Constraint Translator for Ambit® BuildGates® Synthesis and Cadence® PKS

Product Version 4.0.8
May 2001

© 2000 Cadence Design Systems, Inc. All rights reserved.
Printed in the United States of America.

Cadence Design Systems, Inc., 555 River Oaks Parkway, San Jose, CA 95134, USA

Trademarks: Trademarks and service marks of Cadence Design Systems, Inc. (Cadence) contained in this document are attributed to Cadence with the appropriate symbol. For queries regarding Cadence's trademarks, contact the corporate legal department at the address shown above or call 1-800-862-4522.

All other trademarks are the property of their respective holders.

Restricted Print Permission: This publication is protected by copyright and any unauthorized use of this publication may violate copyright, trademark, and other laws. Except as specified in this permission statement, this publication may not be copied, reproduced, modified, published, uploaded, posted, transmitted, or distributed in any way, without prior written permission from Cadence. This statement grants you permission to print one (1) hard copy of this publication subject to the following conditions:

1. The publication may be used solely for personal, informational, and noncommercial purposes;
2. The publication may not be modified in any way;
3. Any copy of the publication or portion thereof must include all original copyright, trademark, and other proprietary notices and this permission statement; and
4. Cadence reserves the right to revoke this authorization at any time, and any such use shall be discontinued immediately upon written notice from Cadence.

Disclaimer: Information in this publication is subject to change without notice and does not represent a commitment on the part of Cadence. The information contained herein is the proprietary and confidential information of Cadence or its licensors, and is supplied subject to, and may be used only by Cadence's customer in accordance with, a written agreement between Cadence and its customer. Except as may be explicitly set forth in such agreement, Cadence does not make, and expressly disclaims, any representations or warranties as to the completeness, accuracy or usefulness of the information contained in this document. Cadence does not warrant that use of such information will not infringe any third party rights, nor does Cadence assume any liability for damages or costs of any kind that may result from use of such information.

Restricted Rights: Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor.

Contents

Preface 8
About This Manual 8
Other Information Sources 8
Syntax Conventions 9
Text Command Syntax 9
About the Graphical User Interface 10
Using Menus 10
Using Forms 11

1
Getting Started 12
Setting Up the BuildGates Synthesis Constraint Translator 14
Constraint Modeling Guidelines 14
Supported Constructs 15
Unsupported Constructs 17
Translating Constraint Files 18
Globals 20

2
Translating Constraints 22
all_clocks 25
Mapping 25
Messages 25
all_inputs 26
Mapping 26
Messages 26
all_outputs 27
Mapping 27
Messages 27

create_clock 28
Mapping 28
Messages 29
current_design 30
Mapping 30
Messages 30
current_instance 31
Mapping 31
Messages 31
find 32
Mapping 32
Messages 33
get_cells 34
Mapping 34
Messages 34
get_clocks 36
Mapping 36
Messages 36
get_lib_cells 37
Mapping 37
Messages 37
get_libs 39
Mapping 39
Messages 39
get_nets 41
Mapping 41
Messages 41
get_pins 43
Mapping 43
Messages 43
get_ports 45
Mapping 45
Messages 45
set_capacitance 47
set_case_analysis 48
Mapping 48

Messages . . . 48
set_clock_latency . . . 49
Mapping . . . 49
Messages . . . 50
set_clock_skew . . . 51
Mapping . . . 51
Messages . . . 54
set_clock_transition . . . 55
Mapping . . . 55
Messages . . . 56
set_clock_uncertainty . . . 57
Mapping . . . 57
Messages . . . 58
set_disable_timing . . . 60
Mapping . . . 60
Messages . . . 61
set_dont_touch . . . 63
Mapping . . . 63
Messages . . . 64
set_dont_touch_network . . . 65
Mapping . . . 65
Messages . . . 65
set_drive . . . 67
Mapping . . . 67
Messages . . . 67
set_driving_cell . . . 68
Mapping . . . 68
Messages . . . 69
set_false_path . . . 71
Mapping . . . 71
Messages . . . 72
set_fanout_load . . . 74
Mapping . . . 74
Messages . . . 74
set_input_delay . . . 75
Mapping . . . 75

Messages . . . 76
set_input_transition . . . 78
Mapping . . . 78
Messages . . . 78
set_load . . . 79
Mapping . . . 79
Messages . . . 80
set_logic_one . . . 81
Mapping . . . 81
Messages . . . 81
set_logic_zero . . . 82
Mapping . . . 82
Messages . . . 82
set_max_capacitance . . . 83
Mapping . . . 83
Messages . . . 84
set_max_delay . . . 85
Mapping . . . 85
Messages . . . 85
set_max_fanout . . . 86
Mapping . . . 86
Messages . . . 87
set_max_transition . . . 88
Mapping . . . 88
Messages . . . 89
set_min_capacitance . . . 90
Mapping . . . 90
Messages . . . 91
set_min_delay . . . 92
Mapping . . . 92
Messages . . . 92
set_min_transition . . . 93
Mapping . . . 93
Messages . . . 94
set_multicycle_path . . . 95
Mapping . . . 95

Messages 97
set_operating_conditions 99
Mapping 99
Messages 99
set_output_delay 101
Mapping 101
Messages 102
set_port_fanout_number 104
Mapping 104
Messages 104
set_propagated_clock 105
Mapping 105
Messages 105
set_resistance 106
Mapping 106
Messages 106
set_wire_load 107
Mapping 107
Messages 108
set_wire_load_mode 109
Mapping 109
Messages 109
set_wire_load_model 110
Mapping 110
Messages 111
set_wire_load_selection_group 112
Mapping 112
Messages 112

Preface

This preface contains the following sections:

- About This Manual on page 8
- Other Information Sources on page 8
- Syntax Conventions on page 9
- About the Graphical User Interface on page 10

About This Manual

This manual describes how to translate constraints in the Synopsys Design Compiler (DC) or PrimeTime (PT) script command languages to the Ambit® BuildGates® synthesis constraint format. The output from the translator is suitable for driving BuildGates synthesis tools.

This manual is for design engineers who are making the transition from Synopsys synthesis tools to BuildGates synthesis tools. To use this manual, you should be familiar with the BuildGates synthesis and Synopsys DC constraint formats.

Other Information Sources

For more information about Ambit BuildGates synthesis and other related products, you can consult the sources listed here.

- *Ambit BuildGates Synthesis User Guide*
- *Command Reference for Ambit BuildGates Synthesis and Cadence PKS*
- *Timing Analysis for Ambit BuildGates Synthesis and Cadence PKS*
- *Test Synthesis for Ambit BuildGates Synthesis and Cadence PKS*
- *HDL Modeling for Ambit BuildGates Synthesis*
- *Distributed Processing of Ambit BuildGates Synthesis*

Depending on the product licenses your site has purchased, you could also have these documents.

- *PKS User Guide*
- *Datapath Option of Ambit BuildGates Synthesis and Cadence PKS*
- *Low Power Option of Ambit BuildGates Synthesis and Cadence PKS*

BuildGates synthesis is often used with other Cadence® tools during various design flows. The following documents provide information about these tools and flows. Availability of these documents depends on the product licenses your site has purchased.

- *Cadence Timing Library Format Reference*
- *Cadence Pearl Timing Analyzer User Guide*
- *Cadence General Constraint Format Reference*

The following books are helpful references.

- IEEE 1364 Verilog HDL LRM
- TCL Reference, *Tcl and the Tk Toolkit*, John K. Ousterhout, Addison-Wesley Publishing Company

Syntax Conventions

This section provides the Text Command Syntax used in this document.

Text Command Syntax

The list below describes the syntax conventions used for the Ambit BuildGates synthesis text interface commands.

Important

Command names and arguments are case sensitive. User-defined information is case sensitive for Verilog designs and, depending on the value specified for the global variable `hdl_vhdl_case`, may be case sensitive as well.

`literal`	Nonitalic words indicate keywords that you must enter literally. These keywords represent command or option names.

`argument`	Words in italics indicate user-defined arguments or information for which you must substitute a name or a value.
`|`	Vertical bars (OR-bars) separate possible choices for a single argument.
`[ ]`	Brackets denote optional arguments. When used with OR-bars, they enclose a list of choices from which you can choose one.
`{ }`	Braces are used to indicate that a choice is required from the list of arguments separated by OR-bars. You must choose one from the list. `{ argument1 | argument2 | argument3 }`
`{ }`	Bold braces are used in Tcl commands to indicate that the braces must be typed in literally.
`...`	Three dots (...) indicate that you can repeat the previous argument. If the three dots are used with brackets (that is, `[argument]...`), you can specify zero or more arguments. If the three dots are used without brackets (`argument...`), you must specify at least one argument, but can specify more.
#	The pound sign precedes comments in command files.

About the Graphical User Interface

This section describes the conventions used for the BuildGates synthesis graphical user interface (GUI) commands and describes how to use the menus and forms in the BuildGates synthesis software.

Using Menus

The GUI commands are located on menus at the top of the window. They can take one of three forms.

CommandName	A command name with no dots or arrow executes immediately.
CommandName...	A command name with three dots displays a form for choosing options.

CommandName ->	A command name with a right arrow displays an additional menu with more commands. Multiple layers of menus and commands are presented in what are called command sequences, for example: *File – Import – LEF*. In this example, you go to the File menu, then the Import submenu, and, finally, the LEF command.

Using Forms

…	A menu button that contains only three dots provides browsing capability. When you select the browse button, a list of choices appears.
Ok	The *Ok* button executes the command and closes the form.
Cancel	The *Cancel* button cancels the command and closes the form.
Defaults	The *Defaults* button displays default values for options on the form.
Apply	The *Apply* button executes the command but does not close the form.
Help	The *Help* button provides information about the command.

1

Getting Started

The Ambit® BuildGates® synthesis constraint translator automatically translates Synopsys constraints into the BuildGates synthesis constraint format.

In the Synopsys environment, constraints can be specified or derived from a variety of sources. Users can enter hand-written constraints in Synopsys design compiler (DC) using the `dc_shell` language constructs, or using the `pt_shell` language constructs for PrimeTime (PT). Both `dc_shell` and `pt_shell` provide constraint commands along with general scripting capability including control structures, and variables. DC and PT are both capable of generating a `write_script` output, which is essentially an unrolled view of the original constraints.

The BuildGates synthesis constraint translator is capable of translating DC or PT `write_script` output (in `dcsh` and Tcl formats) into a BuildGates synthesis format constraints file using the `read_dc_script` command. The translator issues detailed error and warning messages if problems are encountered during translation.

This chapter provides the following information:

- Setting Up the BuildGates Synthesis Constraint Translator on page 14
- Constraint Modeling Guidelines on page 14
- Supported Constructs on page 15
- Unsupported Constructs on page 17
- Translating Constraint Files on page 18

The translator supports `write_script` output from Synopsys design compiler or PrimeTime. The translated constraints can be used with BuildGates synthesis tools running version 3.0 or greater. Figure 1-1 on page 13 shows the translation flow.

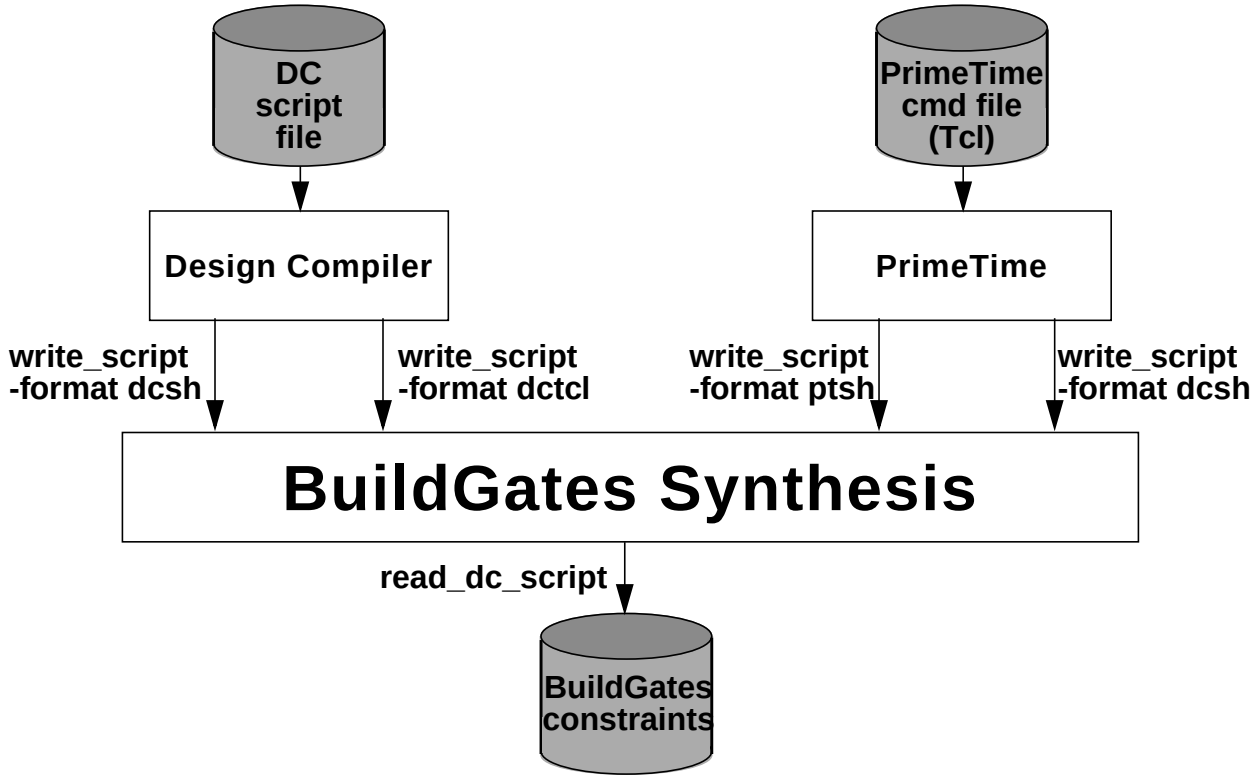

Figure 1-1 Synopsys to BuildGates Synthesis Translation Flow

Setting Up the BuildGates Synthesis Constraint Translator

The translator is invoked using the `read_dc_script` Tcl command in the BuildGates synthesis `ac_shell` or `pks_shell` environments. Prior to invoking the translator, you must do the following:

1. Invoke `ac_shell` (or `pks_shell`)
2. Read in the libraries.
3. Read the design netlist into `ac_shell` (or `pks_shell`)

   This is a must for translation.
4. Run `do_build_generic` (if you read in the RTL netlist)

Now you are ready to translate Synopsys constraint files.

Constraint Modeling Guidelines

For the translator to perform effectively, the following set of guidelines are recommended and should be applied to the Synopsys constraint file being converted.

- The Synopsys constraints should be in the `write_script` output form (using `write_script` without the `-hierarchy` option).
- The name used for the operating condition in the Synopsys construct `set_operating_conditions` should match a name for the operating conditions in the Ambit library (and the two environments must specify the same values for the process, voltage and temperature operating conditions).
- The library and cell names of the driver cell (in the `set_driving_cell` construct) should match those used with BuildGates synthesis.

Note: The translator may run into issues with certain commands, in which case it will issue warning or error messages to the `ac_shell.log` file.

Important

Examine this file closely and manually fix any problem areas before using the translated BuildGates synthesis constraints. To check the accuracy of the translation process, always generate a timing report in the BuildGates synthesis tool and compare this to the report that is generated by Design Compiler or PrimeTime using the original constraints.

Supported Constructs

The following Synopsys constructs are supported by the translator:

- `all_clocks`
- `all_inputs`
- `all_outputs`
- `create_clock`
- `current_design`
- `current_instance`
- `find`
- `get_cells`[1]
- `get_clocks`
- `get_lib_cells`
- `get_libs`
- `get_nets`
- `get_pins`
- `get_ports`
- `set_capacitance`
- `set_case_analysis`
- `set_clock_latency`

1. `get_cells`, `get_clocks`, `get_lib_cells`, `get_libs`, `get_nets`, `get_pins` and `get_ports` can only be used in Tcl format Synopsys scripts.

- `set_clock_skew`
- `set_clock_transition`
- `set_clock_uncertainty`
- `set_disable_timing`
- `set_dont_touch`
- `set_dont_touch_network`
- `set_drive`
- `set_driving_cell`
- `set_false_path`
- `set_fanout_load`
- `set_input_delay`
- `set_input_transition`
- `set_load`
- `set_logic_one`
- `set_logic_zero`
- `set_max_capacitance`
- `set_max_delay`
- `set_max_fanout`
- `set_max_transition`
- `set_min_delay`
- `set_multicycle_path`
- `set_operating_conditions`
- `set_output_delay`
- `set_port_fanout_number`
- `set_propagated_clock`
- `set_resistance`
- `set_wire_load`

- `set_wire_load_mode`
- `set_wire_load_model`
- `set_wire_load_selection_group`

Unsupported Constructs

The constructs below are not supported by the translator. An error message is issued if the translator encounters any of the following constructs:

- `get_lib_pins`
- `set_logic_dc`
- `set_max_area`
- `set_max_time_borrow`
- `set_min_capacitance`
- `set_wire_load_min_block_size`

Translating Constraint Files

From an `ac_shell` (or `pks_shell`) prompt, type:

```
read_dc_script [-ambit ambit_output_file]
     [-time_scale_factor scaling_factor]
     [-capacitance_scale_factor scaling_factor]
     [-resistance_scale_factor scaling_factor]
     [-length_scale_factor scaling_factor] [-no_apply] [-no_mcp_hold_errors]
     [-voltage_scale_factor scaling_factor] [-write_only] [-apply_only] [-tcl]
     [-verbose] [-help] [-V] write_script_file
```

- `-ambit ambit_output_file`

  Specifies the name of the Ambit constraints output file. The default is to print the Synopsys constraints to the standard output device.

- `-time_scale_factor scaling_factor`

  Specifies the scaling factor to be applied to all time values in the Synopsys script during the translation. The default is 1.0.

- `-capacitance_scale_factor scaling_factor`

  Specifies the scaling factor to be applied to all capacitance values in the Synopsys script during the translation. The default is 1.0.

- `-resistance_scale_factor scaling_factor`

  Specifies the scaling factor to be applied to all resistance values in the Synopsys script during the translation. The default is 1.0.

- `-length_scale_factor scaling_factor`

  Specifies the scaling factor to be applied to all length values in the Synopsys script during the translation. The default is 1.0.

- `-write_only`

  Specifies that the translator will only write the translated commands and will not apply them to the current design.

- `-voltage_scale_factor scaling_factor`

  Specifies the scaling factor to be applied to all voltage values in the Synopsys script during the translation. The default is 1.0.

- `-apply_only`

  Specifies that the translator should only apply the translated commands to the current design and not write them to the output file.

- `-tcl`

  Specifies that the Synopsys constraints file is in Tcl format (and not in the default `dc_shell` format).

- `-verbose`[1]

  Specifies that the translator should write the line number and the corresponding Synopsys command in the BuildGates synthesis constraints output along with the translated command.

- `-help`

  Prints the `read_dc_script` usage message.

- `write_script_file`

  Specifies the name of the Synopsys constraints input file. This is a required argument and must be the last specified argument.

1. The false and multicycle path assertions are compressed by the translator; meaning that several Synopsys commands end up corresponding to a single Synopsys command. Rather than printing all the original Synopsys commands, the translator only prints the compressed Synopsys command in comments for the verbose mode.

Globals

The following `set` command is used with the translator:

- `set syn2ambit_tmp_dir dir_name`

  Specifies the directory to store temp files (`dir_name`). If not specified, the translator stores the temp files in the directory used to store the output BuildGates synthesis constraints file. If no BuildGates synthesis constraints file is given (output sent to `stdout`), the translator stores the temp files in the current working directory (`pwd`).

2

Translating Constraints

This chapter describes how to translate Synopsys design compiler (DC) or PrimeTime (PT) constraints for use in both the Ambit® BuildGates® synthesis and Cadence® physically knowledgeable synthesis (PKS) tools. Below is a list of the supported DC constraints and equivalent BuildGates synthesis commands.

Synopsys	BuildGates Synthesis
all_clocks	(No equivalent BuildGates synthesis command. The translator returns a list of all clock names)
all_inputs	find -port -input *
all_outputs	find -port -output *
create_clock	set_clock
	set_clock_arrival_time
current_design	set_current_module
current_instance	set_current_instance
find (design, pattern)	find -module
get_cells	find -instance
find (cell, pattern)	
get_clocks find (clock, pattern)	(No equivalent BuildGates synthesis command. The translator returns a list of specified clock names)
get_lib_cells	find -cellref
find (lib_cell, pattern)	
get_libs	find -techlib
find (library, pattern)	

Synopsys	BuildGates Synthesis
get_nets	find -net
find (net, pattern)	
get_pins	find -pin
find (pin, pattern)	
get_ports	find -port
find (port, pattern)	find -port
find (reference)	find -instance -of _cell _type
set_capacitance (PT command)	set_port_capacitance
	set_wire_capacitance
set_case_analysis	set_constant_for_timing
set_clock_latency	set_clock_insertion_delay
set_clock_skew -ideal|-propagated set_propagated_clock	set_clock_propagation ideal | propagated
set_clock_skew -uncertainty	set_clock_uncertainty
set_clock_transition	set_slew_time -clock -pos
set_clock_uncertainty	set_clock_uncertainty
set_disable_timing cell_name	set_disable_timing -to cell_name/ output_pin_names
set_disable_timing cell_name -from fpin -to tpin	set_disable_timing -from cell_name/ fpin -to cell_name/tpin
set_disable_timing lib_cell_name	set_disable_cell_timing
set_disable_timing pin_name	set_disable_timing -from pin_name
	set_disable_timing -to pin_name
set_dont_touch	set_dont_modify
set_dont_touch_network	set_dont_modify -network
set_drive	set_drive_resistance
set_driving_cell	set_drive_cell
set_false_path	set_false_path

Synopsys	BuildGates Synthesis
set_fanout_load	set_fanout_load
set_input_delay	set_input_delay
set_input_transition	set_slew_time
set_load	set_port_capacitance
	set_wire_capacitance
set_load -fanout_number	set_num_external_sinks
set_logic_one	set_logic1
set_logic_zero	set_logic0
set_max_capacitance	set_port_capacitance_limit -max
set_max_delay	set_path_delay_constraint -late
set_max_fanout	set_fanout_load_limit
set_max_transition	set_slew_time_limit -max
set_min_capacitance	set_port_capacitance_limit -min
set_min_delay	set_path_delay_constraint -early
set_min_transition	set_slew_time_limit -min
set_multicycle_path	set_cycle_addition
set_operating_conditions	set_operating_condition
set_output_delay	set_external_delay
set_port_fanout_number	set_num_external_sinks
set_propagated_clock	set_clock_propagation propagated
set_resistance	set_wire_resistance
set_wire_load	set_wire_load
set_wire_load -mode	set_wire_load_mode
set_wire_load_mode	
set_wire_load_model	set_wire_load
set_wire_load -port_list	set_port_wire_load
set_wire_load_selection_group	set_wire_load_selection_table

all_clocks

The `all_clocks` command does not map to a BuildGates synthesis command. The translator keeps track of all create_clock statements and stores all clocks in a database; it then implements the Synopsys command "`all_clocks`" using this database. A list of the clock names is returned in the argument.

Mapping

Not Applicable

Messages

None

all_inputs

The `all_inputs` command maps to the BuildGates synthesis `find` command. Refer to find on page 32.

Mapping

Synopsys	BuildGates Synthesis
all_inputs	find -port -input *
[-clock `clock_name`]	(No translation)
[-edge_triggered | -level_sensitive]	(No translation)

Messages

Ignored argument -clock (all_inputs)

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Replace the all_inputs command with a list of ports clocked by the specified clock.

Ignored argument -edge_triggered (all_inputs)

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Replace the `all_inputs` command with a list of `edge_triggered` input and inout ports.

Ignored argument -level_sensitive (all_inputs)

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Replace the `all_inputs` command with a list of `level_sensitive` input and inout ports.

all_outputs

The `all_outputs` command maps to the BuildGates synthesis `find` command. Refer to find on page 32.

Mapping

Synopsys	BuildGates Synthesis
all_outputs	find -port -output *
[-clock clock_name]	(No translation)
[-edge_triggered | -level_sensitive]	(No translation)

Messages

Ignored argument -clock (all_outputs)

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Replace the `all_outputs` command with a list of ports clocked by the specified clock.

Ignored argument -edge_triggered (all_outputs)

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Replace the `all_outputs` command with a list of `edge_triggered` output and inout ports.

Ignored argument -level_sensitive (all_outputs)

- Severity -Error
- Action - This message informs you of the BuildGates synthesis command limitations. Replace the `all_outputs` command with a list of `level_sensitive` output and inout ports.

create_clock

The `create_clock` command maps to the BuildGates synthesis `set_clock` command if the `port_pin_list` option is not specified. If the `port_pin_list` option is specified, `create_clock` maps to a combination of `set_clock` and `set_clock_root` commands.

Synopsys requires either the clock name or the clock source list to be specified in a `create_clock` command; if the clock name is omitted, then the name of the first clock source is assumed to be the clock name.

The `set_clock` command defines an ideal clock. An ideal clock must be defined as a global reference signal for all the data signals in the design. The ideal clock is defined by its name, period, and position of the leading edge and trailing edge. In a single clock design, only one `set_clock` command is used. In a multi-phase clock system several `set_clock` commands are used to define each phase of the clock. For purely combinational design, an ideal clock (or any clock) definition is not necessary. The scope of this command is the entire session of `ac_shell`. Once an ideal clock is defined, it remains accessible to all subsequent commands. The clock generated by this command is not saved in the .adb file.

The `set_clock_root` command assigns a polarity to a previously specified ideal clock and associates a list of clock pins or ports with the ideal clock signal.

Mapping

Synopsys	BuildGates Synthesis
create_clock	set_clock
[-name clock_name]	clock_name
-period period_value	-period period_value
[-waveform edge_list]	-waveform lead_time trail_time
[clock_source_list]	set_clock_root -clock clock_name {-pos | -neg} port_pin_list
Examples	
`create_clock -name clk1 -period 10`	`set_clock clk1 -period 10 -waveform {0 5}`
`create_clock -period 5 -waveform {0 3} {clk_src1 clk_src2}`	`set_clock clk_src1 -period 5 -waveform {0 3}` `set_clock_root -clock clk_src1 -pos {clk_src1 clk_src2}`

Messages

Ignoring extraneous waveform edges (create_clock)

- Severity - Error
- Explanation - The waveform specification in the BuildGates synthesis `set_clock` command does not support more than two edges unlike the Synopsys `create_clock` command.
- Action - This message informs you of the BuildGates synthesis command limitations; this clock waveform cannot be modeled in BuildGates synthesis. Remove this clock waveform specification or remodel it using two edges.

The clock period must be specified (create_clock)

- Severity - Error
- Action - Specify the clock period in the `create_clock` command and run the translator again.

Either -clock or source_list must be specified (create_clock)

- Severity - Error
- Explanation - Either a clock name or a clock source list must be specified in Synopsys for it to be a valid command.
- Action - Fix the command and rerun the translator.

current_design

The `current_design` command maps to the BuildGates synthesis `set_current_module` command.

The `set_current_module` command sets the module represented by `module_name` to be the current module. Design objects referenced by subsequent commands can be found relative to this module. All searches for design objects are started in this module. If design objects are referenced hierarchically, `module_name` is used as the root (top) of the hierarchy. If a module has multiple instances and each instance requires a separate set of constraints, then the `set_current_instance` command is used. Note that when the `set_current_module` command is used, the `current_instance` is reset to the `top_timing_module`.

Mapping

Synopsys	BuildGates Synthesis
current_design	set_current_module
design_name	module_name
Examples	
`current_design sub_design1`	`set_current_module sub_design1`

Messages

current_design should have a design specified for an equivalent in Ambit

- Severity - Error
- Explanation - This message arises if no `design_name` is specified; this is acceptable in Synopsys and the command would just return the name of the current design. However, in BuildGates synthesis there is no equivalent command and you would need to replace this command with the name of the current working design.
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with the name of the current working design.

current_instance

The `current_instance` command maps to the BuildGates synthesis `set_current_instance` command.

The `set_current_instance` command sets the current instance of your design.

Mapping

Synopsys	BuildGates Synthesis
current_instance	set_current_instance
instance_name	instance_name
Examples	
`current_instance sub_inst1`	`set_current_instance sub_inst1`

Messages

set_current_instance (Ambit) does not support "." or ".."

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Replace the "`.`" or "`..`" with a fully qualified instance name.

find

The `find` command maps to the BuildGates synthesis `find` command.

The BuildGates synthesis `find` command finds various design objects and prepares the list for other `ac_shell` commands. The design objects are found in the design database using glob style pattern matching, unless the `-regexp` option is used. Multiple wildcard matching in a name is permitted.

Typically, the output of the `find` command (list of objects found by this command) is an argument of another command that accepts a list of objects and performs some task. For example, all input ports of a module can be set to the same arrival time as follows:

```
set_data_arrival_time 0.2 [find -ports -input *] -clock master_ck
```

The search for finding objects is carried out in the current module. If the object name has hierarchy specified in it (`cnt99/*`), then the search is carried out at the appropriate hierarchy level.

Mapping

Synopsys	BuildGates Synthesis
find	find
design	-module
clock	(Implemented by translator)[a]
port	-port
reference	-instance of_cell_type
cell	-instance
net	-net
pin	-pin
cluster	(No translation)
multi-bit	(No translation)
library	-techlib
lib_cell	-cellref
lib_pin	(No translation)

Synopsys	BuildGates Synthesis
operator	(No translation)
module	(No translation)
implementation	(No translation)
file	(No translation)
-hierarchy	-hierarchical
-flat	(No translation)
(no translation)	`-full_path_name`[b]
pattern	pattern
Examples	
`find(cell, inst1)`	`find -instance -full_path_name inst1`

a. The translator keeps track of all `create_clock` statements and stores all clocks in a database; it then implements the Synopsys command "`find clock`" using this database. This requires that all clocks be defined prior to using the "`find clock`" command.

b. By default, the BuildGates synthesis `find` command returns object ids; since Synopsys `find` returns object names, the translator always uses the "`-full_path_name`" switch for the BuildGates synthesis command to return object names instead of ids.

Messages

Ignored argument -flat (find)

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Replace the `find` command with a list of objects that this `find` command returns in Synopsys.

Synopsys object type 'lib_pin' is not supported in Ambit

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations.

  Note: A similar error is issued for the other unsupported types - `cluster`, `multibit`, `operator`, `module`, `implementation` and `file`.

get_cells

The `get_cells` command maps to the BuildGates synthesis `find -instance` command. Refer to find on page 32.

Mapping

Synopsys	BuildGates Synthesis
get_cells	find -instance
-quiet	(No translation)
-regexp	(No translation)
-exact	(No translation)
-nocase	(No translation)
-of_objects objects	(No translation)
-hierarchical	-hierarchical
-filter @ref_name==cell_name	-of_cell_type cell_name
-filter @is_black_box==true	-blackboxes
-filter @dont_touch==true	-dont_modify
-filter @is_sequential==true	-registers
-filter filter_expr	(No translation)
pattern	pattern
Examples	
`get_cells inst1`	`find -instance -full_path_name inst1`

Messages

-quiet is not supported in dc_shell

- Severity - Warning
- Action - This message informs you of an error in the Synopsys command; fix and rerun the translator.

  Note: Similar warnings are also issued for `-regexp`, `-exact`, and `-nocase`.

'get_cells -of_objects' cannot be supported in Ambit

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with an alternate command that is supported by the translator.

-register selects all sequential leaf cells while -sequential selects both hierarchical and leaf sequential cells

- Severity - Warning
- Action - This message informs you of a difference between BuildGates synthesis and Synopsys. Check if the translation is acceptable.

'filter[1]' is not supported in Ambit

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with an alternate command that is supported by the translator.

1. The actual filter expression is substituted here in the error message.

get_clocks

The `get_clocks` command is similar to the all_clocks command; there is no equivalent BuildGates synthesis command. This command is implemented by the translator (the same way as it does with the `all_clocks` command). The translator requires that all clock definition statements precede any `get_clocks` command.

Mapping

Synopsys	BuildGates Synthesis
get_clocks	(Implemented by translator)
-quiet	(No translation)
-regexp	(No translation)
-nocase	(No translation)
-filter filter_expr	(No translation)
pattern	pattern

Messages

'-quiet' is not supported in dc_shell

- Severity - Warning
- Action - This message informs you of an error in the Synopsys command; fix and rerun the translator.

  Note: Similar warnings are also issued for `-regexp` and `-nocase`.

Ignored argument -filter (get_clocks)

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with an alternate command that is supported by the translator.

get_lib_cells

The `get_lib_cells` command maps to the BuildGates synthesis `find -cellref` command. Refer to find on page 32.

Mapping

Synopsys	BuildGates Synthesis
get_lib_cells	find -cellref
-quiet	(No translation)
-regexp	(No translation)
-exact	(No translation)
-nocase	(No translation)
-of_objects objects	(No translation)
-filter filter_expr	(No translation)
pattern	pattern
Examples	
`get_lib_cells and*`	`find -cellref -full_path_name and*`

Messages

'-quiet' is not supported in dc_shell

- Severity - Warning
- Action - This message informs you of an error in the Synopsys command; fix and rerun the translator.

  Note: Similar warnings are also issued for `-regexp`, `-exact`, `-nocase` and `of_objects`.

Filters are not supported for cellrefs in Ambit

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with an alternate command that is supported by the translator.

get_libs

The `get_libs` command maps to the BuildGates synthesis `find -techlib` command. Refer to find on page 32.

Mapping

Synopsys	BuildGates Synthesis
get_libs	find -techlib
-quiet	(No translation)
-regexp	(No translation)
-exact	(No translation)
-nocase	(No translation)
-of_objects objects	(No translation)
-filter filter_expr	(No translation)
pattern	pattern
Examples	
`get_libs lca500*`	`find -techlib -full_path_name lca500*`

Messages

'-quiet' is not supported in dc_shell

- Severity - Warning
- Action - This message informs you of an error in the Synopsys command; fix and rerun the translator.

  Note: Similar warnings are also issued for `-regexp`, `-exact`, `-nocase` and `of_objects`.

Filters are not supported for techlibs in Ambit

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with an alternate command that is supported by the translator.

get_nets

The `get_nets` command maps to the BuildGates synthesis `find -net` command. Refer to find on page 32.

Mapping

Synopsys	BuildGates Synthesis
get_nets	find -net
-quiet	(No translation)
-regexp	(No translation)
-exact	(No translation)
-nocase	(No translation)
-of_objects objects	(No translation)
-hierarchical	-hierarchical
-filter filter_expr	(No translation)
pattern	pattern
Examples	
`get_nets net1`	`find -net -full_path_name net1`

Messages

'-quiet' is not supported in dc_shell

- Severity - Warning
- Action - This message informs you of an error in the Synopsys command; fix and rerun the translator.

  Note: Similar warnings are also issued for `-regexp`, `-exact`, `-nocase` and `of_objects`.

'filter[1]' is not supported in Ambit

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with an alternate command that is supported by the translator.

1. The actual filter expression is substituted here in the error message.

get_pins

The `get_pins` command maps to the BuildGates synthesis `find -pin` command. Refer to find on page 32.

Mapping

Synopsys	BuildGates Synthesis
get_pins	find -pin
-quiet	(No translation)
-regexp	(No translation)
-exact	(No translation)
-nocase	(No translation)
-of_objects objects	(No translation)
-hierarchical	-hierarchical
-filter filter_expr	(No translation)
pattern	pattern
Examples	
`get_pins inst1/pin1`	`find -pin -full_path_name inst1/pin1`

Messages

'-quiet' is not supported in dc_shell

- Severity - Warning
- Action - This message informs you of an error in the Synopsys command; fix and rerun the translator.

  Note: Similar warnings are also issued for `-regexp`, `-exact`, `-nocase` and `of_objects`.

'filter[1]' is not supported in Ambit

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with an alternate command that is supported by the translator.

1. The actual filter expression is substituted here in the error message.

get_ports

The `get_ports` command maps to the BuildGates synthesis `find -port` command. Refer to find on page 32.

Mapping

Synopsys	BuildGates Synthesis
get_ports	find -port
-quiet	(No translation)
-regexp	(No translation)
-exact	(No translation)
-nocase	(No translation)
-of_objects objects	(No translation)
-hierarchical	-hierarchical
-filter filter_expr	(No translation)
pattern	pattern
Examples	
`get_ports in1`	`find -port -full_path_name in1`

Messages

'-quiet' is not supported in dc_shell

- Severity - Warning
- Action - This message informs you of an error in the Synopsys command; fix and rerun the translator.

  Note: Similar warnings are also issued for `-regexp`, `-exact`, `-nocase` and `of_objects`.

'filter[1]' is not supported in Ambit

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with an alternate command that is supported by the translator.

1. The actual filter expression is substituted here in the error message.

set_capacitance

The `set_capacitance` command in PrimeTime has been replaced by the set_load constraint. Refer to set_load on page 79 for more information.

set_case_analysis

The `set_case_analysis` command maps to the BuildGates synthesis `set_constant_for_timing` command.

The `set_constant_for_timing` command sets the value of the specified pin(s) to either a `1` or `0` for use by the timing engine. The applied state value(s) propagate through the corresponding combinational logic cone(s) and enable or disable sections of logic.

Mapping

Synopsys	BuildGates Synthesis
set_case_analysis	set_constant_for_timing
-port_list	-port_list

Messages

None

set_clock_latency

The `set_clock_latency` command maps to the BuildGates synthesis `set_clock_insertion_delay` command.

The `set_clock_insertion_delay` command is used to specify clock insertion delay in the ideal and propagated modes. Clock insertion delay has two components

Source latency

Specified by the `-source` option, source latency is the delay from the external clock generator to the port of the design. Source insertion delay is considered in both ideal and propagated modes.

Network latency (clock tree insertion delay)

The default delay type. Network delay is the delay from the clock port to the register. This delay is only considered in ideal mode. In propagated mode, this delay is replaced by the actual clock tree delays.

Mapping

Synopsys	BuildGates Synthesis
set_clock_latency	set_clock_insertion_delay
Object type clock waveform (no -source):	
-rise	-lead
-fall	-trail
-min	-pvt min
-max	-pvt max
delay	delay
find (clock)	*clock name*
Object type clock waveform (w/-source)	
-rise	-lead
-fall	-trail
-late	-late

Synopsys	BuildGates Synthesis
-early	-early
-source	-source
delay	delay
find (clock)	*clock name*
Object type clock tree pin (no -source)	
-rise	-rise[a]
-fall	-fall
-min	-pvt min
-max	-pvt max
delay	delay
find (port)	-pin *clock name*
Object type clock tree pin (w/-source)	
-rise	-rise[a]
-fall	-fall
-early	-early
-late	-late
-source	-source
delay	delay
find (port)	-pin *clock name*

a.The Synopsys -rise and -fall options refer to the signal on the port or the clock pin of the register. The BuildGates synthesis software refers to the clock port in which the assertion was set. This causes wrong translations if an inverter is between the port and clock pin of the register.

Messages

None

set_clock_skew

The `set_clock_skew` command maps to the BuildGates synthesis `set_clock_propagation` and `set_clock_uncertainty` commands.

Note that the `set_clock_skew` command has been replaced by the following Synopsys commands: `set_propagated_clock, set_clock_latency, set_clock_uncertainty, remove_propagated_clock, remove_clock_latency, remove_clock_uncertainty` in Synopsys version 1999.05 and later. However the translator does support this command for older scripts.

The `set_clock_propagation` command tells the timing analyzer to treat all clock signals associated with the current timing top module as either ideal or propagated signals for the purpose of timing analysis. One of the signal types must be specified when using this command. Also refer to set_clock_uncertainty on page 57.

Mapping

Synopsys	BuildGates Synthesis
set_clock_skew	set_clock_propagation
	set_clock_uncertainty
-ideal	set_clock_propagation ideal
-propagated	set_clock_propagation propagated
Object type clock waveform: -uncertainty uncertainty_value	set_clock_uncertainty -ideal | -propagated -clock_to clock_name -edge_to leading uncertainty_value[a]
	set_clock_uncertainty -ideal | -propagated -clock_to clock_name -edge_to trailing uncertainty_value
Object type clock tree pin: -uncertainty uncertainty_value	set_clock_uncertainty -ideal | -propagated -to -pin clock_name -edge_to leading uncertainty_value[b]
	set_clock_uncertainty -ideal | -propagated -to -pin clock_name -edge_to trailing uncertainty_value

Synopsys	BuildGates Synthesis
Object type clock waveform: -minus_uncertainty uncertainty_value	set_clock_uncertainty -ideal | -propagated -late -clock_to clock_name -edge_to leading uncertainty_value set_clock_uncertainty -ideal | -propagated -late -clock_to clock_name -edge_to trailing uncertainty_value
Object type clock tree pin: -minus_uncertainty uncertainty_value	set_clock_uncertainty -ideal | -propagated -late -to -pin clock_name -edge_to leading uncertainty_value set_clock_uncertainty -ideal | -propagated -late -to -pin clock_name -edge_to trailing uncertainty_value
Object type clock waveform: -plus_uncertainty uncertainty_value	set_clock_uncertainty -ideal | -propagated -early -clock_to *clock_name* -edge_to leading *uncertainty_value* set_clock_uncertainty -ideal | -propagated -early -clock_to *clock_name* -edge_to trailing *uncertainty_value*
Object type clock tree pin: -plus_uncertainty uncertainty_value	set_clock_uncertainty -ideal | -propagated -early -to -pin *clock_name* -edge_to leading *uncertainty_value* set_clock_uncertainty -ideal | -propagated -early -to -pin *clock_name* -edge_to trailing *uncertainty_value*
-min	-pvt min
-max	-pvt max
Object type clock waveform: -delay delay_value	set_clock_insertion_delay -clock *clock_name* delay_value
Object type clock tree pin: -delay delay_value	set_clock_insertion_delay -pin *clock tree pin list* delay_value
Object type clock waveform: -rise_delay delay_value	set_clock_insertion_delay -lead[b] -clock *clock_name* delay_value

Synopsys	BuildGates Synthesis
Object type clock tree pin: -rise_delay delay_value	set_clock_insertion_delay -rise -pin *`clock tree pin list`* delay_value
Object type clock waveform: -fall_delay delay_value	set_clock_insertion_delay -trail[b] -clock *`clock_name`* delay_value
Object type clock tree pin: -fall_delay delay_value	set_clock_insertion_delay -fall -pin *`clock tree pin list`* delay_value
`object_list`[c]	See above.
Examples	
`set_clock_skew -propagated all_clocks()`	`set_clock_propagation propagated`
`set_clock_skew -uncertainty 0.5 ideal_clk1`	`set_clock_uncertainty -clock_to ideal_clk1 -edge_to leading 0.5` `set_clock_uncertainty -clock_to ideal_clk1 -edge_to trailing 0.5`

a. The default behavior in the BuildGates synthesis `set_clock_uncertainty` command is that only the leading edge of the destination clock is affected by the uncertainty specification, while in Synopsys both the leading and trailing edges are affected. To model Synopsys behavior, two BuildGates synthesis `set_clock_uncertainty` statements need to be issued by the translator; one for each destination clock edge.

b. This assumes a positive clock.

c. Synopsys allows you to specify clocks, ports, cells or pins in the `object_list` for propagation mode, delay and uncertainty characteristics. On the other hand, BuildGates synthesis only supports propagated clock delays for all clock networks and not on selected networks. For uncertainty, BuildGates synthesis only supports clocks, pins, or ports; for all other object types, the translator issues warnings.

Messages

The BuildGates synthesis `set_clock_propagation` command does not allow selected clock networks to have propagated clock latency.

Setting all clock networks to have propagated clock latency

- Severity - Warning
- Explanation - BuildGates synthesis only supports propagated clock delays for all clock networks and not on selected networks. The translator goes ahead and sets all clock networks to have propagated clock latency.
- Action - This message informs you of the BuildGates synthesis command limitations. Check if the translator action is acceptable.

Setting the entire clock network to have ideal clock latency

- Severity - Warning
- Explanation - BuildGates synthesis only supports propagated clock delays for the entire clock network and not on selected portions. The translator goes ahead and sets the entire clock network to have ideal clock latency.
- Action - This message informs you of the BuildGates synthesis command limitations.

Object_name is not a clock, pin, or port and is being ignored

where *Object_name* is the actual object name in the warning message.

- Severity - Warning
- Explanation - This message arises for uncertainty specification on unsupported object types.
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with an alternate command that is supported in BuildGates synthesis.

set_clock_transition

The set_clock_transition command maps to the BuildGates synthesis `set_slew_time -clock -pos` command.

The `set_slew_time` command is used to specify the slew time for a clock on the top level module of a design or for a pin on an instance of a technology cell anywhere in the hierarchy. It is useful only in specifying a hard ramp signal as if the pin were driven by a very powerful driver. It does not change the arrival time of an input, but changes the slew time used to compute the delay of the cell on the sink of a net.

The `-clock` option specifies the relationship between the slew time and the ideal clock specified by `clock_name`. The default is `@`, a signal without an associated clock.

The `-pos` option specifies that the slew time should be applied to an actual clock having a positive polarity with respect to the ideal clock.

Note: The `set_slew_time` command is ignored for a port if `set_drive_cell` or `set_drive_resistance` constraints have been set for that port.

Mapping

Synopsys	BuildGates Synthesis
set_clock_transition	set_slew_time -clock -pos
	set_slew_time -clock -neg
-min	-early
-max	-late
-rise	(Not supported)
-fall	(Not supported)
transition_value	slew_value
clock_list	clock_list
Examples	
`set_clock_transition -min 0.5 {clk1}`	`set_slew_time -clock -pos -early 0.5 {clk1}`

Messages

A clock list must be specified to set_clock_transition

- Severity - Error

set_clock_uncertainty

The `set_clock_uncertainty` command maps to the BuildGates synthesis `set_clock_uncertainty` command.

The `set_clock_uncertainty` command is used to specify the clock uncertainty between two clock edges for one or more paths. The specified uncertainty applies to paths starting from the `clock_from` signal and ending at registers clocked by the `clock_to` signal.

Mapping

Synopsys	BuildGates Synthesis
set_clock_uncertainty	set_clock_uncertainty
-from from_clock	-clock_from from_clock
-to to_clock	-clock_to to_clock
-rise	-edge_to leading; for clock waveform[a] -rise; for clock tree pins or ports
-fall	-edge_to trailing; for clock waveform[a] -fall; for clock tree pins or ports
-setup	-late
-hold	-early
uncertainty_value	uncertainty_value
`object_list`[b]	object_list -ideal | propagated[c]
Examples	

`set_clock_uncertainty -from clk1 -to clk2 0.2`	`set_clock_uncertainty -ideal -clock_from clk1 -edge_from leading -clock_to clk2 -edge_to leading 0.2`[d]
	`set_clock_uncertainty -ideal -clock_from clk1 -edge_from leading -clock_to clk2 -edge_to trailing 0.2`
	`set_clock_uncertainty -ideal -clock_from clk1 -edge_from trailing -clock_to clk2 -edge_to leading 0.2`
	`set_clock_uncertainty -ideal -clock_from clk1 -edge_from trailing -clock_to clk2 -edge_to trailing 0.2`

`set_clock_uncertainty 0.5 ideal_clk1`	`set_clock_uncertainty -ideal -clock_to ideal_clk1 -edge_to leading 0.5`[e]

Synopsys	BuildGates Synthesis
	`set_clock_uncertainty -ideal -clock_to ideal_clk1 -edge_to trailing 0.5`

a. For positive clock polarity.

b. Synopsys allows you to specify clocks, ports, cells or pins in the `object_list` for `set_clock_uncertainty`. On the other hand, BuildGates synthesis supports clocks, ports, or pins; for all other object types, the translator issues warnings.

c. This switch is set based on the value of set_propagated_clock on set_clock_skew.

d. The default behavior in the BuildGates synthesis command `set_clock_uncertainty` is that only the leading edge of the source and destination clocks is affected by the uncertainty specification, while in Synopsys both the leading and trailing edges are affected. To model Synopsys behavior, four BuildGates synthesis `set_clock_uncertainty` statements need to be issued by the translator; one for each source and destination clock edge combinations (leading/leading, leading/trailing, trailing/leading and trailing/trailing).

e. The default behavior in the BuildGates synthesis command `set_clock_uncertainty` is that only the leading edge of the destination clock is affected by the uncertainty specification, while in Synopsys both the leading and trailing edges are affected. To model Synopsys behavior, two BuildGates synthesis `set_clock_uncertainty` statements need to be issued by the translator; one for each destination clock edge.

Messages

Set_clock_uncertainty does not allow both inter-clock and simple uncertainty to be specified in a single command

- Severity - Error
- Action - This message informs you of an error in the Synopsys command; fix and rerun the translator.

Both -from and -to clocks must be specified for inter-clock uncertainty

- Severity - Error
- Action - This message informs you of an error in the Synopsys command; fix and rerun the translator.

Edge specification is invalid for simple clock uncertainty

- Severity - Warning
- Action - This message informs you of an error in the Synopsys command; fix and rerun the translator.

set_disable_timing

The `set_disable_timing` command maps to the BuildGates synthesis `set_disable_timing` and `set_disable_cell_timing` commands.

The `set_disable_timing` command disables (snips) timing arc(s) of an instance or interconnect and is used for breaking combinational loops. This command differs from the `set_false_path` command as follows: `set_disable_timing` actually physically snips a timing arc such that neither arrival times nor constant values propagate through it. `set_false_path` selectively disables the arrival time information depending upon the precedence rules. Constant values are not affected by the `set_false_path` command.

The `set_disable_cell_timing` command disables all timing arcs (delay as well as check arcs) between the specified `from` and `to` pins for the specified cell in the library. The timing arcs `from`/`to` internal pins inside IP cells can also be disabled.

Mapping

Synopsys	BuildGates Synthesis
set_disable_timing	set_disable_timing
-from from_pin_name	-from from_pin_name
-to to_pin_name	-to *to_pin_name*
-restore	(No translation)
pin_list	set_disable_timing -from *pin_name*
	set_disable_timing -to *pin_name*
instance_list	set_disable_timing -to *instance_name/**[a]
	set_disable_timing -from *instance_name/ from_pin_name* -to *instance_name/ to_pin_name*[b]
cellref_list[c]	set_disable_cell_timing -cell *cellref* -from *from_pin_name* -to *to_pin_name*

Examples

`set_disable_timing {inst1 pin1}`	`set_disable_timing -to inst1/*`
	`set_disable_timing -from pin1`
	`set_disable_timing -to pin1`

Synopsys	BuildGates Synthesis
`set_disable_timing {inst2 inst3} -from A -to B`	`set_disable_timing -from inst2/A -to inst2/B` `set_disable_timing -from inst3/A -to inst3/B`
`set_disable_timing {cell1} -from In -to Out`	`set_disable_cell_timing -cell cell1 -from In -to Out`

a. If the `from`/`to` pins are not specified in the Synopsys command, then this is the BuildGates synthesis equivalent.

b. If the `from`/`to` pins are specified, then this is the BuildGates synthesis equivalent.

c. For cellrefs, the `from`/`to` pins must be specified.

Messages

Unsupported argument -restore (set_disable_timing)

- Severity - Warning
- Action - This message informs you of the BuildGates synthesis command limitations. Remove both this statement and the statement which was used to disable timing for the specified pins or cells.

Object type 'object_type[1]' with -from and -to is unsupported

- Severity - Error
- Action - This message informs you of an error in the `set_disable_timing` statement; fix the command and rerun the translator.

Unable to locate specified object name object_name[2]

- Severity - Error
- Action - This message informs you of an error in the `set_disable_timing` statement; fix the command and rerun the translator.

1. The actual object type is substituted here in the error message.
2. The actual object name is substituted here in the error message.

The 'cellref' object type is not supported without specifying the from/to pins

- Severity - Error
- Action - This message informs you of an error in the `set_disable_timing` statement; fix the command and rerun the translator.

Set_disable_timing with only one of -from or -to is unsupported

- Severity - Error
- Action - This message informs you of an error in the `set_disable_timing` statement; fix the command and rerun the translator.

Unsupported object type 'object_type[1]'

- Severity - Error
- Action - This message informs you of an error in the `set_disable_timing` statement; fix the command and rerun the translator.

1. The actual object type is substituted here in the error message.

set_dont_touch

The `set_dont_touch` command maps to the BuildGates synthesis `set_dont_modify`, `set_cell_property`, or `reset_dont_modify` commands.

The `set_dont_modify` command is used to keep the instances, modules, ports, and nets from being modified. When a net is specified, all the connecting instances are affected by the set_dont_modify property. This is useful when those objects have already been optimized. No further optimization will be done on the specified instance, module or net. The command is also useful when macros are instantiated from a library. Once a module is marked for no modifications, the commands `do_uniquely_instantiate` or `do_dissolve_hierarchy` are ignored for that module. If a module has a `dont_touch` attribute applied to it, then before dissolving any instance of that module, the parent of the module is checked for the `dont_modify` flag.

The `reset_dont_modify` command removes the `dont_modify` attribute set by a previous `set_dont_modify` command, applied to a design module, instance, or port. Use `set_cell_property` for applying and removing properties on library cells.

Mapping

Synopsys	BuildGates Synthesis
set_dont_touch	set_dont_modify
object_list[a]	object_list
reference (library_cells)	set_cell_property
true[b]	(No translation)
|false	reset_dont_modify object_list
Examples	
`set_dont_touch {net1 cell1}`	`set_dont_modify {net1 cell1}`
`set_dont_touch {net2} false`	`reset_dont_modify {net2}`

a. Synopsys allows `dont_touch` attribute on cells, nets, references, and designs in the `current_design`, and on library cells while BuildGates synthesis only allows it on instances, modules, ports, pins, and nets. The common subset is instances, references, and nets which are supported by the translator. `references` (lib cells) are supported using the `set_cell_property` command; all other object types are unsupported.

b. Setting the `dont_touch` attribute is the default behavior in both Synopsys and BuildGates synthesis; so no translation is required for `true`.

Messages

The object list specified to set_dont_touch does not contain any Ambit supported object types

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with an alternate command that is supported in BuildGates synthesis.

set_dont_touch_network

The `set_dont_touch_network` command maps to the BuildGates synthesis `set_dont_modify` command. Refer to set_dont_touch on page 63.

Mapping

Synopsys	BuildGates Synthesis
set_dont_touch_network	set_dont_modify -network -hierarchical
object_list[a]	object_list
Examples	
`set_dont_touch_network {port1}`	`set_dont_modify -network {port1}`
`set_dont_touch_network {clock}`	`(No translation)`[b]

a. Synopsys allows `dont_touch_network` attribute on clocks, pins or ports in the `current_design`, while BuildGates synthesis only allows it on instances, modules, ports, pins, and nets. The common subset is ports and pins which are supported by the translator; all other object types are unsupported.

b. Note that BuildGates synthesis does not modify clock networks.

Messages

Object_name[1] is being ignored because Ambit does not support pins in set_dont_modify

- Severity - Warning
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with an alternate command that is supported in BuildGates synthesis.

1. The actual object name is substituted here in the warning message.

Object_name[1] is being ignored because Ambit does not support clocks in set_dont_modify

- Severity - Warning
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with an alternate command that is supported in BuildGates synthesis.

Object_name[2] is not a port or a pin or a clock and is being ignored

- Severity - Warning
- Action - This message informs you of an error in the `set_dont_touch_network` statement; fix the command and rerun the translator.

The object list specified to set_dont_touch_network does not contain any ports

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with an alternate command that is supported in BuildGates synthesis.

1. The actual object name is substituted here in the warning message.
2. The actual object name is substituted here in the warning message.

set_drive

The `set_drive` command maps to the BuildGates synthesis `set_drive_resistance` command.

The `set_drive_resistance` command is a simpler version of the `set_drive_cell` command and can be used in many situations where the drive resistance can be specified. It is only used for timing analysis. It does not affect the electrical properties of the design. It is used to specify the drive resistance of a cell. It computes an offset to the arrival time of an input and also changes the slew time used to compute the delay of the cell on the sink of the net.

Mapping

Synopsys	BuildGates Synthesis
set_drive	set_drive_resistance
-min	-early
-max	-late
-rise	-rise
-fall	-fall
drive_value	drive_value
port_or_pin_list	port_or_pin_list
Examples	
`set_drive -min 5 port1`	`set_drive_resistance -early 5 port1`

Messages

None

set_driving_cell

The `set_driving_cell` command maps to the BuildGates synthesis `set_drive_cell` command.

The `set_drive_cell` command is used to accurately model the drive capability of an external driver connected to the input port. It is only used for timing analysis. It does not affect the electrical properties of the design. This means that the capacitance of the output pin of the driving cell does not add to the total capacitance of the input port.

Mapping

Synopsys	BuildGates Synthesis
set_driving_cell	set_drive_cell
-lib_cell *cell_name*	-cell cell_name
-library *lib_name*	-library *lib_name*
-pin *pin_name*	-pin pin_name
-from_pin *from_pin_name*	-from_pin *from_pin_name*
-input_transition_rise *rise_value*	-rise_source_slew rise_value
-input_transition_fall *fall_value*	-fall_source_slew *fall_value*
-rise[a]	-cell_rise *cell_name*
	?-library_rise *lib_name*?
	?-pin_rise *pin_name*?
	?-from_pin_rise *from_pin_name*?
-fall[b]	-cell_fall *cell_name*
	?-library_fall *lib_name*?
	?-pin_fall *pin_name*?
	?-from_pin_fall *from_pin_name*?
-dont_scale	(No translation)
-multiply_by *factor*	(No translation)
-no_design_rule	(No translation)
-none	(No translation)

Synopsys	BuildGates Synthesis
-cell *cell_name*[c]	-cell cell_name
-cell_rise *cell_name*	-cell_rise cell_name
-cell_fall *cell_name*	-cell_fall cell_name
-library_rise *lib_name*	-library_rise *lib_name*
-library_fall *lib_name*	-library_fall *lib_name*
-pin_rise *pin_name*	-pin_rise pin_name
-pin_fall *pin_name*	-pin_fall pin_name
-from_pin_rise *from_pin_name*	-from_pin_rise *from_pin_name*
-from_pin_fall *from_pin_name*	-from_pin_fall *from_pin_name*
port_list	port_list
Examples	
`set_driving_cell -lib_cell INV -pin Z -library tech_lib {IN1}`	`set_drive_cell -cell INV -pin Z -library tech_lib {IN1}`
`set_driving_cell -lib_cell INV -pin Z -library tech_lib -rise {IN2}`	`set_drive_cell -cell_rise INV -pin_rise Z -library_rise tech_lib {IN2}`
`set_driving_cell -lib_cell BUF -pin Z -library tech_lib -fall {IN2}`	`set_drive_cell -cell_fall BUF -pin_fall Z -library_fall tech_lib {IN2}`

a. `-rise`, combined with `-lib_cell`, `-library`, `-pin`, and `-from_pin`, replaces the obsolete `-cell_rise`, `-library_rise`, `-pin_rise`, and `-from_pin_rise` options. Consequently, `-rise` cannot be used with those obsolete options.

b. `-fall`, combined with `-lib_cell`, `-library`, `-pin`, and `-from_pin` is intended to replace the obsolete `-cell_fall`, `-library_fall`, `-pin_fall`, `-from_pin_fall` options, respectively. Consequently, `-fall` cannot be used with those obsolete options.

c. The Synopsys options `-cell`, `-cell_rise`, `-cell_fall`, `-library_rise`, `-library_fall`, `-pin_rise`, `-pin_fall`, `-from_pin_rise` and `-from_pin_fall` have been made obsolete in v1999.05, but are still functional. So the translator does support these options.

Messages

Ignored argument -dont_scale (set_driving_cell)

- Severity - Warning
- Action - This message informs you of the BuildGates synthesis command limitations. Check whether the translator action is acceptable.

Ignored argument -multiply_by (set_driving_cell)

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with an alternate command that is supported in BuildGates synthesis.

Ignored argument -no_design_rule (set_driving_cell)

- Severity - Warning
- Action - This message informs you of the BuildGates synthesis command limitations. Check whether the translator action is acceptable.

Unsupported argument -none (set_driving_cell)

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Remove this statement and the earlier statement that specified a driving cell for the specified ports.

Cell must be specified

- Severity - Error
- Action - A driving cell must be specified; fix the command and rerun the translator.

set_false_path

The `set_false_path` command maps to the BuildGates synthesis `set_false_path` command.

The `set_false_path` command is used in timing analysis to identify false paths in a design. All paths starting in one clock domain and ending in another clock domain are ignored. You can not mix clock domain paths and explicit paths (specified using `-from`, `-to` and `-through`). Path exceptions between clock domains are accepted. The command is also used to break or disable specific instance timing arcs in a design.

Mapping

Synopsys	BuildGates Synthesis
set_false_path	set_false_path
-hold	-early
-setup	-late
-rise[a]	-rise_to
-fall	-fall_to
-reset_path	(No translation)
-from *object_list*[b]	-from *pin_port_cell_list*
	-clock_from *clock_list*
-through *pin_port_cell_list*[c]	-through *pin_port_cell_list*
-to *object_list*	-to *pin_port_cell_list*
	-clock_to *clock_list*

Examples

`set_false_path -from {in1 inst1} -to {out1 inst2}`	`set_false_path -from {in1 inst1} -to {out1 inst2}`
`set_false_path -through i1 -through i2 -through i3`	`set_false_path -through i1 -through i2 -through i3`
`set_false_path -setup -from clk2 -to clk3`	`set_false_path -late -clock_from clk2 -clock_to clk3`
`set_false_path -rise -to u10`	`set_false_path -rise_to u10`

Synopsys	BuildGates Synthesis

`set_false_path -from {u1 clk1} -through {u2} -to {u2 clk2}`	`set_false_path -from {u1} -through {u2} -to {u2}`
	`set_false_path -from {u1} -through {u2} -clock_to {clk2}`
	`set_false_path -clock_from {clk1} -through {u2} -to {u2}`
	`set_false_path -clock_from {clk1} -through {u2} -clock_to {clk2}`

a. Synopsys allows you to mark only rising or falling delays false, as measured on the path endpoint. BuildGates synthesis allows you to specify rising or falling false paths, as measured on the path endpoint, ONLY if the to option is also specified in the false path, in which case `rise_to`/`fall_to` replace the to option.

b. Synopsys allows clocks, pins, ports or cells in the `from`/`to` lists. BuildGates synthesis also allows all these object types in its `set_false_path` command, but the clocks need to be specified in a separate list from the other object types. The translator takes care of doing this separation.

c. Both Synopsys and BuildGates synthesis allow multiple through options to be specified in a `set_false_path` command.

Messages

Unsupported argument -reset_path (set_false_path)

- Severity - Warning
- Action - This message informs you of the BuildGates synthesis command limitations. Remove this statement and the earlier statement that specified this path as a false path.

Ignoring '-rise' for 'to_clock_list[1]' since Ambit only supports edge specification if the path endpoint is specified

- Severity - Warning
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this construct with a statement that is supported in BuildGates synthesis.

  Note: A similar warning is issued for ‘-fall’.

1. The actual clock names are substituted here in the warning message.

Ignoring clocks 'clock_list[1]' in -through list

- Severity - Warning
- Action - Fix the Synopsys command and rerun the translator.

Unable to locate objects specified to set_false_path in from/through/to lists

- Severity - Error
- Explanation - This message appears if the objects specified in the `from`/`through`/`to` lists cannot be found in the current design.
- Action - Fix the Synopsys command and rerun the translator.

1. The actual clock names are substituted here in the warning message.

set_fanout_load

The `set_fanout_load` command maps to the BuildGates synthesis `set_fanout_load` command.

The `set_fanout_load` command specifies the fanout load on the ports of a cell. While port capacitance affects timing analysis, fanout loads are used to enforce the design rule checks.

Mapping

Synopsys	BuildGates Synthesis
set_fanout_load	set_fanout_load
-fanout_load_value	-fanout_load_value
-port_list	-port_list

Messages

-port_list must be specified to set_fanout_load

- Severity - Error

set_input_delay

The `set_input_delay` command maps to the BuildGates synthesis `set_input_delay` command.

The `set_input_delay` command specifies the data arrival time for signals at data input ports. Signal changes arrive at the input ports of a module at a certain time. The arrival time of a signal is associated with a clock edge that causes the signal change.

Mapping

Synopsys	BuildGates Synthesis
set_input_delay	set_input_delay
-min	-early
-max	-late
-rise	-rise
-fall	-fall
-clock *clock_name*	-clock *clock_name*
-clock_fall	-trail
-add_delay	-worst_case
-level_sensitive	(No translation)
delay_value	delay_value
port_or_pin_list	port_or_pin_list
Examples	
`set_input_delay -min -clock clk1 5 {port1 pin1}`	`set_input_delay -early -clock clk1 5 {port1 pin1}`
`set_input_delay -max -clock clk1 10 {port1 pin1}`	`set_input_delay -late -clock clk1 10 {port1 pin1}`

Messages

Unsupported argument -level_sensitive (set_input_delay)

- Severity - Warning
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with one which specifies `level_sensitive` pins or ports

set_input_delay does not accept hierarchical pins (object_name[1])

- Severity - Warning
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with an alternate command that is supported in BuildGates synthesis.

Object_name[2] in the object list for set_input_delay is neither a port nor a pin

- Severity - Warning
- Action - Fix the Synopsys command and rerun the translator.

Set_input_delay pin_port_list does not contain any ports or non-hierarchical pins

- Severity - Error
- Explanation - This warning message appears if the `object_list` for `set_input_delay` contains only hierarchical pins or other invalid design objects (invalid for Synopsys as well).
- Action - Fix the Synopsys command and rerun the translator. For hierarchical pins, replace them with leaf pins.

---

1. The actual object name is substituted here in the warning message.
2. The actual object name is substituted here in the warning message.

Set_input_delay references a clock which is not previously defined

- Severity - Error
- Action - The clock definition statement (`create_clock`) should precede the input delay statement. Fix and rerun the translator.

set_input_transition

The `set_input_transition` command maps to the BuildGates synthesis `set_slew_time` command.

The `set_slew_time` command is used to specify the slew time for a port on the top level module of a design or for a pin on an instance of a technology cell anywhere in the hierarchy. It is useful only in specifying a hard ramp signal as if the pin were driven by a very powerful driver. It does not change the arrival time of an input, but changes the slew time used to compute the delay of the cell on the sink of a net.

Note: The `set_slew_time` command is ignored for a port if `set_drive_cell` or `set_drive_resistance` constraints have been set for that port.

Mapping

Synopsys	BuildGates Synthesis
set_input_transition	set_slew_time
-min	-early
-max	-late
-rise	-rise
-fall	-fall
transition_value	slew_value
port_list	port_list
Examples	
`set_input_transition -min 0.5 {in1}`	`set_slew_time -early 0.5 {in1}`

Messages

A port list must be specified to set_input_transition

- Severity - Error

set_load

The `set_load` command maps to the BuildGates synthesis `set_port_capacitance`, `set_wire_capacitance`, and `set_num_external_sinks` commands.

The `set_port_capacitance` command specifies the capacitance external to the design based on input and output loading from other ports and nets connected to the ports of the current module.

The `set_num_external_sinks` command can be specified on top level input and output ports only. This command sets the constraint for the number of external sinks that are connected to the ports in the port list. This number is factored into the wire capacitance and the wire resistance estimation done for the port nets using the wire load models.

Mapping

Synopsys	BuildGates Synthesis
set_load	set_port_capacitance
-pin_load[a]	(No translation)
-wire_load	(No translation)
-subtract_pin_load	(No translation)
-min	-pvt min
-max	-pvt max
load_value	capacitance_value
port_list	port_list
net_list	set_wire_capacitance
-fanout_number	set_num_external_sinks fanout_value port_list

Examples

`set_load 0.3 {in1 in2}`	`set_port_capacitance 0.3 {in1 in2}`
`set_load 5 -fanout_number {in3}`	`set_num_external_sinks 5 {in3}`

a. This is the default in both BuildGates synthesis and Synopsys.

Messages

Unsupported argument -wire_load (set_load)

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with an alternate command that can be supported in BuildGates synthesis.

Unsupported argument "-subtract_pin_load" (set_load)

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with an alternate command that can be supported in BuildGates synthesis.

The object object_name[1] is neither a port nor a net

- Severity - Error
- Action - This message informs you of an error in the `set_load` statement; fix the command and rerun the translator.

Object_name[2] is an internal net and unsupported in set_num_external_sinks

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with an alternate command that can be supported in BuildGates synthesis.

set_load -fanout_number does not support a minimum/maximum switch

- Severity - Error
- Action - This message informs you of an error in the `set_load` statement. Fix the command and rerun the translator.

---

1. The actual object name is substituted here in the warning message.
2. The actual object name is substituted here in the warning message.

set_logic_one

The `set_logic_one` command maps to the BuildGates synthesis `set_logic1` command for ports and to the `set_constant_for_timing` command for pins.

The `set_logic1` command sets ports to logic 1 so that the optimization tool can do a better optimization of the design.

Mapping

Synopsys	BuildGates Synthesis
set_logic_one	
port_list	set_logic1 *port_list*
pin_list	set_constant_for_timing 1 *pin_list*
Examples	
`set_logic_one {in1}` `set_logic_one {inst1/pin1}`	`set_logic1 {in1}` `set_constant_for_timing 1 {inst1/pin1}`

Messages

None

set_logic_zero

The `set_logic_zero` command maps to the BuildGates synthesis `set_logic0` command for ports and to the `set_constant_for_timing` command for pins.

The `set_logic0` command is used to set ports to logic 0 so that the optimization tool can better optimize the design.

Mapping

Synopsys	BuildGates Synthesis
set_logic_zero	
port_list	set_logic0 port_list
pin_list	set_constant_for_timing 0 *pin_list*
Examples	
`set_logic_zero {in1}` `set_logic_zero {inst1/pin1}`	`set_logic0 {in1}` `set_constant_for_timing 0 {inst1/pin1}`

Messages

None

set_max_capacitance

The `set_max_capacitance` command maps to the BuildGates synthesis `set_port_capacitance_limit` command for ports and to the `set_global max_capacitance_limit` for current design.

The `set_port_capacitance_limit -max` command sets the limit on the capacitance external to the design based on input and output loading from other ports and nets connected to the specified ports of the current module.

The `set_global max_capacitance_limit` command specifies the default capacitance limit on all cells.

Mapping

Synopsys	BuildGates Synthesis
set_max_capacitance	set_port_capacitance_limit -max
capacitance_value	capacitance_value
port_list	port_list
design_list[a]	set_global max_capacitance_limit *capacitance_value*
Examples	
`set_max_capacitance 10 {in1 in2}`	`set_port_capacitance_limit -max 10 {in1 in2}`
`set_max_capacitance 5 current_design`	`set_global max_capacitance_limit 5`

a. Only `current_design` is supported; if the `design_list` contains any other design names, then the translator just issues an error.

Messages

Object_name[1] which is not a port cannot be supported in set_port_capacitance_limit

- Severity - Warning
- Explanation - Only ports and the `current_design` can be supported in the BuildGates synthesis `set_port_capacitance_limit` command.
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with an alternate command that can be supported in BuildGates synthesis.

1. the actual object name is substituted here in the warning message.

set_max_delay

The `set_max_delay` command maps to the BuildGates synthesis `set_path_delay_constraint` command.

The `set_path_delay_constraint` command lets you override the path delay calculated by the timing analyzer. This can be useful for debugging your design or for adding or removing pessimism.

Note: The `set_path_delay_constraint` command requires `set_global path_style_timing_constraint` to be set to `true` which is the default.

Mapping

Synopsys	BuildGates Synthesis
set_max_delay	set_path_delay_constraint -late
-rise	-rise_to
-fall	-fall_to
-from	-from
-through	-through
-to	-to
Examples	
`set_max_delay 10 -from in1 -to out1`	`set_path_delay_constraint 10 -late -from in1 -to out1`

Messages

None

set_max_fanout

The `set_max_fanout` command maps to the BuildGates synthesis `set_fanout_load_limit` command for ports and to the `set_global fanout_load_limit` command for current design.

The `set_fanout_load_limit` command is used to specify the fanout load limit (maximum value) on the ports of a cell. While port capacitance affects timing analysis, fanout loads are used to enforce the design rule checks. The design rule requirement of a maximum fanout load value is set using the global attribute `fanout_load_limit` (see below).

The `set_global fanout_load_limit` command specifies the default fanout load limit used for design rule fixing. The units for fanout are a function of the `default_fanout_load` unit specified in the library (usually = 1).

Mapping

Synopsys	BuildGates Synthesis
set_max_fanout	set_fanout_load_limit
fanout_value	fanout_value
port_list	port_list
design_list[a]	set_global fanout_load_limit *fanout_value*
Examples	
`set_max_fanout 10 {in1 in2}`	`set_fanout_load_limit 10 {in1 in2}`
`set_max_fanout 5 current_design`	`set_global fanout_load_limit 5`

a. Only current_design is supported; if the design_list contains any other design names, then the translator just issues an error.

Messages

Object_name[1] which is not a port cannot be supported in set_max_fanout

- Severity - Warning
- Explanation - Only ports and the `current_design` can be supported in the BuildGates synthesis `set_fanout_load_limit` command.
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with an alternate command that can be supported in BuildGates synthesis.

1. The actual object name is substituted here in the warning message.

set_max_transition

The `set_max_transition` command maps to the BuildGates synthesis `set_slew_time_limit` command for ports and to the `set_global max_slew_time_limit` command for current design.

The `set_slew_time_limit -max` command specifies the limit (maximum) for slew time at the input and output ports of a module. The slew limit is usually derived from the design rule constraints placed on the cells that an output port is driving or an input port is driven by.

The `set_global max_slew_time_limit` command specifies the default slew limit for all cells. The ac_shell will use worse (tighter) of the two limits set in the technology library and set by this variable.

Mapping

Synopsys	BuildGates Synthesis
set_max_transition	set_slew_time_limit -max
transition_value	slew_value
port_list	port_list
design_list[a]	set_global max_slew_time_limit *slew_value*
Examples	
`set_max_transition 10 {in1 in2}`	`set_slew_time_limit -max 10 {in1 in2}`
`set_max_transition 5 current_design`	`set_global max_slew_time_limit 5`

a. Only `current_design` is supported; if the `design_list` contains any other design names, then the translator just issues an error.

Messages

Object_name[1] which is not a port cannot be supported in set_max_transition

- Severity - Warning
- Explanation - Only ports and the `current_design` can be supported in the BuildGates synthesis `set_slew_time_limit` command.
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with an alternate command that can be supported in BuildGates synthesis.

1. The actual object name is substituted here in the warning message.

set_min_capacitance

The `set_min_capacitance` command maps to the BuildGates synthesis `set_port_capacitance_limit` command for ports and to the `set_global min_capacitance_limit` for current design.

The `set_port_capacitance_limit -min` command sets the limit on the capacitance external to the design based on input and output loading from other ports and nets connected to the specified ports of the current module.

The `set_global min_capacitance_limit` command specifies the default capacitance limit on all cells.

Mapping

Synopsys	BuildGates Synthesis
set_min_capacitance	set_port_capacitance_limit -min
capacitance_value	capacitance_value
port_list	port_list
design_list[a]	set_global min_capacitance_limit *capacitance_value*
Examples	
`set_min_capacitance 10 {in1 in2}`	`set_port_capacitance_limit -min 10 {in1 in2}`
`set_min_capacitance 5 current_design`	`set_global min_capacitance_limit 5`

a. Only `current_design` is supported; if the `design_list` contains any other design names, then the translator just issues an error.

Messages

Object_name[1] which is not a port cannot be supported in set_port_capacitance_limit

- Severity - Warning
- Explanation - Only ports and the `current_design` can be supported in the BuildGates synthesis `set_port_capacitance_limit` command.
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with an alternate command that can be supported in BuildGates synthesis.

1. the actual object name is substituted here in the warning message.

set_min_delay

The `set_min_delay` command maps to the BuildGates synthesis `set_path_delay_constraint` command.

The `set_path_delay_constraint` command lets you override the path delay calculated by the timing analyzer. This can be useful for debugging your design or for adding or removing pessimism.

Note: The `set_path_delay_constraint` command requires `set_global path_style_timing_constraint` to be set to `true` which is the default.

Mapping

Synopsys	BuildGates Synthesis
set_min_delay	set_path_delay_constraint -early
-rise	-rise_to
-fall	-fall_to
-from	-from
-through	-through
-to	-to
Examples	
`set_min_delay 10 -from in1 -to out1`	`set_path_delay_constraint 10 -early -from in1 -to out1`

Messages

None

set_min_transition

The `set_min_transition` command maps to the BuildGates synthesis `set_slew_time_limit` command for ports and to the `set_global min_slew_time_limit` command for current design.

The `set_slew_time_limit -min` command specifies the limit (minimum) for slew time at the input and output ports of a module. The slew limit is usually derived from the design rule constraints placed on the cells that an output port is driving or an input port is driven by.

The `set_global min_slew_time_limit` command specifies the default slew limit for all cells. The ac_shell will use best (tighter) of the two limits set in the technology library and set by this variable.

Mapping

Synopsys	BuildGates Synthesis
set_min_transition	set_slew_time_limit -min
transition_value	slew_value
port_list	port_list
design_list[a]	set_global min_slew_time_limit *slew_value*
Examples	
`set_min_transition 10 {in1 in2}`	`set_slew_time_limit -min 10 {in1 in2}`
`set_min_transition 5 current_design`	`set_global min_slew_time_limit 5`

a. Only `current_design` is supported; if the `design_list` contains any other design names, then the translator just issues an error.

Messages

Object_name[1] which is not a port cannot be supported in set_min_transition

- Severity - Warning
- Explanation - Only ports and the `current_design` can be supported in the BuildGates synthesis `set_slew_time_limit` command.
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with an alternate command that can be supported in BuildGates synthesis.

1. The actual object name is substituted here in the warning message.

set_multicycle_path

The `set_multicycle_path` command maps to the BuildGates synthesis `set_cycle_addition` command.

The `set_cycle_addition` command identifies multi-cycle paths in the design for timing analysis. Certain paths in the design are multicycle paths because the signal propagation from the start point to the end point of the path spans over multiple clock cycles. This command lets you add a certain number of cycles to the path under consideration. The information about number of cycles for a path is attached to the pins of the path.

All paths starting in one clock domain and ending in another clock domain are taken as multicycle paths. You can not mix clock domain paths and explicit paths (specified using `-from`, `-to`, and `-through`).

Mapping

Synopsys	BuildGates Synthesis
set_multicycle_path	set_cycle_addition
-hold[a]	-early
-setup[b]	-late
-rise[c]	-rise_to
-fall	-fall_to
-reset_path	(No translation)
-end[d]	-target
-start	(No translation)
-from *object_list*[e]	-from *pin_port_cell_list*
	-clock_from *clock_list*
-through *pin_port_cell_list*[f]	-through *pin_port_cell_list*
-to *object_list*	-to *pin_port_cell_list*
	-clock_to *clock_list*
path_multiplier[g]	cycle_addition_value

Examples

Synopsys	BuildGates Synthesis
`set_multicycle_path 2 -setup -from {in1 inst1} -to {out1 inst2}`	`set_cycle_addition 1 -late -target -from {in1 inst1} -to {out1 inst2}`
`set_multicycle_path 3 -setup -through i1 -through i2 -through i3`	`set_cycle_addition 2 -late -target -through i1 -through i2 -through i3`
`set_multicycle_path 2 -setup -from clk2 -to clk3`	`set_cycle_addition 1 -late -target -clock_from clk2 -clock_to clk3`
`set_multicycle_path 2 -rise -setup -to u10`	`set_cycle_addition 1 -late -target -rise_to u10`
`set_multicycle_path 2 -hold -from {in1 inst1} -to {out1 inst2}`[h]	`set_cycle_addition ? -early -from {in1 inst1} -to {out1 inst2}`
`set_multicycle_path 2 -from {in2} -to {out2}`	`set_cycle_addition 1 -late -target -from {in2} -to {out2}` `set_cycle_addition ? -early -from {in2} -to {out2}`

`set_multicycle_path 2 -setup -from {u1 clk1} -through {u2} -to {u2 clk2}`	`set_cycle_addition 1 -late -target -from {u1} -through {u2} -to {u2}`
	`set_cycle_addition 1 -late -target -from {u1} -through {u2} -clock_to {clk2}`
	`set_cycle_addition 1 -late -target -clock_from {clk1} -through {u2} -to {u2}`
	`set_cycle_addition 1 -late -target -clock_from {clk1} -through {u2} -clock_to {clk2}`

a. Synopsys and BuildGates synthesis tools apply the hold checks differently; in Synopsys modifying the setup check also results in moving the hold check relative to the same active clock edges as the setup check while in BuildGates synthesis, modifying the setup check has no effect on the hold check which is still performed relative to the original active edges. Since the translator cannot determine the hold multiplier value to be used, it puts a '?' for the value, and the translated statement is put in a comment. It is up to you to determine the hold multiplier, substitute the value and uncomment the BuildGates synthesis command corresponding to the hold.

b. If both `setup` and `hold` switches are omitted in Synopsys, then the specified path multiplier applies to the setup check while the hold check multiplier gets reset to '0'. On the other hand in BuildGates synthesis, if neither the `late` or the `early` switch is specified, the cycle addition applies to only the late (or setup) check. So for a `set_multicycle_path` that does not specify setup or hold, there are two corresponding BuildGates synthesis `set_cycle_addition` statements.

c. Synopsys allows you to mark only rising or falling paths multicycle, as measured on the path endpoint. BuildGates synthesis allows you to specify rising or falling multicycle paths, as measured on the path endpoint, only if the `to` option is also specified in the multicycle path, in which case `rise_to`/`fall_to` replace the `to` option.

d. The default in Synopsys is to perform the setup check relative to the end clock, and the hold check relative to the start clock. The default in BuildGates synthesis is to perform both the setup and hold checks relative to the start clock. Hence there is no switch in BuildGates synthesis for `-start`.

e. Synopsys allows clocks, pins, ports or cells in the `from`/`to` lists. BuildGates synthesis also allows all these object types in its `set_cycle_addition` command, but the clocks need to be specified in a separate list from the other object types. The translator takes care of doing this separation.

f. Both Synopsys and BuildGates synthesis allow multiple through options to be specified in a `set_multicycle_path`/`set_cycle_addition` command.

g. In Synopsys, this value denotes the number of cycles that the data path must have for setup or hold relative to the startpoint or endpoint clock before data is required at the endpoint. In BuildGates synthesis, the corresponding value denotes the number of cycles that need to be added to the default single-cycle relationship. The translator takes care of adjusting this value for setup checks; for hold, it is up to you to determine this value.

h. Note that the translated BuildGates synthesis statement is commented and it is up to you to determine the hold multiplier, substitute the value and uncomment this BuildGates synthesis command.

Messages

Unsupported argument -reset_path (set_multicycle_path)

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Remove this statement and the earlier statement that specifies this path as a multicycle path.

Ignoring '-rise' for 'to_clock_list[1]' since BuildGates synthesis only supports edge specification if the path endpoint is specified

- Severity - Warning
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this construct with an alternate construct which can be supported in BuildGates synthesis.

  Note: A similar warning is issued for ‘-fall’.

Ignoring clocks 'clock_list[2]' in -through list

- Severity - Warning
- Action - Fix the Synopsys command and rerun the translator.

---

1. the actual clock names are substituted here in the warning message.
2. the actual clock names are substituted here in the warning message.

Unable to locate objects specified to set_multicycle_path in from/through/to lists

- Severity - Error
- Explanation - This message arises if the objects specified in the `from`/`through`/`to` lists cannot be found in the current design.
- Action - Fix the Synopsys command and rerun the translator.

Set_multicycle_path takes either a -start or a -end switch; not both at the same time

- Severity - Error
- Action - Fix the Synopsys command and rerun the translator.

Ignoring hold multiplier since set_multicycle_path and set_cycle_addition interpret hold relationships differently

- Severity - Error
- Action - Find the corresponding commented `set_cycle_addition -early` statement in the BuildGates synthesis output, determine the hold multiplier, substitute the value and uncomment this BuildGates synthesis command.

set_operating_conditions

The `set_operating_conditions` command maps to the BuildGates synthesis `set_operating_condition` command.

The `set_operating_conditions` command specifies the operating conditions; process, temperature and voltage for the design from a given library. The named operating condition is used by the mapping and optimization commands to appropriately select cells.

Mapping

Synopsys	BuildGates Synthesis
set_operating_conditions	set_operating_conditions
-library *lib_name*	-library lib_name
-min *min_condition*[a]	-pvt min min_condition
-max *max_condition*	-pvt max max_condition
-min_library *lib_name*	-library lib_name
-max_library *lib_name*	-library lib_name
condition	condition
Examples	
`set_operating_conditions -library TECH WCCOM`	`set_operating_conditions -library TECH WCCOM`
`set_operating_conditions -max_library TECH -max WCCOM`	`set_operating_conditions -library TECH -pvt max WCCOM`
`set_operating_conditions -max WCCOM -min BCCOM`	`set_operating_conditions -pvt max WCCOM` `set_operating_conditions -pvt min BCCOM`

a. The min condition cannot be specified without also specifying the max condition.

Messages

The -min option cannot be used without the -max option of set_operating_conditions

- Severity - Error
- Action - Fix the Synopsys command and rerun the translator.

An operating condition must be specified in set_operating_conditions

- Severity - Error
- Action - Fix the Synopsys command and rerun the translator.

set_output_delay

The `set_output_delay` command maps to the BuildGates synthesis `set_external_delay` command.

The `set_external_delay` command provides an alternate way to specify the data required time at the output port using external delay requirement. It considers the timing requirement of the configuration of the downstream block the output port is connected to.

Mapping

Synopsys	BuildGates Synthesis
set_output_delay	set_external_delay
-min	-early
-max	-late
-rise	-rise
-fall	-fall
-clock *clock_name*	-clock *clock_name*
-clock_fall	-trail
-add_delay[a]	(No translation)
-level_sensitive	(No translation)
-group_path	(No translation)
delay_value	delay_value
port_or_pin_list	port_or_pin_list
Examples	
`set_output_delay -min -clock clk1 5 {port1 pin1}`	`set_external_delay -early -clock clk1 5 {port1 pin1}`
`set_output_delay -max -clock clk1 10 {port1 pin1}`	`set_external_delay -late -clock clk1 10 {port1 pin1}`

a. In Synopsys the `add_delay` switch specifies whether to add delay information to the existing output delay, or to overwrite `add_delay` enables you to capture information about multiple paths leading from an output port that are relative to different clocks or clock edges. The default behavior in BuildGates synthesis is to allow different delay information relative to different clocks/edges.

Messages

Ignored argument -group_path (set_output_delay)

- Severity - Warning
- Action - This message informs you of the BuildGates synthesis command limitations. Check if the translation is acceptable.

Unsupported argument -level_sensitive (set_output_delay)

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with one which specifies `level_sensitive` pins or ports.

Set_external_delay does not accept hierarchical pins (object_name[1])

- Severity - Warning
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with an alternate command that is supported in BuildGates synthesis.

Object_name[2] in the object list for set_output_delay is neither a port nor a pin

- Severity - Warning
- Action - Fix the Synopsys command and rerun the translator.

1. The actual object name is substituted here in the warning message.
2. The actual object name is substituted here in the warning message.

Set_output_delay pin_port_list does not contain any ports or non-hierarchical pins

- Severity - Error
- Explanation - This warning message appears if the `object_list` for `set_output_delay` contains only hierarchical pins or other invalid design objects (invalid for Synopsys as well).
- Action - Fix the Synopsys command and rerun the translator; for hierarchical pins, replace them with leaf pins.

If multiple set_external_delay commands are applied, then the worst case required time will be used (modeling -add_delay behavior)

- Severity - Warning
- Action - This message informs you of the BuildGates synthesis command behavior. Multiple `set_external_delay` commands applied to the same output port do not overwrite previous delay values:

  If the delay information is relative to different clocks then all the delay information is retained and used appropriately.

  If the delay is relative to a clock for which there is a previously specified value, then the worst case required time is established for that output port relative to the specified clock. This, in essence, models Synopsys behavior if the `-add_delay` switch were specified.

set_port_fanout_number

The `set_port_fanout_number` command maps to the BuildGates synthesis `set_num_external_sinks` command.

The `set_num_external_sinks` command can be specified on top level input and output ports only. This command sets the constraint for the number of external sinks that are connected to the ports in the port list. This number is factored into the wire capacitance and the wire resistance estimation done for the port nets using the wire load models.

Mapping

Synopsys	BuildGates Synthesis
set_port_fanout_number	set_num_external_sinks
fanout_value	fanout_value
port_list	port_list
Examples	
`set_port_fanout_number 10 {in1 in2}`	`set_num_external_sinks 10 {in1 in2}`

Messages

None

set_propagated_clock

The `set_propagated_clock` command maps to the BuildGates synthesis `set_clock_propagation` command.

The `set_clock_propagation` command tells the timing analyzer to treat all clock signals associated with the current timing top module as either ideal or propagated signals for the purpose of timing analysis. One of the signal types must be specified when using this command.

Mapping

Synopsys	BuildGates Synthesis
set_propagated_clock	set_clock_propagation propagated
object_list[a]	(No translation)
Examples	
`set_propagated_clock all_clocks()`	`set_clock_propagation propagated`

a. Synopsys allows clocks, ports, pins, or cells in the object list.

Messages

The BuildGates synthesis command: `set_clock_propagation` does not allow selected clock networks to have propagated clock latency.

Setting all clock networks to have propagated clock latency

- Severity - Warning
- Explanation - BuildGates synthesis only supports propagated clock delays for all clock networks and not on selected networks. The translator goes ahead and sets all clock networks to have propagated clock latency.
- Action - This message informs you of the BuildGates synthesis command limitations. Check whether the translation is acceptable.

set_resistance

The `set_resistance` command maps to the BuildGates synthesis `set_wire_resistance` command.

The `set_wire_resistance` command is used to backannotate the resistance on each net after the physical design has been done.The netlist can then be re-analyzed for timing.

Mapping

Synopsys	BuildGates Synthesis
set_resistance	set_wire_resistance
resistance_value	resistance_value
net_list	net_list
-min|-max	-pvt min|max
Examples	
`set_resistance 10 {net1 net2}`	`set_wire_resistance 10 {net1 net2}`

Messages

None

set_wire_load

The `set_wire_load` command maps to the BuildGates synthesis `set_wire_load` and `set_port_wire_load` commands.

See set_wire_load_mode on page 109 for more information.

See set_wire_load_model on page 110 for information about `set_wire_load`.

The `set_port_wire_load` command specifies the wire load for an input or output top level port of a design. The net connected to the port is associated with the specified wireload model, which is used for wire capacitance and resistance estimation.

Mapping

Synopsys	BuildGates Synthesis
set_wire_load	set_wire_load
-library *library_name*	-library library_name
-min | -max	-pvt min | max
-min_block_size *value*	(No translation)
-cluster *cluster_name*	(No translation)
-selection_group *group_name*	(No translation)
wire_load	model_name
port_list	set_port_wire_load *wire_load port_list*[a]
-mode *mode_name*	set_wire_load_mode *mode_name*
Examples	
`set_wire_load -library lca -name wire_load_model1`	`set_wire_load -library lca wire_load_model1`
`set_wire_load -library lca -name wire_load2 {port1 port2}`	`set_port_wire_load -library lca wire_load2 {port1 port2}`
`set_wire_load -mode top`	`set_wire_load_mode top`

a. If the Synopsys command uses any of the options like `-min`|`-max` or `-library`, then those will be translated over to BuildGates synthesis `set_port_wire_load` in exactly the same way as for BuildGates synthesis `set_wire_load`.

Messages

Unsupported argument “-cluster” (set_wire_load)

- Severity - Error
- Explanation - Synopsys allows you to define a cluster of instances and specify a wire load model for the entire cluster in a single command. However, the BuildGates synthesis `set_wire_load` command does not support clusters; it does support a list of instances.
- Action - This message informs you of the BuildGates synthesis command limitations. You can manually translate this statement to a `set_wire_load` with a list of instances that form the cluster

Unsupported argument -selection_group (set_wire_load)

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with an alternate command that is supported in BuildGates synthesis.

Unsupported argument -min_block_size (set_wire_load)

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Replace this command with an alternate command that is supported in BuildGates synthesis.

The 'segmented' wire load mode is not supported in Ambit

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Choose one of the modes that is supported in BuildGates synthesis.

set_wire_load_mode

The `set_wire_load_mode` command maps to the BuildGates synthesis `set_wire_load_mode` command.

The `set_wire_load_mode` command controls the determination of the wireload model.

Mapping

Synopsys	BuildGates Synthesis
set_wire_load_mode	set_wire_load_mode
top	top
enclosed	enclosed
segmented	(No translation)
Examples	
`set_wire_load_mode enclosed`	`set_wire_load_mode enclosed`

Messages

The 'segmented' wire load mode is not supported in Ambit

- Severity - Error
- Action - This message informs you of the BuildGates synthesis command limitations. Choose one of the modes that is supported in BuildGates synthesis.

set_wire_load_model

The `set_wire_load_model` command maps to the BuildGates synthesis `set_wire_load` and `set_port_wire_load` commands.

The `set_wire_load` command is used to specify the wireload model to be used from the technology library. It sets the wire load on the current instance, if the list of instances is omitted. Otherwise, the wire load is set on all instances specified.

The `set_port_wire_load` command specifies the wire load for an input or output top level port of a design. The net connected to the port is associated with the specified wireload model, which is used for wire capacitance and resistance estimation.

Mapping

Synopsys	BuildGates Synthesis
set_wire_load_model	set_wire_load
-library *library_name*	-library library_name
-min | -max	-pvt min | max
-cluster *cluster_name*	(No translation)
-name *model_name*	model_name
port_list	set_port_wire_load *model_name port_list*[a]
Examples	
`set_wire_load_model -library lca -name wire_load_model1`	`set_wire_load -library lca wire_load_model1`
`set_wire_load_model -library lca -name wire_load2 {port1 port2}`	`set_port_wire_load -library lca wire_load2 {port1 port2}`

a. If the Synopsys command uses any of the options like `-min`, `-max`, or `-library`, those will be translated over to the BuildGates synthesis `set_port_wire_load` command (similar to the BuildGates synthesis `set_wire_load` command).

Messages

Unsupported argument -cluster (set_wire_load_model)

- Severity - Error
- Explanation - Synopsys allows you to define a cluster of instances and specify a wire load model for the entire cluster in a single command. However, the BuildGates synthesis `set_wire_load` command does not support clusters; it does support a list of instances.
- Action - This message informs you of the BuildGates synthesis command limitations. You can manually translate this statement to a `set_wire_load` with a list of instances that form the cluster.

set_wire_load_selection_group

The `set_wire_load_selection_group` command maps to the BuildGates synthesis `set_wire_load_selection_table` command.

The `set_wire_load_selection_table` command is used to override the wire load selection table in the library, `library_name`. When `library_name` is the target technology, the specified `wire_load_selection_table` is in effect.

Mapping

Synopsys	BuildGates Synthesis
set_wire_load_selection_group	set_wire_load_selection_table
-library	-library
-cluster	Unsupported
-min	-pvt min
-max	-pvt max

Messages

set_wire_load_selection_group for a cluster cannot be supported in Ambit

- Severity - Error
- Action - Informs you of BuildGates synthesis command limitations.