</code><br><code>[&gt;&gt; filename]</code> |
|-------------------|---|

*pin\_path* Hierarchical path to a pin.

---

**report\_power** [-instance *instance*]  
[-digits *digits*]  
[> *filename*]  
[>> *filename*]

*instance* Report the power for *instance*.

*digits* The number of digits after the decimal point to report. The default is default\_significant\_digits.

Report power used by the design or a specific instance. The internal, switching, leakage and total power are reported. Design power is reported separately for combinational, sequential, macro and pad groups.

---

**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.



---

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

---

|                           |   |
|---------------------------|---|
| <b>set_assigned_delay</b> | -cell -net<br>[-rise]<br>[-fall]<br>[-corner <i>corner_name</i> ]<br>[-min]<br>[-max]<br>[-from <i>from_pins</i> ]<br>[-to <i>to_pins</i> ]<br><i>delay</i> |
| -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.  |
| <i>from_pins</i>          | A list of pins.   |
| <i>to_pins</i>            | A list of pins.   |
| <i>delay</i>              | The delay between <i>from_pins</i> and <i>to_pins</i> .   |

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.

---

|                                |   |
|--------------------------------|---|
| <b>set_assigned_transition</b> | [-rise]<br>[-fall]<br>[-corner <i>corner_name</i> ]<br>[-min]<br>[-max]<br><i>slew</i><br><i>pin_list</i> |
| -rise                          | Annotate the rising transition.   |

|                       |                                       |
|-----------------------|---------------------------------------|
| <code>-fall</code>    | Annotate the falling transition.      |
| <code>-min</code>     | Annotate the minimum transition time. |
| <code>-max</code>     | Annotate the maximum transition time. |
| <code>slew</code>     | The pin transition time.              |
| <code>pin_list</code> | 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.

---

|                                       |   |
|---------------------------------------|---|
| <b><code>set_case_analysis</code></b> | <code>0 1 zero one rise rising fall falling</code><br><code>port_or_pin_list</code> |
| <code>port_or_pin_list</code>         | 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 in all modes (unlike `set_logic_zero`, etc).

---

|  |  |
|--|--|
| <b><code>set_clock_gating_check</code></b> | <code>[-setup <i>setup_time</i>]</code><br><code>[-hold <i>hold_time</i>]</code><br><code>[-rise]</code><br><code>[-fall]</code><br><code>[-high]</code><br><code>[-low]</code><br><code>[<i>objects</i>]</code> |
| <code>setup_time</code>                    | Clock enable setup margin.   |
| <code>hold_time</code>                     | Clock enable hold margin.  |
| <code>-rise</code>                         | The setup/hold margin is for the rising edge of the clock enable.  |
| <code>-fall</code>                         | The setup/hold margin is for the falling edge of the clock enable.   |

|                    |  |
|--------------------|--|
| <code>-high</code> | The gating clock is active high (pin and instance objects only). |
| <code>-low</code>  | The gating clock is active low (pin and instance objects only).  |
| <i>objects</i>     | 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 the priority of the commands is shown below (highest to lowest priority).

```

clock enable pin
instance
clock pin
clock
global

```

|                                    |   |
|------------------------------------|---|
| <b>set_clock_groups</b>            | <code>[-name <i>name</i>]</code><br><code>[-logically_exclusive]</code><br><code>[-physically_exclusive]</code><br><code>[-asynchronous]</code><br><code>[-allow_paths]</code><br><code>-group <i>clocks</i></code> |
| <i>name</i>                        | The clock group name.   |
| <code>-logically_exclusive</code>  | 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.  |
| <code>-physically_exclusive</code> | The clocks in different groups cannot be present at the same time on a chip. Paths between clock groups are <i>not</i> considered for noise analysis.   |
| <code>-asynchronous</code>         | 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 define 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_sense`**            `[-positive]`  
                                 `[-negative]`  
                                 `[-pulse pulse_type]`  
                                 `[-stop_propagation]`  
                                 `[-clock clocks]`  
                                 *pins*

|                                |  |
|--------------------------------|--|
| <code>-positive</code>         | The clock sense is positive unite.   |
| <code>-negative</code>         | The clock sense is negative unite.   |
| <code>pulse_type</code>        | rise_triggered_high_pulse<br>rise_triggered_low_pulse<br>fall_triggered_high_pulse<br>fall_triggered_low_pulse |
| <code>-stop_propagation</code> | Stop propagating <i>clocks</i> clocks at <i>pins</i> .   |
| <code>clocks</code>            | A list of clocks to apply the sense  |
| <code>pins</code>              | A list of pins.  |

The `set_clock_sense` command is used to modify the propagation of a clock signal. 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.

|  |  |
|--|--|
| <b><code>set_clock_transition</code></b> | [-rise]<br>[-fall]<br>[-min]<br>[-max]<br><i>transition</i><br><i>clocks</i> |
| <code>-rise</code>                       | The transition time is for the rising edge of the clock.                     |
| <code>-fall</code>                       | The transition time is for the falling edge of the clock.                    |
| <code>-min</code>                        | <i>transition</i> is the minimum transition time.                            |
| <code>-max</code>                        | <i>transition</i> is the maximum transition time.                            |
| <i>transition</i>                        | Clock transition time (slew).  |
| <i>clocks</i>                            | A list of clocks.  |

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

---

|                              |   |
|------------------------------|---|
| <b>set_clock_uncertainty</b> | [-from -rise_from -fall_from <i>from_clock</i> ]<br>[-to -rise_to -fall_to <i>to_clock</i> ]<br>[-rise]<br>[-fall]<br>[-setup]<br>[-hold]<br><i>uncertainty</i><br><i>objects</i> |
| <i>from_clock</i>            |   |
| <i>to_clock</i>              |   |
| -rise                        | The uncertainty is for the rising edge of the clock.  |
| -fall                        | The uncertainty is for the falling edge of the clock.   |
| -setup                       | <i>uncertainty</i> is the setup check uncertainty.  |
| -hold                        | <i>uncertainty</i> is the hold uncertainty.   |
| <i>uncertainty</i>           | Clock uncertainty.  |
| <i>objects</i>               | A list of clocks, ports or pins.  |

The `set_clock_uncertainty` command specifies the uncertainty or jitter in a clock.

---

|                       |  |
|-----------------------|--|
| <b>set_data_check</b> | [-from <i>from_object</i> ]<br>[-rise_from <i>from_object</i> ]<br>[-fall_from <i>from_object</i> ]<br>[-to <i>to_object</i> ]<br>[-rise_to <i>to_object</i> ]<br>[-fall_to <i>to_object</i> ]<br>[-setup]<br>[-hold]<br>[-clock <i>clock</i> ]<br><i>margin</i> |
| <i>from_object</i>    | A pin used as the timing check reference.  |
| <i>to_object</i>      | A pin that the setup/hold check is applied to.   |
| -setup                | Add a setup timing check.  |

|                     |                                |
|---------------------|--------------------------------|
| <code>-hold</code>  | Add a hold timing check.       |
| <code>clock</code>  | The setup/hold check clock.    |
| <code>margin</code> | 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*

|                |  |
|----------------|--|
| <i>objects</i> | 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*

|                |   |
|----------------|---|
| <i>objects</i> | A list of instances, ports, pins, cells or [library/]cell/port. |
|----------------|---|

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

All timing paths through 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
```

---

|                    |   |
|--------------------|---|
| <b>set_drive</b>   | <code>[-rise]</code><br><code>[-fall]</code><br><code>[-max]</code><br><code>[-min]</code><br><i>resistance</i><br><i>port_list</i> |
| <code>-rise</code> | This is the drive resistance of the rising edge of the input.   |
| <code>-fall</code> | This is the drive resistance of the falling edge of the input.  |
| <code>-max</code>  | This is the drive resistance for maximum path delays.   |
| <code>-min</code>  | This is the drive resistance for minimum path delays.   |
| <i>resistance</i>  | The external drive resistance.  |
| <i>port_list</i>   | A list of ports.  |

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

---

|                         |   |
|-------------------------|---|
| <b>set_driving_cell</b> | <code>[-lib_cell cell_name]</code><br><code>[-library library]</code><br><code>[-rise]</code><br><code>[-fall]</code><br><code>[-min]</code><br><code>[-max]</code><br><code>[-pin pin]</code><br><code>[-from_pin from_pin]</code><br><code>[-input_transition_rise trans_rise]</code><br><code>[-input_transition_fall trans_fall]</code><br><i>port_list</i> |
| <i>cell_name</i>        | The cell of driver.   |

|                   |   |
|-------------------|---|
| <i>library</i>    | The library of the driving cell.                                    |
| <i>-rise</i>      | This is the driving cell for the rising edge of the input.          |
| <i>-fall</i>      | This is the driving cell for the falling edge of the input.         |
| <i>-max</i>       | This is the driving cell for maximum path delays.                   |
| <i>-min</i>       | This is the driving cell for minimum path delays.                   |
| <i>pin</i>        | The output port of the driving cell.                                |
| <i>from_pin</i>   | Use paths through the driving cell from this pin to the output pin. |
| <i>trans_rise</i> | The transition time for a rising input at <i>from_pin</i> .         |
| <i>trans_fall</i> | The transition time for a falling input at <i>from_pin</i> .        |
| <i>port_list</i>  | A list of ports.  |

The `set_driving_cell` command describes an input port external driver.

---

|                       |   |
|-----------------------|---|
| <b>set_false_path</b> | [-setup]<br>[-hold]<br>[-rise]<br>[-fall]<br>[-from <i>from_list</i> ]<br>[-rise_from <i>from_list</i> ]<br>[-fall_from <i>from_list</i> ]<br>[-through <i>through_list</i> ]<br>[-rise_through <i>through_list</i> ]<br>[-fall_through <i>through_list</i> ]<br>[-to <i>to_list</i> ]<br>[-rise_to <i>to_list</i> ]<br>[-fall_to <i>to_list</i> ]<br>[-reset_path] |
| <i>-setup</i>         | Only apply to setup checks.   |
| <i>-hold</i>          | Only apply to hold checks.  |

|                           |  |
|---------------------------|--|
| <code>-rise</code>        | Only apply to rising path edges.   |
| <code>-fall</code>        | Only apply to falling path edges.  |
| <code>-reset_path</code>  | Remove any matching <code>set_false_path</code> , <code>set_multicycle_path</code> , <code>set_max_delay</code> , <code>set_min_delay</code> exceptions first. |
| <code>from_list</code>    | A list of clocks, instances, ports or pins.  |
| <code>through_list</code> | A list of instances, pins or nets.   |
| <code>to_list</code>      | 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.

---

|                                     |                                   |
|-------------------------------------|-----------------------------------|
| <b><code>set_fanout_load</code></b> | <i>fanout</i><br><i>port_list</i> |
|-------------------------------------|-----------------------------------|

This command is ignored.

---

|   |                  |
|---|------------------|
| <b><code>set_hierarchy_separator</code></b> | <i>separator</i> |
|---|------------------|

|                  |  |
|------------------|--|
| <i>separator</i> | 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 `'/'`.

---

|                                       |  |
|---------------------------------------|--|
| <b><code>set_ideal_latency</code></b> | <code>[-rise] [-fall] [-min] [-max] delay objects</code> |
|---------------------------------------|--|

The `set_ideal_latency` command is parsed but ignored.

---

|                                       |  |
|---------------------------------------|--|
| <b><code>set_ideal_network</code></b> | <code>[-no_propagation] objects</code> |
|---------------------------------------|--|

The `set_ideal_network` command is parsed but ignored.

---

|                             |  |
|-----------------------------|--|
| <b>set_ideal_transition</b> | <code>[-rise] [-fall] [-min] [-max] <i>transition_time</i></code><br><code><i>objects</i></code> |
|-----------------------------|--|

The `set_ideal_transition` command is parsed but ignored.

---

|  |  |
|--|--|
| <b>set_input_delay</b>                     | <code>[-rise]</code><br><code>[-fall]</code><br><code>[-max]</code><br><code>[-min]</code><br><code>[-clock <i>clock</i>]</code><br><code>[-clock_fall]</code><br><code>[-reference_pin <i>ref_pin</i>]</code><br><code>[-source_latency_included]</code><br><code>[-network_latency_included]</code><br><code>[-add_delay]</code><br><code><i>delay</i></code><br><code><i>port_pin_list</i></code> |
| <br><code>-rise</code>                     | <br>This is the arrival time for the rising edge of the input.   |
| <br><code>-fall</code>                     | <br>This is the arrival time for the falling edge of the input.  |
| <br><code>-max</code>                      | <br>This is the minimum arrival time.  |
| <br><code>-min</code>                      | <br>This is the maximum arrival time.  |
| <br><code>clock</code>                     | <br>The arrival time is from this clock.   |
| <br><code>-clock_fall</code>               | <br>The arrival time is from the falling edge of <i>clock</i>  |
| <br><code>ref_pin</code>                   | <br>The arrival time is with respect to the clock that arrives at <i>ref_pin</i> .   |
| <br><code>-source_latency_included</code>  | <br>If <code>-source_latency_included</code> is not specified the clock source latency (insertion delay) is added to the delay value.  |
| <br><code>-network_latency_included</code> | <br>If <code>-network_latency_included</code> 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`.

---

|                             |   |
|-----------------------------|---|
| <b>set_input_transition</b> | <code>[-rise]</code><br><code>[-fall]</code><br><code>[-max]</code><br><code>[-min]</code><br><i>transition</i><br><i>port_list</i> |
| <code>-rise</code>          | This is the transition time for the rising edge of the input.   |
| <code>-fall</code>          | This is the transition time for the falling edge of the input.  |
| <code>-max</code>           | This is the minimum transition time.  |
| <code>-min</code>           | This is the maximum transition time.  |
| <i>transition</i>           | The transition time (slew).   |
| <i>port_list</i>            | 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.

---

|                    |   |
|--------------------|---|
| <b>set_load</b>    | [-rise]<br>[-fall]<br>[-max]<br>[-min]<br>[-subtract_pin_load]<br>[-pin_load]<br>[-wire_load]<br><i>capacitance</i><br><i>objects</i> |
| -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 <i>capacitance</i> .                                     |
| -pin_load          | <i>capcitanace</i> is external instance pin capacitance (ports only).   |
| -wire_load         | <i>capcitanace</i> is external wire capacitance (ports only).   |
| <i>capacitance</i> | The capacitance, in library capacitance units.  |
| <i>objects</i>     | 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\_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.

---

**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.

---

**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.

---

|                       |   |
|-----------------------|---|
| <b>set_max_delay</b>  | [-rise]<br>[-fall]<br>[-from <i>from_list</i> ]<br>[-rise_from <i>from_list</i> ]<br>[-fall_from <i>from_list</i> ]<br>[-through <i>through_list</i> ]<br>[-rise_through <i>through_list</i> ]<br>[-fall_through <i>through_list</i> ]<br>[-to <i>to_list</i> ]<br>[-rise_to <i>to_list</i> ]<br>[-fall_to <i>to_list</i> ]<br>[-reset_path]<br>[-ignore_clock_latency]<br><i>delay</i> |
| -rise                 | Only constrain paths to rising edges.   |
| -fall                 | Only constrain paths to falling edges.  |
| <i>from_list</i>      | A list of clocks, instances, ports or pins.   |
| <i>through_list</i>   | A list of instances, pins or nets.  |
| <i>to_list</i>        | 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.   |
| <i>delay</i>          | 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.



---

|                       |                                 |
|-----------------------|---------------------------------|
| <b>set_max_fanout</b> | <i>fanout</i><br><i>objects</i> |
|-----------------------|---------------------------------|

*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.

---

|                              |                              |
|------------------------------|------------------------------|
| <b>set_max_leakage_power</b> | <i>power</i> [ <i>unit</i> ] |
|------------------------------|------------------------------|

The `set_max_leakage_power` command is ignored.

---

|                            |                                |
|----------------------------|--------------------------------|
| <b>set_max_time_borrow</b> | <i>delay</i><br><i>objects</i> |
|----------------------------|--------------------------------|

*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.

---

|                           |  |
|---------------------------|--|
| <b>set_max_transition</b> | <i>[-data_path]</i><br><i>[-clock_path]</i><br><i>[-rise]</i><br><i>[-fall]</i><br><i>transition</i><br><i>objects</i> |
|---------------------------|--|

*transition* The maximum transition time (slew).

*objects* List of clocks, ports or designs.

The `set_max_transition` command 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_` input option restricts the maximum transition to clocks data paths. The `-clock_path`, `-data_path`, `-rise` and `-fall` options only apply to clock objects.

---

|                            |                                      |
|----------------------------|--------------------------------------|
| <b>set_min_capacitance</b> | <i>capacitance</i><br><i>objects</i> |
|----------------------------|--------------------------------------|

*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.

---

|                                    |  |
|------------------------------------|--|
| <b>set_min_delay</b>               | <code>[-rise]</code><br><code>[-fall]</code><br><code>[-from from_list]</code><br><code>[-rise_from from_list]</code><br><code>[-fall_from from_list]</code><br><code>[-through through_list]</code><br><code>[-rise_through through_list]</code><br><code>[-fall_through through_list]</code><br><code>[-to to_list]</code><br><code>[-rise_to to_list]</code><br><code>[-fall_to to_list]</code><br><code>[-ignore_clock_latency]</code><br><code>[-reset_path]</code><br><i>delay</i> |
| <code>-rise</code>                 | Only constrain paths to rising edges.  |
| <code>-fall</code>                 | Only constrain paths to falling edges.   |
| <i>from_list</i>                   | A list of clocks, instances, ports or pins.  |
| <i>through_list</i>                | A list of instances, pins or nets.   |
| <i>to_list</i>                     | A list of clocks, instances, ports or pins.  |
| <code>-ignore_clock_latency</code> | Ignore clock latency at the source and target registers.   |
| <code>-reset_path</code>           | Remove any matching <code>set_false_path</code> , <code>set_multicycle_path</code> , <code>set_max_delay</code> , <code>set_min_delay</code> exceptions first.   |
| <i>delay</i>                       | 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.

---

|                            |   |
|----------------------------|---|
| <b>set_min_pulse_width</b> | [-high]<br>[-low]<br><i>min_width</i><br><i>objects</i> |
| -high                      | Set the minimum high pulse width.                       |
| -low                       | Set the minimum low pulse width.                        |
| <i>min_width</i>           |   |
| <i>objects</i>             | List of pins, instances or clocks.                      |

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

---

|                            |   |
|----------------------------|---|
| <b>set_multicycle_path</b> | [-setup]<br>[-hold]<br>[-rise]<br>[-fall]<br>[-start]<br>[-end]<br>[-from <i>from_list</i> ]<br>[-rise_from <i>from_list</i> ]<br>[-fall_from <i>from_list</i> ]<br>[-through <i>through_list</i> ]<br>[-rise_through <i>through_list</i> ]<br>[-fall_through <i>through_list</i> ]<br>[-to <i>to_list</i> ]<br>[-rise_to <i>to_list</i> ]<br>[-fall_to <i>to_list</i> ]<br>[-reset_path]<br><i>path_multiplier</i> |
| -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.   |

|                          |  |
|--------------------------|--|
| <code>-start</code>      | Multiply the source clock period by <i>period_multiplier</i> .   |
| <code>-end</code>        | Multiply the target clock period by <i>period_multiplier</i> .   |
| <code>-reset_path</code> | Remove any matching <i>set_false_path</i> , <i>set_multicycle_path</i> , <i>set_max_delay</i> , <i>set_min_delay</i> exceptions first. |
| <i>from_list</i>         | A list of clocks, instances, ports or pins.  |
| <i>through_list</i>      | A list of instances, pins or nets.   |
| <i>to_list</i>           | A list of clocks, instances, ports or pins.  |
| <i>path_multiplier</i>   | 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.

---

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

|                      |  |
|----------------------|--|
| <i>min_condition</i> | The operating condition to use for min paths and hold checks.  |
| <i>max_condition</i> | The operating condition to use for max paths and setup checks. |
| <i>min_lib</i>       | The name of the library that contains <i>min_condition</i> .   |
| <i>max_lib</i>       | The name of the library that contains <i>max_condition</i> .   |

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

---

|                         |  |
|-------------------------|--|
| <b>set_output_delay</b> | <div> <div>[-rise]</div> <div>[-fall]</div> <div>[-max]</div> <div>[-min]</div> <div>[-clock <i>clock</i>]</div> <div>[-clock_fall]</div> <div>[-reference_pin <i>ref_pin</i>]</div> <div>[-source_latency_included]</div> <div>[-network_latency_included]</div> <div>[-add_delay]</div> <div><i>delay</i></div> <div><i>port_pin_list</i></div> </div> |
| -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.  |
| <i>clock</i>            | The departure time is from this clock.   |
| -clock_fall             | The departure time is from the falling edge of <i>clock</i> .  |
| <i>ref_pin</i>          | The departure time is with respect to the clock that arrives at <i>ref_pin</i> .   |

**-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\_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.

---

|                |  |
|----------------|--|
| <b>set_pvt</b> | <i>instances</i><br>[-min]<br>[-max]<br>[-process <i>process</i> ]<br>[-voltage <i>voltage</i> ]<br>[-temperature <i>temperature</i> ] |
|----------------|--|

|                  |                   |
|------------------|-------------------|
| <i>instances</i> | A list instances. |
|------------------|-------------------|

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

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

|                |                          |
|----------------|--------------------------|
| <i>process</i> | A process value (float). |
|----------------|--------------------------|

|                |                          |
|----------------|--------------------------|
| <i>voltage</i> | A voltage value (float). |
|----------------|--------------------------|

|                    |                              |
|--------------------|------------------------------|
| <i>temperature</i> | 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.

---

|                          |   |
|--------------------------|---|
| <b>set_timing_derate</b> | [-rise]<br>[-fall]<br>[-early]<br>[-late]<br>[-clock]<br>[-data]<br>[-net_delay]<br>[-cell_delay]<br>[-cell_check]<br><i>derate</i><br>[ <i>objects</i> ] |
|--------------------------|---|

|        |                           |
|--------|---------------------------|
| -early | Derate early (min) paths. |
|--------|---------------------------|

|       |                          |
|-------|--------------------------|
| -late | Derate late (max) paths. |
|-------|--------------------------|

|        |                                    |
|--------|------------------------------------|
| -clock | Derate paths in the clock network. |
|--------|------------------------------------|

|       |                    |
|-------|--------------------|
| -data | Derate data paths. |
|-------|--------------------|

|                          |  |
|--------------------------|--|
| <code>-net_delay</code>  | Derate net (interconnect) delays.            |
| <code>-cell_delay</code> | Derate cell delays.                          |
| <code>-cell_check</code> | Derate cell timing check margins.            |
| <code>derate</code>      | The derating factor (0.0 to 1.0).            |
| <code>objects</code>     | 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 `reset_timing_derate` command to remove all derating factors.

---

|                                    |  |
|------------------------------------|--|
| <b><code>set_resistance</code></b> | <code>[-max]</code><br><code>[-min]</code><br><code>resistance</code><br><code>net_list</code> |
| <code>-min</code>                  | The resistance for minimum path delay calculation.   |
| <code>-max</code>                  | The resistance for maximum path delay calculation.   |
| <code>resistance</code>            | The net resistance.  |
| <code>net_list</code>              | A list of nets.  |

---

|                               |   |
|-------------------------------|---|
| <b><code>set_units</code></b> | <code>[-capacitance cap_unit]</code><br><code>[-resistance res_unit]</code><br><code>[-time time_unit]</code><br><code>[-voltage voltage_unit]</code><br><code>[-current current_unit]</code><br><code>[-power power_unit]</code> |
| <code>cap_unit</code>         | The capacitance scale factor followed by 'f'.   |
| <code>res_unit</code>         | The resistance scale factor followed by 'ohm'.  |
| <code>time_unit</code>        | The time scale factor followed by 's'.  |



|                     |   |
|---------------------|---|
| <i>voltage_unit</i> | The voltage scale factor followed by 'v'. |
| <i>current_unit</i> | The current scale factor followed by 'A'. |
| <i>power_unit</i>   | The power scale factor followed by 'w'.   |

The `set_units` command is used to change the units used by the STA command interpreter when parsing commands and reporting results.

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.

|                |   |
|----------------|---|
| <i>library</i> | Library to look for <i>model_name</i> .         |
| -max           | The wire load model is for maximum path delays. |
| -min           | The wire load model is for minimum path delays. |
| <i>objects</i> | Not supported.                                  |

---

```

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

```

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

The `set_wire_load_selection_group` command is parsed but not supported.

---

|                 |   |
|-----------------|---|
| <b>source</b>   | [-echo]<br>[-verbose]<br><i>filename</i><br>> <i>filename</i><br>>> <i>filename</i> ] |
| -echo           | Print each command before evaluating it.  |
| -verbose        | Print each command before evaluating it as well as the result it returns.             |
| <i>filename</i> | 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.

---

|                            |                         |
|----------------------------|-------------------------|
| <b>unset_case_analysis</b> | <i>port_or_pin_list</i> |
|----------------------------|-------------------------|

|                         |                          |
|-------------------------|--------------------------|
| <i>port_or_pin_list</i> | A list of ports or pins. |
|-------------------------|--------------------------|

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

---

|                            |  |
|----------------------------|--|
| <b>unset_clock_latency</b> | <code>[-source]</code><br><i>objects</i> |
|----------------------------|--|

---

|                      |   |
|----------------------|---|
| <code>-source</code> | Specifies source clock latency (clock insertion delay). |
|----------------------|---|

|                |                                  |
|----------------|----------------------------------|
| <i>objects</i> | A list of clocks, pins or ports. |
|----------------|----------------------------------|

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

---

|                               |               |
|-------------------------------|---------------|
| <b>unset_clock_transition</b> | <i>clocks</i> |
|-------------------------------|---------------|

|               |                   |
|---------------|-------------------|
| <i>clocks</i> | A list of clocks. |
|---------------|-------------------|

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

---

|                                |   |
|--------------------------------|---|
| <b>unset_clock_uncertainty</b> | <code>[-from -rise_from -fall_from from_clock]</code><br><code>[-to -rise_to -fall_to to_clock]</code><br><code>[-rise]</code><br><code>[-fall]</code><br><code>[-setup]</code><br><code>[-hold]</code><br><i>objects</i> |
|--------------------------------|---|

*from\_clock*

*to\_clock*

|                    |  |
|--------------------|--|
| <code>-rise</code> | The uncertainty is for the rising edge of the clock. |
|--------------------|--|

|                    |   |
|--------------------|---|
| <code>-fall</code> | The uncertainty is for the falling edge of the clock. |
|--------------------|---|

|                     |  |
|---------------------|--|
| <code>-setup</code> | <i>uncertainty</i> is the setup check uncertainty. |
| <code>-hold</code>  | <i>uncertainty</i> is the hold uncertainty.        |
| <i>uncertainty</i>  | Clock uncertainty.                                 |
| <i>objects</i>      | A list of clocks, ports or pins.                   |

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

---

|                                      |   |
|--------------------------------------|---|
| <b><code>unset_data_check</code></b> | <code>[-from <i>from_object</i>]</code><br><code>[-rise_from <i>from_object</i>]</code><br><code>[-fall_from <i>from_object</i>]</code><br><code>[-to <i>to_object</i>]</code><br><code>[-rise_to <i>to_object</i>]</code><br><code>[-fall_to <i>to_object</i>]</code><br><code>[-setup]</code><br><code>[-hold]</code><br><code>[-clock <i>clock</i>]</code> |
| <i>from_object</i>                   | A pin used as the timing check reference.   |
| <i>to_object</i>                     | A pin that the setup/hold check is applied to.  |
| <code>-setup</code>                  | Add a setup timing check.   |
| <code>-hold</code>                   | Add a hold timing check.  |
| <i>clock</i>                         | 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_gating`**  
**`ating`**

|                |  |
|----------------|--|
| <i>objects</i> | 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.

---

|                             |  |
|-----------------------------|--|
| <b>unset_disable_timing</b> | [-from <i>from_port</i> ]<br>[-to <i>to_port</i> ]<br><i>objects</i> |
| <i>from_port</i>            |  |
| <i>to_port</i>              |  |
| <i>objects</i>              | 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.

---

|                          |   |
|--------------------------|---|
| <b>unset_input_delay</b> | [-rise]<br>[-fall]<br>[-max]<br>[-min]<br>[-clock <i>clock</i> ]<br>[-clock_fall]<br><i>port_pin_list</i> |
| -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.   |
| <i>clock</i>             | The arrival time is from this clock.  |
| -clock_fall              | The arrival time is from the falling edge of <i>clock</i>   |
| <i>pin_port_list</i>     | A list of pins or ports.  |

The `unset_input_delay` command removes a previously defined `set_input_delay`.

---

|                           |   |
|---------------------------|---|
| <b>unset_output_delay</b> | [-rise]<br>[-fall]<br>[-max]<br>[-min]<br>[-clock <i>clock</i> ]<br>[-clock_fall]<br><i>port_pin_list</i> |
| -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.   |
| <i>clock</i>              | The arrival time is from this clock.  |
| -clock_fall               | The arrival time is from the falling edge of <i>clock</i>   |
| <i>pin_port_list</i>      | A list of pins or ports.  |

The unset\_output\_delay command a previously defined set\_output\_delay.

---

|                              |  |
|------------------------------|--|
| <b>unset_path_exceptions</b> | [-setup]<br>[-hold]<br>[-rise]<br>[-fall]<br>[-from <i>from_list</i> ]<br>[-rise_from <i>from_list</i> ]<br>[-fall_from <i>from_list</i> ]<br>[-through <i>through_list</i> ]<br>[-rise_through <i>through_list</i> ]<br>[-fall_through <i>through_list</i> ]<br>[-to <i>to_list</i> ]<br>[-rise_to <i>to_list</i> ]<br>[-fall_to <i>to_list</i> ]<br> |
| -setup                       | Only apply to setup checks.  |

|                           |   |
|---------------------------|---|
| <code>-hold</code>        | Only apply to hold checks.                  |
| <code>-rise</code>        | Only apply to rising path edges.            |
| <code>-fall</code>        | Only apply to falling path edges.           |
| <code>from_list</code>    | A list of clocks, instances, ports or pins. |
| <code>through_list</code> | A list of instances, pins or nets.          |
| <code>to_list</code>      | 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.

---

**update\_timing**      `[-full]`

`-full`      Update all arrivals from scratch (non-incrementally).

The `update_timing` command updates the arrival times for all pins in the design.

---

**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.

---

|                            |   |
|----------------------------|---|
| <b>write_sdc</b>           | <code>[-no_timestamp]</code><br><code>[-digits <i>digits</i>]</code><br><i>filename</i> |
| <code>-no_timestamp</code> | Do not include a time and date in the SDC file.   |
| <i>digits</i>              | The number of digits after the decimal point to report. The default is 4.               |
| <i>filename</i>            | The name of the file to write the constraints to.                                       |

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

---

|                            |   |
|----------------------------|---|
| <b>write_sdf</b>           | <code>[-corner <i>corner_name</i>]</code><br><code>[-divider / .]</code><br><code>[-digits <i>digits</i>]</code><br><code>[-gzip]</code><br><code>[-no_timestamp]</code><br><code>[-no_version]</code><br><i>filename</i> |
| <i>corner_name</i>         | Process corner delays to write.   |
| <code>-divider</code>      | Divider to use between hierarchy levels in pin and instance names. Must be '/' or '.'.  |
| <i>digits</i>              | The number of digits after the decimal point to report. The default is 4.   |
| <code>-gzip</code>         | Write a gzip compressed file.   |
| <code>-no_timestamp</code> | Do not write a DATE statement.  |
| <code>-no_version</code>   | Do not write a VERSION statement.   |
| <i>filename</i>            | The name of the file to write the constraints to.   |
| <i>digits</i>              | 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.



---

|                         |   |
|-------------------------|---|
| <b>write_path_spice</b> | -path_args <i>path_args</i><br>-spice_directory <i>spice_directory</i><br>-lib_subckt_file <i>lib_subckts_file</i><br>-model_file <i>model_file</i><br>-power <i>power</i><br>-ground <i>ground</i> |
| <i>path_args</i>        | -from -through -to arguments as in report_checks.   |
| <i>spice_directory</i>  | Spice output directory.   |
| <i>lib_subckts_file</i> | Cell transistor level subckts.  |
| <i>model_file</i>       | Transistor model definitions .included by <i>spice_file</i> .   |
| <i>power</i>            | Voltage supply name in voltage_map of the default liberty library.  |
| <i>ground</i>           | 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.

## Variables

---

|                            |                |
|----------------------------|----------------|
| <b>hierarchy_separator</b> | Any character. |
|----------------------------|----------------|

---

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

---

|                              |       |
|------------------------------|-------|
| <b>link_make_black_boxes</b> | 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.

---

|   |       |
|---|-------|
| <b>power_default_signal_toggle_rate</b> | float |
|---|-------|

---

The toggle rate used to find switching power for non-clock signal. The default value is 0.1.

---

|                                       |       |
|---------------------------------------|-------|
| <b>sta_bidirect_net_paths_enabled</b> | 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.

---

|                              |       |
|------------------------------|-------|
| <b>sta_continue_on_error</b> | 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 1.

---

|                      |                            |
|----------------------|----------------------------|
| <b>sta_crpr_mode</b> | 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.

---

|                                      |       |
|--------------------------------------|-------|
| <b>sta_cond_default_arcs_enabled</b> | 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.

---

|                         |       |
|-------------------------|-------|
| <b>sta_crpr_enabled</b> | 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 Pessimism Removal, or “CRPR”. Enabling CRPR slows down the analysis. The default value is 1.

---

|                                  |       |
|----------------------------------|-------|
| <b>sta_dynamic_loop_breaking</b> | 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.

---

|                                       |       |
|---------------------------------------|-------|
| <b>sta_gated_clock_checks_enabled</b> | 0   1 |
|---------------------------------------|-------|

When sta\_gated\_clock\_checks\_enabled is 1, clock gating setup and hold timing checks are checked. The default value is 1.

---

|                                     |       |
|-------------------------------------|-------|
| <b>sta_input_port_default_clock</b> | 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.

---

|  |       |
|--|-------|
| <b><code>sta_internal_bidirect_instance_paths_enabled</code></b> | 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.

---

|  |       |
|--|-------|
| <b><code>sta_propagate_all_clocks</code></b> | 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.

---

|  |       |
|--|-------|
| <b><code>sta_propagate_gated_clock_enable</code></b> | 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.

---

|   |       |
|---|-------|
| <b><code>sta_recovery_removal_checks_enabled</code></b> | 0   1 |
|---|-------|

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

---

|   |         |
|---|---------|
| <b><code>sta_report_default_digits</code></b> | integer |
|---|---------|

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

---

|   |       |
|---|-------|
| <b><code>sta_preset_clear_arcs_enabled</code></b> | 0   1 |
|---|-------|

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

Version 2.0.4, Jan 5, 2019

Copyright (c) 2019, Parallax Software, Inc.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<https://www.gnu.org/licenses/>>.