

You are here: Synthesis Man Pages > **Synthesis Tool Commands** > **c** > create_clock

create_clock

NAME

SYNTAX

ARGUMENTS

DESCRIPTION

EXAMPLES

SEE ALSO

NAME

create_clock

Creates a clock object and defines its waveform in the current design.

SYNTAX

```
status create_clock
    [-name clock_name]
    [-add]
    [source_objects]
    [-period period_value]
    [-waveform edge_list]
    [-comment comment_string]
```

Data Types

<i>clock_name</i>	string
<i>source_objects</i>	list
<i>period_value</i>	float
<i>edge_list</i>	list
<i>comment_string</i>	string

ARGUMENTS

-name *clock_name*

Specifies the name of the clock being created. If you do not use this option, the clock is given the same name as the first clock source specified in *source_objects*. If you do not use *source_objects*, you must use this option, which creates a virtual clock not associated with a port or pin. Use this option along with *source_objects* to give the clock a more descriptive name than that of the pin or port where it is applied.

If you specify the **-add** option, you must use the **-name** option and the clocks with the same source must have different names.

-add

Specifies whether to add this clock to the existing clock or to overwrite the existing clock. Use this option to capture the case where multiple clocks must be specified on the same source for simultaneous analysis with different clock waveforms. When you specify this option, you must also use the **-name** option. Defining multiple clocks on the same source pin or port causes longer runtime and higher memory usage than a single clock, because the synthesis timing engine must explore all possible combinations of launch and capture clocks. Use the **set_false_path** command to disable unwanted clock combinations. This option is ignored (the default), unless multiple clocks analysis is enabled by setting the **timing_enable_multiple_clocks_per_reg** variable to **true**.

source_objects

Specifies a list of pins or ports on which to apply this clock. If you do not use this option, you must use **-name** *clock_name*, which creates a virtual clock not associated with a port or pin. If you specify a clock on a pin that already has a clock, the new clock replaces the old clock unless you use the **-add** option.

-period *period_value*

Specifies the period of the clock waveform in library time units.

-waveform *edge_list*

Specifies the rise and fall edge times, in library time units, of the clock over an entire clock period. The first time in the list is a rising transition, typically the first rising transition after time zero. There must be an even number of increasing times, and they are assumed to be alternating rise and fall times. The numbers must represent one full clock period. If **-waveform** *edge_list* is not specified, but **-period** *period_value* is, a default waveform with a rise edge of 0.0 and a fall edge of *period_value*/2 is assumed.

-comment *comment_string*

Allows the command to accept a comment string. The tool honors the annotation and preserves it with the SDC object so that the exact string is written out when the constraint is written out when you use the **write_sdc** or **write_script** command. The comment remains intact through the synthesis, place-and-route, and timing-analysis flows.

DESCRIPTION

The **create_clock** command creates a clock object in the current design. The command defines the specified *source_objects* as clock sources in the current design. A pin or port can be a source for a single clock. If *source_objects* is not specified, but a *clock_name* is given, a virtual clock is created. A virtual clock can be created to represent an off-chip clock for input or output delay specification. For more information about input and output delay, refer to the **set_input_delay** and **set_output_delay** command man pages.

Clock objects hold attributes that affect the clock network, such as **dont_touch_network**, **fix_hold**, and **propagated_clock**. Using **create_clock** on an existing clock object overwrites the attributes previously set on the clock object. The **create_clock** command also defines the waveform for the clock. The clock can have multiple pulses per period. Setup and hold path delays are automatically derived from the clock waveforms of the path startpoint and endpoint. The **fix_hold** attribute (set by the **set_fix_hold** command) directs **compile** to fix hold violations for a clock.

By default, a new path group is created for the clock. This groups together the endpoints related to this clock for cost function calculation. To remove the clock from its assigned group, use the **group_path** command to reassign the clock to another group or to the default path group. For more information, refer to the **group_path** command man page.

The new clock has ideal timing, so no propagation delay through the clock network is assumed. To enable propagation delay through the clock network, use the **set_propagated_clock** command. To add skew or uncertainty to the ideal waveform, use the **set_clock_latency** or **set_clock_uncertainty** command.

To show information about all clock sources in a design, use the **report_clock** command. To get a list of clock sources, use the **get_clocks** command. To return sequential cells related to a given clock, use the **all_registers** command. To undo **create_clock**, use the **remove_clock** command.

Multicorner-Multimode Support

This command uses information from the current scenario only.

EXAMPLES

The following example creates a clock on port *PHI1* with a period of 10.0, rise at 5.0, and fall at 9.5:

```
prompt> create_clock "PHI1" -period 10 -waveform {5.0 9.5}
```

In the following example, the clock has a falling edge at 5 and a rising edge at 10, with a period of 10. Because the **-waveform** option expects the edges to be ordered first rise then fall, and to increase in value,

the fall edge can be given as 15; that is, the next falling edge after the first rise edge at 10.

```
prompt> create_clock "PHI2" -period 10 -waveform {10 15}
```

The following example creates a clock named *CLK* on pin *u13/Z*, with a period of 25, fall at 0.0, rise at 5.0, fall at 10.0, rise at 15.0, and so forth:

```
prompt> create_clock "u13/Z" -name "CLK" -period 25 -waveform {5 10 15 25}
```

The following example creates a virtual clock named *HI2* with a period of 10.0, rise at 0.0, and fall at 5.0:

```
prompt> create_clock -name "PHI2" -period 10 -waveform {0.0 5.0}
```

The following example creates a clock with multiple sources and a complex waveform:

```
prompt> create_clock -name "clk2" -period 10 -waveform {0.0 2.0 4.0 6.0} \
{clkgen1/Z clkgen2/Z clkgen3/Z}
```

SEE ALSO

```
all_clocks(2)
all_registers(2)
check_timing(2)
compile(2)
current_design(2)
get_clocks(2)
group_path(2)
remove_clock(2)
report_clock(2)
reset_design(2)
set_clock_latency(2)
set_clock_uncertainty(2)
set_dont_touch_network(2)
set_fix_hold(2)
set_max_delay(2)
set_min_delay(2)
set_output_delay(2)
set_propagated_clock(2)
```