Quick Reference Guide for the Design Constraints Description Language (DCDL)

Sponsor

Accellera

Last modified: 7/25/00

Copyright © 1999-2000 Accellera. All rights reserved.

This document is not part of the proposed standard, and it is subject to change. Permission is hereby granted for standards committee participants to reproduce this document for purposes of standardization activities. Use of information contained in this draft is at your own risk.
This document is automatically generated using the DCDL specification as input. The DCDL specification should be referenced for details about each command. Version 0.3.7 of this guide matches version 0.3.7 of the specification.
1. Introduction

This document provides a quick method to look up command descriptions and syntax. For details about each command, the DCDL specification should be referenced.

The commands are listed in this document based on the constraint domain for which they belong.

This guide is a “living” document that changes based on changes to the specification. The status of each command is indicated by a symbol below the command name:

- ● represents a reviewed and approved command. This command description is stable.
- ◥ represents a command that is currently being reviewed and further refined. This command description could change.
- ◡ represents the initial draft description of a command. This command description is very likely to change.
2. Clock Commands

clock

The clock command associates a waveform with actual design pins or ports.

Usage

clock

-waveform waveform_identifier (-pins pin_list | -ports port_list) [-parent_pin pin_identifier | -parent_port port_identifier]

clock_arrival_time

The clock_arrival_time command defines a window of time in which clock signals will arrive at pins and ports with respect to a specified reference point (waveform).

Usage

clock_arrival_time

-waveform waveform_identifier [-lead | -trail] [-early | -late] -ports port_list | -pins pin_list clock_arrival_time_value_list

clock_delay

The clock_delay command specifies the delay characteristics of a clock network or a portion of a hierarchical clock network.

Usage

clock_delay

-waveform waveform_identifier | (-root_port port_identifier | -root_pin pin_identifier) | (-leaf pin_identifier) [-rise | -fall] [-early | -late] delay_unsigned_time_value_list

clock_mode

The clock_mode command specifies the default analysis for clock network delays.

Usage

clock_mode

[-root_port port_list | -root_pin pin_list] -ideal | -actual

clock_required_time

The clock_required_time command defines a window of time in which clock signals are insured to arrive at pins and ports with respect to a specified reference point (waveform).

Usage

clock_required_time

-waveform waveform_identifier [-lead | -trail] [-early | -late] -ports port_list | -pins pin_list clock_required_time_value_list

clock_skew

The clock_skew command specifies skew characteristics of a clock network (or network portion).

Usage

clock_skew

(-root_port port_identifier | -root_pin pin_identifier) [-rise | -fall] [-early | -late] skew_unsigned_time_value_list
clock_uncertainty

The clock_uncertainty command specifies the worst-case uncertainty between two clock distribution networks.

Usage

```
clock_uncertainty
  [-from root_waveform_identifier] [-to target_waveform_identifier]
  [-from_edge rise | fall]
  [-to_edge rise | fall] [-early | -late]
  [-absolute | -increment]
  [-ideal | -actual] { uncertainty_value }
```

derived_waveform

The derived_waveform command specifies a new waveform, derived from an existing waveform.

Usage

```
derived_waveform
  -waveform parent_waveform_identifier
  -name derived_waveform_identifier
  [-inverted]
  [-phase { offset_shift_rsvalue_list }]
  ( [-multiplier mult_unsigned_number]
    [-divisor divisor_unsigned_number ] )
  [-derived_edges]
    { lead_edge_unsigned_number
      trail_edge_unsigned_number } ]
  [-lead_jitter jitter_value | -trail_jitter jitter_value]
```

common_insertion_delay

The common_insertion_delay command specifies the portion of the external insertion delay that is common to two clock roots.

Usage

```
common_insertion_delay
  (-from_port clock_port_identifier | -from_pin clock_pin_identifier )
  ( -to_port clock_port_identifier | -to_pin clock_pin_identifier ) [ -rise | -fall ]
  [-early | -late] insertion_rsvalue_list
```

target_uncertainty

The target_uncertainty command specifies the worst-case uncertainty between the clock edge at a target register and the clock edges at any source register.

Usage

```
target_uncertainty
  -port clock_root_identifier | ( -pin clock_leaf_identifier |
    clock_root_identifier ) | -instance
    instance_identifier [-early | -late]
  [-absolute | -increment]
  [-ideal | -actual] uncertainty_value
```
waveform

The `waveform` command specifies an abstract, ideal waveform that can be used in other DCDL commands as a reference.

**Usage**

`waveform
-waveform name waveform_identifier
[ -period period_rvalue ]
[ -edges { lead_rsvalue trail_rsvalue } ]
[ -lead_jitter { left_unsigned_time_value right_unsigned_time_value } | { offset_unsigned_time_value } ]
[ -trail_jitter { left_unsigned_time_value right_unsigned_time_value } | { offset_unsigned_time_value } ]
[ -inverted ] [ -domain domain_identifier ]`
3. Timing Boundary Commands

data_arrival_time

The data_arrival_time command specifies when transitions on data signals arrive at input or bi-directional ports or pins with respect to a specified reference point.

Usage

data_arrival_time

-w waveform waveform_identifier [ -target | -source ] [ -lead | -trail ] [ -early | -late ]
[ -rise | -fall ]
-ports port_list | -pins pin_list
arrival_time_value_list

data_required_time

The data_required_time command specifies the time required for output or bi-directional ports or pins to be stable with respect to a specified reference point.

Usage

data_required_time

-w waveform waveform_identifier [ -target | -source ] [ -lead | -trail ] [ -early | -late ]
-ports port_list | -pins pin_list
required_time_value_list

departure_time

The departure_time command specifies a partial path delay time range beyond a pin or port (not including interconnect and loading due to the external net) for the timing reserved for the remaining path external to the block.

Usage

departure_time

-w waveform waveform_identifier [ -early | -late ]
[ -rise | -fall ]
-ports port_list | -pins pin_list
departure_time_value_list

external_delay

The external_delay command specifies purely combinational delays that are external to the design.

Usage

external_delay

-w waveform waveform_identifier [ -early | -late ]
-ports port_list | -pins pin_list
-rise_range rise_time_value_list
-fall_range fall_time_value_list

slew_limit

The slew_limit command specifies the maximum slew time allowed for input and output pins or ports.

Usage

slew_limit

[ -early | -late ] [ -rise | -fall ]
-ports port_list | -pins pin_list
d slew_limit_time_value

slew_time

The slew_time command specifies the ramp time required for a signal to cross two threshold points for a pin or port. Clock slew can be specified with this command.

Usage

slew_time

[ -early | -late ] [ -rise | -fall ]
-ports port_list | -pins pin_list
d slew_time_value

slew_limit

The slew_limit command specifies the maximum slew time allowed for input and output pins or ports.

Usage

slew_limit

[ -early | -late ] [ -rise | -fall ]
-ports port_list | -pins pin_list
d slew_limit_time_value

4. Timing Exception Commands

borrow_limit

The borrow_limit command specifies the maximum amount of time that can be borrowed from a cycle for level-sensitive latches.

Usage

borrow_limit

false_path

The false_path command identifies timing paths that should not be analyzed.

Usage

false_path

[ -early | -late ] [ -rise | -fall ] path_options

path_options ::= ( -from_port | -from_pin | -from_instance | -from_waveform { object_identifier } ) | ( -from_port | -from_pin | -from_instance | -from_waveform { object_identifier } ) | ( -through_port | -through_pin | -through_instance | -through_net { object_identifier } )

disable

The disable command disables timing arcs in a library cell and all instances of that cell or a particular instance.

Usage

disable

disable

[ -library library_identifier ]
- cell cell_identifier
- instance instance_identifier
[ -from_port port_list | -to_pin pin_list ]
[ -to_port port_list | -to_pin pin_list ]
[ -output_arcs ]
[ -internal_arcs ]
[ -input_arcs ]
[ -internal_arcs ]
multi_cycle_path

The `multi_cycle_path` command identifies timing paths that span over multiple clock cycles.

**Usage**

multi_cycle_path

[ -target | -source ]
- waveform waveform_identifier ]
[ -early | -late ] [ -rise | -fall ]
path_options { cycle_number }

path_options ::= ( -from_port | -from_pin |
- from_instance | -from_waveform
{ object_identifier } ) ] |
( -from_port | -from_pin |
- from_instance | -from_waveform
{ object_identifier } ) ] |
( { -through_port | -through_pin |
- through_instance | -through_net
{ object_identifier } } ) )

cycle_number ::= [ sign ] real | [ sign ]
unsigned_number |
{ [ sign ] real [ sign ] real } |
{ [ sign ] unsigned_number [ sign ]
unsigned_number } |
{ [ sign ] real [ sign ] unsigned_number } |
{ [ sign ] unsigned_number [ sign ] real } |
placeholder

tree_delay

The `tree_delay` command constrains the timing characteristics of a general buffer tree.

**Usage**

tree_delay

- root_port port_identifier | -root_pin
pin_identifier
[ -ideal | -actual ] [ -explicit_leaf pin_list ]
[ -default_insertion delay_rvalue ]
[ -explicit_insertion delay_rvalue ]
[ -internal_insertion delay_rvalue ]
[ -default_skew skew_rvalue ]
[ -default_transition time_rvalue ]
[ -explicit_transition time_rvalue ]
5. Operating Condition Commands

**operating_point**

The *operating_point* command provides a means to set the process, temperature, and voltage operating point for a design all at once (as opposed to using the separate commands available).

**Usage**

**operating_point**

[-voltage_regime voltage_regime_identifier]
 [-temperature_regime temperature_regime_identifier]
 [-library library_identifier]
 -name operating_point_identifier

**operating_process**

The *operating_process* command specifies the process condition that should be applied to the design. Several specification methods are available.

**Usage**

**operating_process**

[-library library_identifier]
 [-value operating_point_rvalue]

**operating_range**

The *operating_range* command provides a method to specify a default range of operating conditions for a design (or design portion) through the use of an operating name specified in a technology library.

**Usage**

**operating_range**

[-library library_identifier]
 operating_range_identifier

**operating_temperature**

The *operating_temperature* command specifies the temperature value that should be applied to the design. Several specification methods are available.

**Usage**

**operating_temperature**

([-temperature_regime temperature_regime_identifier] | [-instances instance_list] [-pin pin_identifier])
operating_voltage
●

The operating_voltage command specifies the voltage value that should be applied to the design. Several specification methods are available.

Usage

operating_voltage

( [-voltage_regime voltage_regime_identifier ] |
  [ -instances instance_list |
  [ -pin pin_identifier ] ) |
  [ -library library_identifier ] |
  -value operating_point_rvalue ] ( -best |
  -nominal | -worst |
  -min_best | -typ_best | -max_best |
  -min_worst | -typ_worst | -max_worst )

voltage_regime
●

The voltage_regime command provides a method to specify a portion of a design within which voltage variations are assumed to be correlated.

Usage

voltage_regime

[ -logical_rail logical_rail_identifier ] |
  [ -physical_rail physical_rail_identifier ] |
  [ -base_voltage voltage_rvalue ] |
  [ -min_voltage minimum_rvalue ] |
  [ -max_voltage maximum_rvalue ] |
  [ ( -cells cell_list |
  [ -port port_identifier ] ) ] |
  [ ( -instances instance_list |
  [ -pin pin_identifier ] ) ] |
  voltage_regime_identifier

temperature_regime
●

The temperature_regime command provides a method to specify a portion of a design within which temperature variations are assumed to be correlated.

Usage

temperature_regime

[ -cells cell_list ] | [ -instances instance_list ]
  temperature_regime_identifier
6. Universal Commands

**constant**

The `constant` command specifies a continuous value for an input, output, or inout pin or port.

**Usage**

```
constant
( -ports port_list | -pins pin_list ) 0 | 1
```

**design_name_space**

The `design_name_space` command either specifies a predefined name space for design objects or a custom name space.

**Usage**

```
design_name_space
-verifylog ( "1995" | "2000" ) |
   -vhdl ( "1987" | "1993" | "2000" ) |
   -edef ( "200" | "300" | "400" ) |
-custom ( -characters " [ character_set ] " |
   " [ character_range ] " |
   " [ character_set character_range ] " |
   " [ character_range character_set ] " ) |
   -case_sensitive | -caseInsensitive |
   -characterEscape " escape_character " |
   -stringEscapeStart " escape_character " |
   -stringEscapeEnd " escape_character " |
) |
   -escapeType include | exclude |
   -busRangeSeparatorUp " index_character " | " index_identifier " |
   -busRangeSeparatorDown " index_character " | " index_identifier " |
   -busBitLeft " bit_character " |
   -busBitRight " bit_character " |
   -hierarchyDelimiter " delimiter_character " |
```

**extend_dcdl**

The `extend_dcdl` command provides a method to call non-standard DCDL commands.

**Usage**

```
extend_dcdl
command_identifier [ -arguments " argument_text " ]
```

**functional_mode**

The `functional_mode` command selects the state-dependent effects (or mode) for analysis of instances.

**Usage**

```
functional_mode
( [ -groupName group_identifier ] |
   -modeName mode_identifier ) |
   ( -all | -default ) instance_list
```

**history**

The `history` command provides a placeholder for comments about the lineage of the DCDL file.

**Usage**

```
history
" history_text "
```

**include**

The `include` command inserts DCDL commands from another file.

**Usage**

```
include
[ -inline ] " pathname_identifier "
```
units

The *units* command specifies a quantity in terms of a multiplier for time, capacitance, resistance, voltage, and temperature values.

**Usage**

units

[ -time multiplier ] [ -capacitance multiplier ]
[ -resistance multiplier ]
[ -voltage multiplier ]
[ -temperature multiplier ]
[ -inductance multiplier ]

version

The *version* command identifies the DCDL specification version to which the commands that follow reference.

**Usage**

version

version_identifier
7. Scoping Commands

current_scope

The current_scope command establishes the design level where all the referenced design objects can be found by subsequent commands.

Usage

```plaintext
current_scope

-instance instance_identifier | -cell cell_identifier | -top |
-up level_unsigned_number
```
8. Parasitic Boundary Commands

**driver_cell**

The *driver_cell* command provides a method to describe the characteristics of the driver cell that is driving the external net that connects to an input or bi-directional port of the design.

Usage

```
driver_cell
[ -library library_identifier ] -cell cell_identifier
[ -instance instance_identifier ]
[ -to port_identifier ]
( ( [ -from port_identifier ] [ -rise slew slew_rvalue ] [ -fall slew slew_rvalue ] )
( [ -multiplier multiplier ] [ -parallel driver unsigned_number ] ) [ -rise | -fall ]
[ -early | -late ]
-ports port_list
```

**driver_resistance**

The *driver_resistance* command specifies a resistance value for the cell connected to an external net that is connected to an input or bi-directional port on the design.

Usage

```
driver_resistance
[ -early | -late ] [ -rise | -fall ]
-ports port_list resistance_rvalue
```

**external_sinks**

The *external_sinks* command specifies the number of external sinks connected to ports.

Usage

```
external_sinks
-ports port_list sinks unsigned_number
```

**external_sources**

The *external_sources* command specifies the number of external sources connected to ports.

Usage

```
external_sources
-port port_list sources unsigned_number
```

**fanout_load**

The *fanout_load* command specifies the number of loads on design ports.

Usage

```
fanout_load
-ports port_list load unsigned_number
```

**fanout_load_limit**

The *fanout_load_limit* command specifies the maximum fanout load allowed on design ports.

Usage

```
fanout_limit
-ports port_list load_limit unsigned_number
```

**port_capacitance**

The *port_capacitance* command specifies the capacitance external to a port in a design, based on input and output loading from other pins and nets connected to the port.

Usage

```
port_capacitance
[ -early | -late ] [ -typ ] [ -pin_load | -wire_load | -lumped_load ]
-ports port_list

capacitance_rvalue_list
```
**port_capacitance_limit**

The `port_capacitance_limit` command specifies the maximum capacitance value from a source external to the design port.

Usage

```
port_capacitance_limit
-ports port_list load_limit_rvalue
```

**port_wire_load**

The `port_wire_load` command specifies a wire load model for a specified port.

Usage

```
port_wire_load
[-library library_identifier] -ports port_list
wire_load_model_identifier
```

**wire_load_model**

The `wire_load_model` command specifies which wire load model should be applied from a library.

Usage

```
wire_load_model
[-library library_identifier]
[-instances instance_list]
wire_load_model_identifier
```
borrow_limit 9

clock 5
clock_arrival_time 5
clock_delay 5
clock_mode 5
clock_required_time 5
clock_skew 5
clock_uncertainty 6

Commands
borrow_limit 9
clock 5
clock_arrival_time 5
clock_delay 5
clock_mode 5
clock_required_time 5
clock_skew 5
clock_uncertainty 6
common_insertion_delay 6
constant 13
current_scope 15
data_arrival_time 8
data_required_time 8
departure_time 8
derived_waveform 6
design_name_space 13
disable 9
driver_cell 16
driver_resistance 16
extend_dcdl 13

Extending DCDL 13
external_delay 8
external_sinks 16
external_sources 16

false_path 9
fanout_load 16
fanout_load_limit 16
functional_mode 13

history 13
include 13
multi_cycle_path 10
operating_point 11
operating_process 11
operating_range 11
operating_temperature 11
operating_voltage 12
port_capacitance 16
port_capacitance_limit 17
port_wire_load 17
slew_limit 8
slew_time 8
target_uncertainty 6
temperature_regime 12
tree_delay 10
tree_mode 10
units 14
version 14
voltage_regime 12
waveform 7
wire_load_model 17
common_insertion_delay 6
constant 13
current_scope 15

Introduction 3
M

Modes 13
multi_cycle_path 10

O

operating_point 11
operating_process 11
operating_range 11
operating_temperature 11
operating_voltage 12

P

port_capacitance 16
port_capacitance_limit 17
port_wire_load 17

S

slew_limit 8
slew_time 8

T

target_uncertainty 6
temperature_regime 12
tree_delay 10
tree_mode 10

U

units 14

V

version 14
voltage_regime 12

W

waveform 7
wire_load_model 17